



Guida per gli sviluppatori

# Amazon Pinpoint



# Amazon Pinpoint: Guida per gli sviluppatori

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

---

# Table of Contents

Che cos'è Amazon Pinpoint? .....	1
Funzionalità di Amazon Pinpoint .....	1
Definizione di segmenti di destinatari .....	1
Coinvolgimento dei destinatari con le campagne di messaggistica .....	2
Invio di messaggi transazionali .....	2
Analisi del comportamento degli utenti .....	2
Disponibilità regionale .....	2
Tutorial .....	4
Utilizzo di Postman con Amazon Pinpoint .....	4
Informazioni sul tutorial .....	4
Prerequisiti .....	5
Passaggio 1: creare policy e ruoli IAM .....	6
Fase 2: configurazione di Postman .....	10
Fase 3: invio di richieste aggiuntive .....	18
Configurazione di un sistema di registrazione SMS .....	25
Informazioni sulla doppia attivazione .....	26
Informazioni su questa soluzione .....	26
Prerequisiti .....	28
Passaggio 1: configurare Amazon Pinpoint .....	29
Passaggio 2: creare policy e ruoli IAM .....	35
Passaggio 1: creare funzioni Lambda .....	38
Passaggio 4: configurare Gateway Amazon API .....	50
Fase 5: creazione e distribuzione del modulo Web .....	55
Fasi successive .....	63
Integrazione con l'applicazione .....	67
Lavorare con gli SDK AWS .....	68
Connessione dell'applicazione front-end con AWS Amplify .....	69
Approfondimenti .....	69
Registrazione di endpoint .....	69
Prima di iniziare .....	70
SDK AWS per dispositivi mobili .....	70
AWS Amplify .....	70
Fasi successive .....	70
Segnalazione degli eventi .....	71

Prima di iniziare .....	72
SDK AWS per dispositivi mobili .....	72
Web e React Native .....	70
API Eventi di Amazon Pinpoint .....	73
Approfondimenti .....	73
Gestione delle notifiche push .....	73
Configurazione delle notifiche push .....	73
Gestione delle notifiche push .....	76
Definizione dei destinatari .....	77
Aggiunta di endpoint .....	78
Esempi .....	78
Informazioni correlate .....	84
Associazione di utenti agli endpoint .....	84
Esempi .....	85
Informazioni correlate .....	90
Aggiunta di un batch di endpoint .....	90
Esempi .....	90
Informazioni correlate .....	98
Importazione di endpoint .....	98
Prima di iniziare .....	99
Esempi .....	99
Informazioni correlate .....	112
Eliminazione di endpoint .....	112
Esempi .....	112
Gestione degli endpoint dei membri del pubblico .....	115
Accesso ai dati dei destinatari .....	117
Ricerca di endpoint .....	118
Esempi .....	118
Informazioni correlate .....	123
Esportazione di endpoint .....	123
Prima di iniziare .....	124
Esempi .....	124
Informazioni correlate .....	135
Acquisizione dell'elenco degli ID endpoint .....	135
Creazione di segmenti .....	139
Creazione di segmenti .....	139

Creazione di segmenti con l'AWS SDK for Java .....	139
Importazione di segmenti .....	143
Importazione di un segmento .....	143
Personalizzazione dei segmenti con AWS Lambda .....	145
Dati eventi .....	147
Creazione di una funzione Lambda .....	148
Assegnazione di una policy di funzione Lambda .....	150
Assegnazione di una funzione Lambda a una campagna .....	153
Creazione di campagne .....	154
Creazione di campagne standard .....	154
Creazione di campagne con AWS SDK for Java .....	154
Creazione di campagne di test A/B .....	157
Creazione di campagne di test A/B con AWS SDK for Java .....	157
Utilizzo dell'API SMS e Voce .....	160
Invio e convalida di codici OTP .....	162
Invio di un messaggio OTP .....	162
Risposta SendOtpMessage .....	165
Convalida di un messaggio OTP .....	165
Risposta VerifyOtpMessage .....	166
Esempi di codice .....	167
Generazione di un ID di riferimento .....	167
Invio di codici OTP .....	167
Convalida dei codici OTP .....	169
Invio e recupero di messaggi in-app .....	171
Recupero dei messaggi in-app per un endpoint .....	171
Informazioni sulle risposte delle API GetInAppMessages .....	173
Oggetto InAppMessageCampaigns .....	175
Oggetto InAppMessage .....	177
Oggetto HeaderConfig .....	178
Oggetto BodyConfig .....	178
Oggetto InAppMessageContent .....	179
Oggetto Schedule .....	180
Oggetto InAppMessageButton .....	181
Oggetto DefaultButtonConfig .....	182
Oggetto OverrideButtonConfig .....	183
Convalida dei numeri di telefono .....	185

Casi d'uso della funzione di convalida dei numeri di telefono .....	185
Utilizzo del servizio di convalida dei numeri di telefono .....	186
Risposte di convalida dei numeri di telefono .....	187
Invio di messaggi .....	191
Invio di e-mail .....	191
Scelta di un metodo per l'invio di e-mail .....	192
Scegliere tra Amazon Pinpoint e Amazon Simple Email Service (SES) .....	192
Utilizzo dell'API .....	192
Invio di un'e-mail con intestazioni di annullamento dell'iscrizione .....	207
Invio di messaggi SMS .....	209
Invio di messaggi vocali .....	222
Invio di notifiche push .....	230
Creazione di canali personalizzati .....	240
Creazione di una campagna che invia messaggi attraverso un canale personalizzato .....	240
Informazioni sui dati degli eventi .....	241
Configurazione dei webhook .....	243
Configurazione delle funzioni Lambda .....	243
Funzione Lambda di esempio .....	243
Formato di risposta della funzione Lambda per Amazon Pinpoint .....	247
Concessione dell'autorizzazione Amazon Pinpoint a richiamare la funzione Lambda .....	249
Streaming di eventi .....	252
Configurazione dello streaming degli eventi .....	253
Prerequisiti .....	253
AWS CLI .....	253
AWS SDK for Java .....	254
Disabilitazione dello streaming di eventi .....	255
Eventi app .....	255
Esempio .....	255
Attributi di eventi dell'app .....	256
Eventi di campagna .....	261
Evento di esempio .....	262
Attributi degli eventi di campagna .....	263
Eventi di viaggio .....	269
Evento di esempio .....	270
Attributi di eventi di viaggio .....	271
Eventi e-mail .....	275

Eventi di esempio .....	276
Attributi di eventi e-mail .....	282
Eventi SMS .....	289
Esempio .....	289
Attributi di eventi SMS .....	290
Query sui dati di analisi .....	301
Parametri supportati .....	301
Nozioni di base delle query .....	303
Policy IAM .....	304
Parametri standard .....	308
Parametri delle applicazioni per le campagne .....	309
Parametri dell'applicazione per messaggi e-mail transazionali .....	315
Parametri dell'applicazione per messaggi SMS transazionali .....	327
Parametri della campagna .....	335
Parametri di coinvolgimento del viaggio .....	348
Parametri di esecuzione del viaggio .....	356
Parametri di esecuzione delle attività di viaggio .....	358
Metriche relative all'esecuzione di percorsi e campagne .....	364
Query sui dati della campagna .....	366
Prerequisiti .....	367
Esecuzione di query sui dati di una campagna .....	368
Interrogazione dei dati per più campagne .....	373
Query sui dati di messaggistica transazionale .....	380
Prerequisiti .....	380
Query dei dati per i messaggi e-mail transazionali .....	381
Query dei dati per i messaggi SMS transazionali .....	386
Utilizzo dei risultati delle query .....	392
Struttura JSON .....	393
Oggetti e campi JSON .....	398
Registrazione di chiamate API .....	400
Informazioni su Amazon Pinpoint in CloudTrail .....	400
Azioni dell'API Amazon Pinpoint che possono essere registrate da CloudTrail .....	401
Azioni dell'API e-mail di Amazon Pinpoint che possono essere registrate da CloudTrail .....	406
Azioni dell'API vocale e SMS di Amazon Pinpoint versione 1 che possono essere registrate da CloudTrail .....	407
Esempi: voci del file di log Amazon Pinpoint .....	407

Assegnazione di tag alle risorse .....	413
Gestione dei tag .....	413
Utilizzo di tag nelle policy IAM .....	414
Aggiunta di tag alle risorse .....	415
Aggiunta di tag mediante l'API .....	415
Aggiunta di tag utilizzando la AWS CLI .....	416
Visualizzazione dei tag per le risorse .....	417
Visualizzazione dei tag mediante l'API .....	418
Visualizzazione dei tag mediante la AWS CLI .....	418
Aggiornamento dei tag per le risorse .....	419
Rimozione dei tag dalle risorse .....	420
Rimozione dei tag mediante l'API .....	420
Rimozione dei tag mediante la AWS CLI .....	421
Informazioni correlate .....	421
Personalizzazione dei suggerimenti con AWS Lambda .....	422
Utilizzo dei suggerimenti nei messaggi .....	422
Creazione di una funzione Lambda .....	425
Dati dell'evento di input .....	425
Dati e requisiti di risposta .....	427
Assegnazione di una policy di funzione Lambda .....	432
Autorizzazione di Amazon Pinpoint a richiamare la funzione .....	434
Configurazione del modello della funzione di suggerimento .....	435
Eliminazione di dati .....	436
Eliminazione di endpoint .....	436
Eliminazione dei dati relativi a segmenti ed endpoint da Amazon S3 .....	437
Eliminazione di tutti i dati del progetto .....	437
Eliminazione di tutti i dati AWS .....	438
Esempi di codice .....	440
Amazon Pinpoint .....	440
Azioni .....	441
API SMS e Voce di Amazon Pinpoint .....	527
Azioni .....	527
Sicurezza .....	537
Protezione dei dati .....	538
Crittografia dei dati .....	540
Riservatezza del traffico Internet .....	540

Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint .....	541
Gestione dell'identità e degli accessi .....	543
Destinatari .....	543
Autenticazione con identità .....	544
Gestione dell'accesso con policy .....	547
Come Amazon Pinpoint funziona con IAM .....	550
Azioni delle policy Amazon Pinpoint .....	557
Esempi di policy basate su identità .....	588
Ruoli IAM per attività comuni .....	602
Risoluzione dei problemi .....	619
Registrazione di log e monitoraggio .....	621
Convalida della conformità .....	623
Resilienza .....	624
Sicurezza dell'infrastruttura .....	624
Analisi della configurazione e delle vulnerabilità .....	625
Best practice di sicurezza .....	626
Quote .....	627
Quote per i progetti .....	627
Quote di richiesta API .....	628
Quote per le campagne .....	630
Quote per l'e-mail .....	632
Quote per i messaggi e-mail .....	633
Quote per mittenti e destinatari di e-mail .....	633
Quote di invio e-mail .....	634
Quote per gli endpoint .....	636
Quote per l'importazione degli endpoint .....	637
Quote per l'inserimento di eventi .....	638
Quote per i viaggi .....	639
Quote di Lambda .....	640
Quote di machine learning .....	641
Quote per i modelli di messaggio .....	642
Quote per le notifiche push .....	644
Quote per i messaggi in-app .....	644
Quote per i segmenti .....	645
Quote per SMS .....	646
Quote 10DLC .....	648

---

Quote per i messaggi vocali .....	649
Richiesta di aumento delle quote .....	651
Cronologia dei documenti .....	654
Aggiornamenti precedenti .....	663
.....	dclxviii

# Che cos'è Amazon Pinpoint?

Amazon Pinpoint è un servizio AWS che puoi utilizzare per interagire con i clienti su diversi canali di messaggistica. Puoi utilizzare Amazon Pinpoint per inviare notifiche push, e-mail, messaggi di testo (SMS) o messaggi vocali.

Le informazioni contenute in questa guida per lo sviluppatore è destinato agli sviluppatori di applicazioni. Questa guida contiene informazioni su come usare le funzionalità di Amazon Pinpoint a livello di programmazione. Contiene anche informazioni di particolare interesse per gli sviluppatori di app per dispositivi mobili, come procedure per [l'integrazione di caratteristiche di analisi e messaggistica con l'applicazione](#).

A questo documento si affiancano molti altri documenti. I documenti seguenti forniscono informazioni di riferimento correlate alle API Amazon Pinpoint:

- [Documentazione di riferimento dell'API Amazon Pinpoint](#)
- [API SMS e Voce di Amazon Pinpoint](#)

Se non hai familiarità con Amazon Pinpoint, prima di continuare con questo documento è consigliabile consultare la [Guida per l'utente di Amazon Pinpoint](#).

## Funzionalità di Amazon Pinpoint

Questa sezione descrive le principali funzionalità di Amazon Pinpoint e le attività che consentono di eseguire.

### Definizione di segmenti di destinatari

Per raggiungere gli utenti appropriati per i tuoi messaggi, puoi [definire segmenti di destinatari](#). Un segmento specifica quali utenti riceveranno i messaggi inviati da una campagna. Puoi definire segmenti dinamici in base ai dati segnalati dall'applicazione, ad esempio il sistema operativo o il tipo di dispositivo mobile. È anche possibile importare segmenti statici definiti utilizzando un altro servizio o un'altra applicazione.

## Coinvolgimento dei destinatari con le campagne di messaggistica

Per coinvolgere i destinatari, puoi [creare una campagna di messaggistica](#). Una campagna prevede l'invio di messaggi personalizzati secondo una pianificazione definita. Puoi creare campagne per inviare messaggi push su dispositivi mobili, e-mail o SMS.

Per sperimentare diverse strategie per le campagne, configura la tua campagna come test A/B e analizza i risultati con gli strumenti di analisi di Amazon Pinpoint.

## Invio di messaggi transazionali

Tieni costantemente informati i tuoi clienti inviando messaggi push e SMS transazionali per dispositivi mobili, come messaggi di attivazione di nuovi account, conferme d'ordine e notifiche di reimpostazione della password, direttamente a utenti specifici. È possibile inviare messaggi transazionali mediante la REST API Amazon Pinpoint.

## Analisi del comportamento degli utenti

Acquisisci informazioni approfondite sui destinatari e sull'efficacia delle tue campagne utilizzando gli strumenti di analisi disponibili in Amazon Pinpoint. Puoi visualizzare le tendenze relative al livello di coinvolgimento, alle attività di acquisto, ai dati demografici e molto altro dei tuoi utenti. È anche possibile monitorare il traffico dei messaggi visualizzando i parametri, ad esempio il numero totale di messaggi che sono stati inviati o aperti per una campagna o un'applicazione. Tramite l'API Amazon Pinpoint, l'applicazione è in grado di trasmettere i dati personalizzati resi disponibili da Amazon Pinpoint per l'analisi ed è quindi possibile eseguire query sui dati di analisi in base a determinate metriche standard.

Per analizzare o archiviare dati di analisi al di fuori di Amazon Pinpoint, puoi configurare Amazon Pinpoint per lo [streaming dei dati](#) su Amazon Kinesis.

## Disponibilità regionale

Amazon Pinpoint è disponibile in diverse regioni AWS in Nord America, Europa, Asia e Oceania. In ogni regione, AWS gestisce più zone di disponibilità. Queste zone di disponibilità sono fisicamente isolate l'una dall'altra, ma sono unite da connessioni di rete private a bassa latenza, a velocità effettiva elevata e altamente ridondanti. Queste zone di disponibilità ci consentono di fornire livelli molto elevati di disponibilità e ridondanza, riducendo al minimo la latenza.

Per ulteriori informazioni sulle Regioni AWS, consulta [Managing Regioni AWS](#) (Gestione delle Regioni AWS) nei Riferimenti generali di Amazon Web Services.. Per un elenco di tutte le regioni in cui Amazon Pinpoint è attualmente disponibile, consulta [Endpoint e quote di Amazon Pinpoint](#) e [Endpoint dei servizi AWS](#) nella Riferimenti generali di Amazon Web Services. Per ulteriori informazioni sul numero di zone di disponibilità presenti in ciascuna regione, consulta [Infrastruttura globale AWS](#).

# Tutorial

I tutorial in questa sezione sono concepiti per mostrare ai nuovi utenti di Amazon Pinpoint come svolgere varie importanti attività. Se non conosci ancora Amazon Pinpoint oppure se hai poca familiarità con determinate funzionalità, questi tutorial sono un ottimo modo per iniziare.

I tutorial in questa guida includono attività orientate verso un pubblico di amministratori di sistema o sviluppatori. Questi tutorial illustrano come eseguire attività utilizzando l'API Amazon Pinpoint, AWS SDK e la AWS CLI. Se interagisci principalmente con Amazon Pinpoint utilizzando la console basata sul Web, consulta la sezione dei tutorial nella Guida per l'utente di Amazon Pinpoint.

## Tutorial

- [Tutorial: utilizzo di Postman con l'API Amazon Pinpoint](#)
- [Tutorial: configurazione di un sistema di registrazione SMS](#)

## Tutorial: utilizzo di Postman con l'API Amazon Pinpoint

Postman è un popolare strumento che consente di testare le API in un ambiente grafico di facile utilizzo. È possibile utilizzare Postman per inviare richieste API a qualsiasi API REST e per ricevere le relative risposte. Postman è una soluzione pratica per testare e risolvere i problemi delle chiamate effettuate all'API Amazon Pinpoint. In questo tutorial sono descritte le procedure per la configurazione e l'utilizzo di Postman con Amazon Pinpoint.

### Note

Postman è sviluppato da una società di terze parti. Non è sviluppato o supportato da Amazon Web Services (AWS). Per ulteriori informazioni sull'utilizzo di Postman o per ricevere assistenza per i problemi correlati a questo strumento, consulta il [Centro di supporto](#) sul sito Web di Postman.

## Informazioni sul tutorial

Questa sezione contiene una panoramica di questo tutorial.

### Destinatari principali

Questo tutorial è rivolto a un pubblico di sviluppatori e di implementatori di sistema. Non è necessario avere familiarità con Amazon Pinpoint o Postman per completare le fasi descritte in questo tutorial. È necessario avere dimestichezza con la gestione di policy IAM e con la modifica degli esempi di codice JSON.

Le procedure in questo tutorial sono state pensate per impedire ai nuovi utenti di utilizzare le operazioni API in grado di eliminare definitivamente le risorse Amazon Pinpoint. Gli utenti esperti possono rimuovere questa limitazione modificando la policy associata ai propri utenti .

## Caratteristiche utilizzate

Questo tutorial include esempi di utilizzo della seguente funzionalità di Amazon Pinpoint:

- Interazione con l'API Amazon Pinpoint tramite Postman

## Tempo richiesto

Il completamento di questo tutorial richiede circa 15 minuti.

## Limitazioni regionali

Non sono previste limitazioni regionali associati all'utilizzo di questa soluzione.

## Costi di utilizzo delle risorse

Non vi sono costi aggiuntivi per la creazione di un account AWS. Tuttavia, implementando questa soluzione, potrebbero essere applicati costi di utilizzo di AWS qualora utilizzassi Postman per eseguire una delle operazioni seguenti:

- Inviare e-mail, SMS, push su dispositivi mobili o messaggi vocali
- Creare e inviare campagne
- Utilizzare la funzione di convalida del numero di telefono

Per ulteriori informazioni sui costi associati all'utilizzo di Amazon Pinpoint, consulta [Prezzi di Amazon Pinpoint](#).

## Prerequisiti

Prima di iniziare questo tutorial, completare i prerequisiti descritti di seguito:

- Devi disporre di un account AWS. Per creare un account AWS, vai su <https://console.aws.amazon.com/> e scegli Crea un nuovo account AWS.
- Verifica che l'account utilizzato per accedere alla AWS Management Console sia in grado di creare nuove policy e nuovi ruoli IAM.
- Assicurati di aver creato almeno un progetto di esempio con la funzionalità e-mail attivata e un'identità e-mail verificata. Consulta [Creazione di un progetto Amazon Pinpoint con supporto e-mail](#) nella Guida per l'utente di Amazon Pinpoint.
- Assicurati di avere un ID account AWS. L'ID account AWS è disponibile nell'angolo superiore destro della console oppure è possibile utilizzare l'interfaccia a riga di comando (CLI). Consulta [Trova il tuo ID Account AWS](#).
- È necessario scaricare e installare Postman sul computer. È possibile scaricare Postman dal [sito Web Postman](#).
- Dopo aver installato Postman sul computer, crea un account Postman. Al primo avvio dell'applicazione Postman, verrà richiesto di effettuare l'accesso o di creare un nuovo account. Segui le istruzioni fornite da Postman per accedere al tuo account o per creare un account se non lo hai già creato.

## Passaggio 1: creare policy e ruoli IAM

Quando usi Postman per testare l'API Amazon Pinpoint, il primo passaggio prevede la creazione di un utente. In questa sezione, dovrai una policy che consenta agli utenti di interagire con tutte le risorse Amazon Pinpoint e quindi creare un utente a cui collegare direttamente la policy.

### Creazione di una policy IAM

Scopri come creare una policy IAM. Gli utenti e i ruoli che utilizzano questa policy sono in grado di interagire con tutte le risorse nell'API Amazon Pinpoint. Tale policy consente inoltre l'accesso alle risorse associate alle API E-mail e SMS e Voce di Amazon Pinpoint.

### Come creare la policy

1. Accedi AWS Management Console e apri la console IAM all'[indirizzo https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
3. Nell'editor delle politiche, seleziona JSON. Elimina qualsiasi codice JSON corrente nell'editor delle politiche in modo che sia vuoto. Copia e incolla il seguente codice JSON nell'editor delle

politiche, quindi nell'editor delle politiche sostituisci tutte le istanze di **123456789012** con il tuo ID. Account AWS

Il tuo Account AWS ID si trova nell'angolo in alto a destra della console oppure puoi usare la CLI, vedi [Trovare l'ID del tuo AWS account](#).

### Note

Per proteggere i dati nell'account Amazon Pinpoint, questa policy include solo le autorizzazioni che consentono la lettura, la creazione e la modifica delle risorse. Non include le autorizzazioni che consentono l'eliminazione delle risorse. È possibile modificare questa policy utilizzando l'editor visivo nella console IAM. Per ulteriori informazioni, consulta [Gestione di policy IAM](#) nella Guida per l'utente di IAM. Puoi anche utilizzare l'[CreatePolicyVersion](#) operazione nell'API IAM per aggiornare questa policy. È inoltre importante tenere presente che questa policy include autorizzazioni che consentono di interagire con i servizi ses e sms-voice, oltre al servizio `mobiletargeting`. Le autorizzazioni ses e sms-voice consentono di interagire rispettivamente con le API E-mail e SMS e Voce di Amazon Pinpoint. Le autorizzazioni `mobiletargeting` consentono di interagire con l'API Amazon Pinpoint.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Update*",
        "mobiletargeting:Get*",
        "mobiletargeting:Send*",
        "mobiletargeting:Put*",
        "mobiletargeting:Create*"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:*:123456789012:apps/*",
        "arn:aws:mobiletargeting:*:123456789012:apps/*/campaigns/*",
        "arn:aws:mobiletargeting:*:123456789012:apps/*/segments/*"
      ]
    }
  ],
}
```

```

    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:TagResource",
        "mobiletargeting:PhoneNumberValidate",
        "mobiletargeting:ListTagsForResource",
        "mobiletargeting:CreateApp"
      ],
      "Resource": "arn:aws:mobiletargeting:*:123456789012:*"
    },
    {
      "Sid": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "ses:TagResource",
        "ses:Send*",
        "ses:Create*",
        "ses:Get*",
        "ses:List*",
        "ses:Put*",
        "ses:Update*",
        "sms-voice:SendVoiceMessage",
        "sms-voice:List*",
        "sms-voice:Create*",
        "sms-voice:Get*",
        "sms-voice:Update*"
      ],
      "Resource": "*"
    }
  ]
}

```

Seleziona Successivo.

4. Per Nome della politica, inserisci un nome per la politica, ad esempio **PostmanAccessPolicy**. Scegli Crea policy.
5. (Facoltativo) Puoi aggiungere tag alla policy selezionando Aggiungi tag.
6. Scegli Prossimo: Rivedi.

## Crea un utente IAM

### Warning

Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per contribuire a mitigare questo rischio, ti consigliamo di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

Dopo aver creato la policy, è possibile creare un utente a cui collegarla. Quando crei l'utente, IAM fornisce un set di credenziali da utilizzare per consentire a Postman di eseguire operazioni dell'API Amazon Pinpoint.

Per creare l'utente

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Sulla console IAM, nel pannello di navigazione, scegli Utenti, quindi scegli Crea utenti.
3. In Dettagli utente, nel campo Nome utente, immetti un nome che identifichi l'account utente, come **PostmanUser**. Quindi scegli Successivo.
4. In Imposta autorizzazioni, nel campo Opzioni delle autorizzazioni, scegli Allega direttamente le policy.
5. In Politiche di autorizzazione, scegli la policy (**PostmanAccessPolicy**) che hai creato in [Crea una policy IAM](#). Quindi scegli Successivo.
6. Nella pagina Rivedi e crea, puoi facoltativamente aggiungere tag per semplificare l'identificazione dell'utente. Per ulteriori informazioni sull'utilizzo di tag, consulta la sezione [Tagging delle risorse IAM](#) nella Guida per l'utente di IAM.
7. Quando si è pronti per creare l'utente, scegliere Create user (Crea utente).

Crea chiavi di accesso

### Warning

Questo scenario richiede agli utenti IAM accesso programmatico e credenziali a lungo termine, il che presenta un rischio per la sicurezza. Per contribuire a mitigare questo rischio, ti consigliamo di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e

di rimuoverli quando non sono più necessari. Le chiavi di accesso possono essere aggiornate se necessario. Per ulteriori informazioni, consulta [Updating access keys](#) nella IAM User Guide.

IAM fornisce un set di credenziali che puoi utilizzare per consentire a Postman di eseguire operazioni dell'API Amazon Pinpoint.

Per creare l'utente

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione della console IAM, scegli Utenti. Seleziona l'utente (**PostmanUser**) creato in [Crea un utente IAM](#), quindi seleziona la scheda Credenziali di sicurezza.
3. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso.
4. Nella pagina Access key best practice e alternative, seleziona Applicazione in esecuzione all'esterno AWS.

Quindi scegli Successivo.

5. (Facoltativo) Puoi aggiungere un tag descrittivo alla policy.
6. Selezionare Create access key (Crea chiave di accesso).
7. Nella pagina Recupera le chiavi di accesso, copia le credenziali visualizzate nelle colonne Chiave di accesso e Chiave di accesso segreta.

#### Note

È necessario fornire sia l'ID chiave di accesso sia la chiave di accesso segreta più avanti in questo tutorial. Questa è l'unica volta in cui è possibile visualizzare la chiave di accesso segreta. Ti consigliamo di copiarla e conservarla in un luogo sicuro.

8. Dopo aver salvato entrambe le chiavi, scegli Fatto.

## Fase 2: configurazione di Postman

Dopo aver creato un account utente in grado di accedere all'API Amazon Pinpoint, puoi configurare Postman. In questa sezione, crea uno o più ambienti in Postman. Quindi, importa una raccolta contenente modello di richiesta di ciascuna delle operazioni nell'API Amazon Pinpoint.

## Crea un'area di lavoro Postman

In Postman, un workspace è un container organizzativo per progetti e ambienti. In questa sezione, dovrai creare almeno un workspace da utilizzare con Amazon Pinpoint.

### Creare un'area di lavoro

In Postman, scegli Altre azioni, quindi seleziona File, Nuovo.

1. Nella finestra Crea nuovo, scegli Workspace.
2. Inserisci un nome, un riepilogo e imposta la visibilità su Personale. Quindi, scegli Crea workspace.

### Crea ambienti Postman

In Postman, un ambiente è un set di variabili che vengono memorizzate come coppie chiave-valore. È possibile utilizzare gli ambienti per modificare la configurazione delle richieste effettuate tramite Postman, senza dover modificare le richieste API.

In questa sezione, dovrai creare almeno un ambiente da utilizzare con Amazon Pinpoint. Ogni ambiente creato contiene un set di variabili specifiche per l'account in un'unica Regione AWS. Se utilizzi le procedure di questa sezione per creare più di un ambiente, puoi passare da una regione a un'altra selezionando un ambiente diverso nel menu Ambiente in Postman.

### Per creare un ambiente

1. In Postman, scegli il menu Altre azioni, quindi seleziona File, Nuovo.
2. Nella finestra Create New (Crea nuovo), scegliere Environment (Ambiente).
3. Nella finestra MANAGE ENVIRONMENTS (GESTISCI AMBIENTI), per Environment Name (Nome ambiente), inserire **Amazon Pinpoint - *Region Name***. Sostituire *Region Name* (*Nome regione*) con uno dei seguenti valori:
  - Stati Uniti orientali (Virginia settentrionale)
  - US West (Oregon)
  - Asia Pacifico (Mumbai)
  - Asia Pacifico (Sydney)
  - Europa (Francoforte)
  - Europa (Irlanda)

**Note**

Come minimo, devi creare solo un ambiente per singolo Regione AWS e che Regione AWS deve contenere un progetto. Se non hai creato un progetto in uno dei progetti elencati in precedenza Regioni AWS, consulta [Creazione di un progetto Amazon Pinpoint con supporto via e-mail](#) nella Guida per l'utente di Amazon Pinpoint.

4. Creare sei nuove variabili: `endpoint`, `region`, `serviceName`, `accountId`, `accessKey` e `secretAccessKey`. Utilizza la tabella seguente per determinare il valore da inserire nelle colonne Valore iniziale e Valore corrente per ogni variabile.

Regione	Variabile	Valore iniziale/Valore corrente
Stati Uniti orientali (Virginia settentrionale)	<code>endpoint</code>	<b>pinpoint.us-east-1 .amazonaws.com</b>
	<code>region</code>	<b>us-east-1</b>
	<code>serviceName</code>	<b>mobiletargeting</b>
	<code>accountId</code>	(l'ID del tuo AWS account)
	<code>accessKey</code>	(ID della chiave di accesso IAM)
	<code>secretAccessKey</code>	(chiave di accesso segreta IAM)
US West (Oregon)	<code>endpoint</code>	<b>pinpoint.us-west-2 .amazonaws.com</b>
	<code>region</code>	<b>us-west-2</b>
	<code>serviceName</code>	<b>mobiletargeting</b>

Regione	Variabile	Valore iniziale/Valore corrente
	accountId	(l'ID AWS del tuo account)
	accessKey	(ID della chiave di accesso IAM)
	secretAccessKey	(chiave di accesso segreta IAM)
Asia Pacifico (Mumbai)	endpoint	<b>pinpoint.ap-south-1.amazonaws.com</b>
	region	<b>ap-south-1</b>
	serviceName	<b>mobiletargeting</b>
	accountId	(l'ID AWS del tuo account)
	accessKey	(ID della chiave di accesso IAM)
	secretAccessKey	(chiave di accesso segreta IAM)
Asia Pacifico (Sydney)	endpoint	<b>pinpoint.ap-southeast-2.amazonaws.com</b>
	region	<b>ap-southeast-2</b>
	serviceName	<b>mobiletargeting</b>
	accountId	(l'ID AWS del tuo account)

Regione	Variabile	Valore iniziale/Valore corrente
	accessKey	(ID della chiave di accesso IAM)
	secretAccessKey	(chiave di accesso segreta IAM)
Europa (Francoforte)	endpoint	<b>pinpoint.eu-central-1-1.amazonaws.com</b>
	region	<b>eu-central-1</b>
	serviceName	<b>mobiletargeting</b>
	accountId	(l'ID AWS del tuo account)
	accessKey	(ID della chiave di accesso IAM)
	secretAccessKey	(chiave di accesso segreta IAM)
Europa (Irlanda)	endpoint	<b>pinpoint.eu-west-1.amazonaws.com</b>
	region	<b>eu-west-1</b>
	serviceName	<b>mobiletargeting</b>
	accountId	(l'ID AWS del tuo account)
	accessKey	(ID della chiave di accesso IAM)

Regione	Variabile	Valore iniziale/Valore corrente
	secretAccessKey	(chiave di accesso segreta IAM)

Una volta create queste variabili, l'aspetto della finestra MANAGE ENVIRONMENTS (GESTISCI AMBIENTI) è simile a quello dell'esempio mostrato nella seguente immagine.

US East (N. Virginia) Fork | 0 Save Share ...

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	endpoint	default ▾	pinpoint.us-east-1.amazonaws.com	pinpoint.us-east-1.amazonaws.com			🗑️ ...
<input checked="" type="checkbox"/>	region	default ▾	us-east-1	us-east-1			
<input checked="" type="checkbox"/>	serviceName	default ▾	mobiletargeting	mobiletargeting			
<input checked="" type="checkbox"/>	accountId	default ▾	123456789012	123456789012			
<input checked="" type="checkbox"/>	accessKey	default ▾	AKIAIOSFODNN7EXAMPLE	AKIAIOSFODNN7EXAMPLE			
<input checked="" type="checkbox"/>	secretAccessKey	default ▾	wJairXUtnFEMI/K7MDENG/bPxrFiCY...	wJairXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY			

Al termine, scegli Salva.

### ⚠ Important

Le chiavi di accesso mostrate nell'immagine precedente sono fittizie. Non condividere le chiavi di accesso IAM con altri utenti.

Postman include funzionalità che consentono di condividere ed esportare ambienti. Se utilizzi queste funzionalità, assicurati di non condividere l'ID chiave di accesso e la chiave di accesso segreta con chiunque non debba avere accesso a tali credenziali.

Per ulteriori informazioni, consultare la sezione [best practice IAM](#) nella Guida per l'utente IAM.

- (Facoltativo) Ripeti i passaggi da 1 a 4 per ogni altro ambiente che desideri creare.

### 📘 Tip

In Postman, è possibile creare tutti gli ambienti di cui si ha bisogno. Puoi utilizzare gli ambienti nei seguenti modi:

- Creare un ambiente separato per ogni regione in cui è necessario testare l'API Amazon Pinpoint.
- Creare ambienti associati a diversi Account AWS.
- Creare ambienti che utilizzano credenziali associate a un altro utente.

6. Una volta terminata la creazione di ambienti, procedere alla sezione successiva.

## Crea una raccolta Amazon Pinpoint in Postman

In Postman, una raccolta è un gruppo di richieste API. Le richieste in una raccolta sono in genere unite da uno scopo comune. In questa sezione, dovrai creare una nuova raccolta contenente un modello di richiesta per ciascuna operazione nell'API Amazon Pinpoint.

Per creare la raccolta Amazon Pinpoint

1. In Postman, scegli il menu Altre azioni, quindi seleziona File, Importa.
2. Nella finestra Importa, scegli Importa da link, quindi inserisci il seguente URL: [https://raw.githubusercontent.com/awsdocs/amazon-pinpoint-developer-guide/master/Amazon%20Pinpoint.postman\\_collection.json](https://raw.githubusercontent.com/awsdocs/amazon-pinpoint-developer-guide/master/Amazon%20Pinpoint.postman_collection.json).

Seleziona Importa. Postman importa la raccolta Amazon Pinpoint, contenente 120 richieste di esempio.

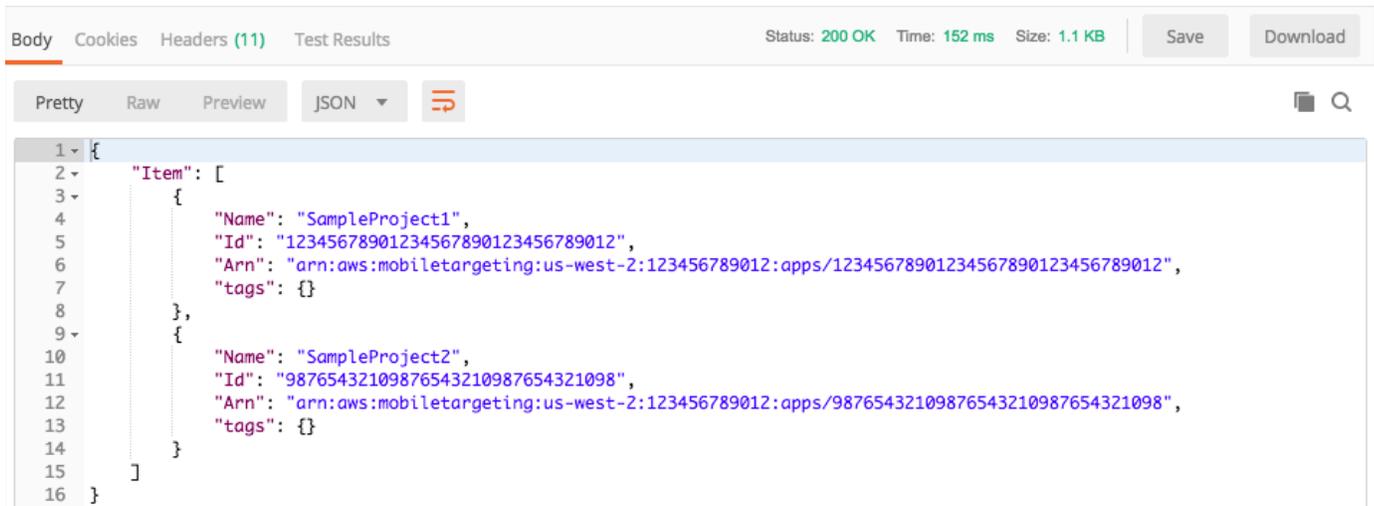
## Test della configurazione di Postman

Dopo aver importato la raccolta Amazon Pinpoint, è consigliabile eseguire un test rapido per verificare la corretta configurazione di tutti i componenti. È possibile provare la configurazione inviando una richiesta GetApps. Questa richiesta restituisce l'elenco di tutti i progetti esistenti nell'account Amazon Pinpoint nella Regione AWS attuale. Questa richiesta non necessita di configurazioni aggiuntive e per tale motivo rappresenta un valido modo per testare la configurazione.

Per testare la configurazione della raccolta Amazon Pinpoint

1. Nel riquadro di navigazione, seleziona Raccolte, espandi la raccolta Amazon Pinpoint e quindi espandi la cartella App.
2. Nell'GetAppselenco delle richieste, scegli.
3. Usa il selettore Ambiente per scegliere l'ambiente che hai creato in Ambienti [Create Postman](#).

4. Scegli **Invia**. Se la richiesta viene inviata correttamente, il riquadro di risposta mostra lo stato **200 OK**. Viene visualizzata una risposta simile all'esempio riportato nella seguente immagine.



```

1 {
2   "Item": [
3     {
4       "Name": "SampleProject1",
5       "Id": "12345678901234567890123456789012",
6       "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/12345678901234567890123456789012",
7       "tags": {}
8     },
9     {
10      "Name": "SampleProject2",
11      "Id": "98765432109876543210987654321098",
12      "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/98765432109876543210987654321098",
13      "tags": {}
14    }
15  ]
16 }

```

### Note

Se non hai creato alcun progetto nel, Regione AWS restituisce Amazon Pinpoint.

```
{ "Item": [] }
```

Questa risposta mostra l'elenco di tutti i progetti Amazon Pinpoint esistenti nell'account nella regione scelta al Passaggio 3.

## Risoluzione dei problemi

Quando invii la richiesta, è possibile che venga visualizzato un errore. Di seguito sono elencati i diversi errori comuni che possono verificarsi e le relative fasi per la risoluzione.

Messaggio di errore	Problema	Risoluzione
Could not get any response (Impossibile ottenere una risposta)	Non esiste alcun valore attuale per la variabile <code>{{endpoint}}</code> , che è impostata quando scegli un ambiente.	Utilizza il selettore di ambiente per scegliere un ambiente.
There was an error connecting to <code>https://{{endpoint}}/v1/apps</code> . (Si è		

Messaggio di errore	Problema	Risoluzione
verificato un errore durante la connessione a https://%7B%7Bendpoint%7D%7D/v1/apps.)		
Il token di sicurezza incluso nella richiesta non è valido.	Postman non era in grado di individuare il valore attuale dell'ID chiave di accesso o della chiave di accesso segreta.	Scegli l'icona a forma di ingranaggio accanto al selettore di ambiente e quindi scegli l'ambiente attuale. Assicurati che i valori <code>accessKey</code> e <code>secretAccessKey</code> vengano visualizzati in entrambe le colonne VALORE INIZIALE e VALORE CORRENTE e verifica di aver inserito le credenziali correttamente.
«Message»: «User: arn:aws:iam::123456789012:user / non PinpointPostmanUser è autorizzato a eseguire: mobiletargeting: on resource: arn:aws:mobiletargeting:us-west-2:123456789012: *» GetApps	La policy IAM associata all'utente non include le autorizzazioni appropriate.	<a href="#">Verifica che il tuo utente disponga delle autorizzazioni descritte in <u>Creare una policy IAM e di aver fornito le credenziali corrette quando hai creato l'ambiente nell'area di lavoro Create Postman.</u></a>

### Fase 3: invio di richieste aggiuntive

Al termine della configurazione e del test di Postman, è possibile iniziare a inviare richieste aggiuntive all'API Amazon Pinpoint. In questa sezione sono incluse informazioni che è necessario conoscere prima di iniziare a inviare le richieste. Include anche due richieste di esempio che descrivono come utilizzare la raccolta Amazon Pinpoint.

**⚠ Important**

Dopo aver completato le procedure di questa sezione, è possibile inviare richieste all'API Amazon Pinpoint. Tali richieste creano nuove risorse nell'account Amazon Pinpoint, modificano le risorse esistenti, inviano messaggi, modificano la configurazione dei progetti Amazon Pinpoint e utilizzare altre funzionalità di Amazon Pinpoint. Presta attenzione quando esegui queste richieste.

**Informazioni sugli esempi nella raccolta Postman di Amazon Pinpoint**

Prima di poterle utilizzare, è necessario configurare la maggior parte delle operazioni nel file di raccolta Postman di Amazon Pinpoint. Per le operazioni GET e DELETE, in genere occorre solo modificare le variabili impostate nella scheda Pre-request Script (Script prerichiesta).

**📘 Note**

Quando utilizzi la policy IAM mostrata in [Creare una policy IAM](#), non puoi eseguire nessuna delle DELETE richieste incluse in questa raccolta.

Ad esempio, l'operazione GetCampaign richiede l'indicazione di un `projectId` e di un `campaignId`. Nella scheda Pre-request Script (Script prerichiesta), sono presenti entrambi i valori, che vengono popolati con valori di esempio. È necessario eliminare i valori di esempio e sostituirli con quelli validi per il progetto o la campagna Amazon Pinpoint.

Di queste variabili, quella più comunemente usata è la variabile `projectId`. Il valore di questa variabile deve essere l'identificatore univoco del progetto a cui si applica la richiesta. Per ottenere l'elenco di questi identificatori dei progetti, è possibile fare riferimento alla risposta alla richiesta GetApps inviata nel passaggio precedente di questo tutorial. In quella risposta, il campo `Id` include l'identificatore univoco di un progetto. Per ulteriori informazioni sull'operazione GetApps e sul significato di ciascun campo nella risposta, consulta [App](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

**📘 Note**

In Amazon Pinpoint, il termine "progetto" equivale al termine "app" o "applicazione".

Per le operazioni POST e PUT, è necessario modificare anche il corpo della richiesta in modo da includere i valori da inviare all'API. Ad esempio, quando si invia una richiesta `CreateApp`, che è una richiesta POST, è necessario specificare un nome per il progetto in fase di creazione. È possibile modificare la richiesta nella scheda Body (Corpo). In questo esempio, il valore accanto a "Name" viene sostituito con il nome del progetto. Se si desidera aggiungere tag al progetto, è possibile specificarli nell'oggetto `tags`. O, se non si desidera aggiungere i tag, è possibile eliminare l'intero oggetto `tags`.

### Note

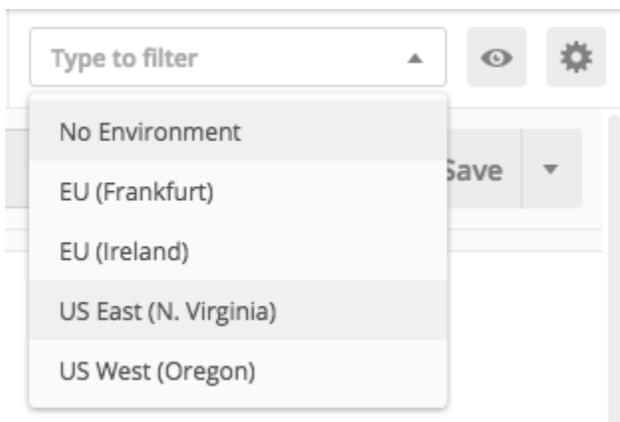
L'operazione `UntagResource` richiede anche l'indicazione di parametri URL. Tali parametri si possono specificare nella scheda Params (Parametri). Sostituire i valori nella colonna VALUE con i tag che si desidera eliminare per la risorsa specificata.

Richiesta di esempio: creazione di un progetto tramite l'operazione **CreateApp**.

Prima di creare segmenti e campagne in Amazon Pinpoint, è prima necessario creare un progetto. In Amazon Pinpoint, un progetto è composto da segmenti, campagne, configurazioni e dati che condividono uno scopo comune. Ad esempio, è possibile utilizzare un progetto per contenere tutti i contenuti relativi a una particolare app o a un determinato brand o iniziativa di marketing. Quando si aggiungono le informazioni sui clienti ad Amazon Pinpoint, quelle informazioni sono associate a un progetto.

Per creare un progetto inviando una richiesta `CreateApp` API

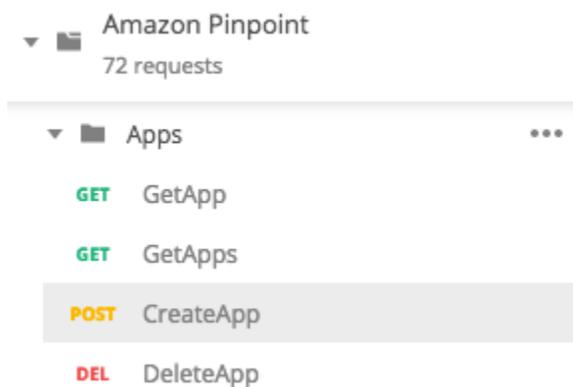
1. Nel menu Ambienti, scegli Regione AWS quello in cui vuoi creare il progetto.



In questo esempio, Postman è stato configurato in modo tale che il menu Environments (Ambienti) mostri le seguenti quattro opzioni:

- Stati Uniti orientali (Virginia settentrionale)
- US West (Oregon)
- Europa (Francoforte)
- Europa (Irlanda)

2. Nella cartella App, scegli l'CreateAppoperazione>.



La cartella App nella raccolta Postman di Amazon Pinpoint viene espansa e mostra le seguenti richieste:

- GetApp
- GetApps
- CreateApp
- DeleteApp

3. Nella scheda Body (Corpo), accanto a "Name", sostituire il valore del segnaposto ("string") con un nome della campagna, ad esempio **"MySampleProject"**.
4. Elimina la virgola dopo il nome della campagna, quindi elimina l'intero oggetto tags nelle righe da 3 a 5. Al termine, la richiesta deve essere simile all'esempio illustrato nel seguente frammento di codice.

```
{
  "Name": "MySampleProject"
}
```

Postman è configurato per inviare la richiesta come un payload JSON raw.

5. Scegli **Invia**. Se la campagna viene creata correttamente, il riquadro di risposta mostra lo stato **201 Created**.

```
{
  "Name": "MySampleProject"
  "Id": "12345678901234567890123456789012",
  "Arn": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/12345678901234567890123456789012",
  "tags": {}
}
```

### Esempio: invio di un'e-mail tramite l'operazione **SendMessage**

È molto comune utilizzare l'API **SendMessage** di Amazon Pinpoint per inviare messaggi transazionali. Uno dei vantaggi dell'invio di messaggi tramite l'API **SendMessage** rispetto alla creazione delle campagne è la possibilità di inviare messaggi a qualsiasi indirizzo, come indirizzi e-mail, numeri di telefono o token di dispositivi. Non è necessario che l'indirizzo a cui vengono inviati i messaggi sia già presente nell'account Amazon Pinpoint. Confrontiamo questa opzione con l'invio di messaggi tramite la creazione di campagne. Prima di inviare una campagna in Amazon Pinpoint, devi aggiungere endpoint all'account Amazon Pinpoint, creare segmenti e quindi creare ed eseguire la campagna.

L'esempio in questa sezione mostra come inviare un messaggio e-mail transazionale direttamente a un indirizzo e-mail specifico. È possibile modificare questa richiesta per inviare messaggi tramite altri canali, ad esempio SMS, push su dispositivi mobili o voce.

Per inviare un messaggio e-mail inviando una richiesta **SendMessage**

1. Verifica che il canale e-mail sia abilitato per il progetto e che l'indirizzo e-mail o il dominio che desideri utilizzare per inviare e ricevere il messaggio sia configurato. Per ulteriori informazioni, consulta [Abilitazione e disabilitazione del canale e-mail](#) e [Verifica delle identità e-mail](#) nella Guida per l'utente di Amazon Pinpoint.

**Note**

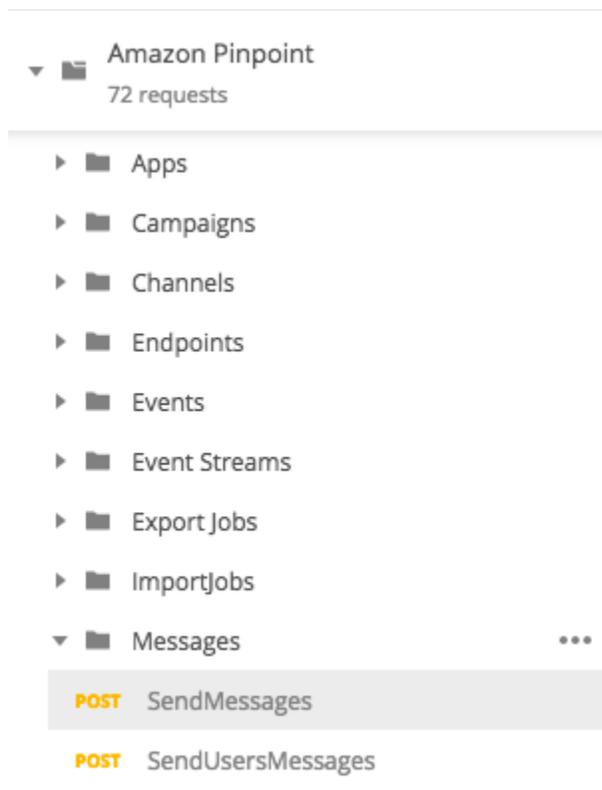
Per completare la procedura riportata in questa sezione, è necessario verificare un indirizzo e-mail.

2. Nel menu Ambienti, scegli il Regione AWS dispositivo da cui desideri inviare il messaggio.

In questo esempio, Postman è stato configurato in modo tale che il menu Environments (Ambienti) mostri le seguenti quattro opzioni:

- Stati Uniti orientali (Virginia settentrionale)
- US West (Oregon)
- Europa (Francoforte)
- Europa (Irlanda)

3. Nella cartella Messaggi, scegli l'SendMessageoperazione.



4. Nella scheda Pre-request Script (Script pre-richiesta), sostituire il valore della variabile `projectId` con l'ID di un progetto già esistente nella regione selezionata nella Fase 2 di questa sezione.

5. Nella scheda Body (Corpo), eliminare la richiesta di esempio illustrata nell'editor della richiesta. Incolla il codice seguente:

```
{
  "MessageConfiguration":{
    "EmailMessage":{
      "FromAddress":"sender@example.com",
      "SimpleEmail":{
        "Subject":{
          "Data":"Sample Amazon Pinpoint message"
        },
        "HtmlPart":{
          "Data":"<h1>Test message</h1><p>This is a sample message sent
from <a href=\"https://aws.amazon.com/pinpoint\">Amazon Pinpoint</a> using the
SendMessages API.</p>"
        },
        "TextPart":{
          "Data":"This is a sample message sent from Amazon Pinpoint
using the SendMessages API."
        }
      }
    }
  },
  "Addresses":{
    "recipient@example.com": {
      "ChannelType": "EMAIL"
    }
  }
}
```

6. Nel codice precedente, sostituire *sender@example.com* con l'indirizzo e-mail verificato. Sostituisci *recipient@example.com* con l'indirizzo e-mail verificato a cui desideri inviare il messaggio.

#### Note

Se l'account è ancora nell'ambiente di sperimentazione (sandbox) per e-mail di Amazon Pinpoint, è possibile inviare e-mail solo agli indirizzi o domini verificati nell'account Amazon Pinpoint. Per ulteriori informazioni su come far rimuovere l'account dall'ambiente di sperimentazione (sandbox), consulta la sezione relativa alla [richiesta dell'accesso](#)

[all'ambiente di produzione per il canale e-mail](#) nella Guida per l'utente di Amazon Pinpoint.

7. Scegli **Invia**. Se il messaggio viene inviato correttamente, il riquadro di risposta mostra lo stato **200 OK**.

```
{
  "ApplicationId": "12345678901234567890123456789012",
  "RequestId": "<sampleValue>",
  "Result": {
    "recipient@example.com": {
      "DeliveryStatus": "SUCCESSFUL",
      "StatusCode": 200,
      "StatusMessage": "<sampleValue>",
      "MessageId": "<sampleValue>"
    }
  }
}
```

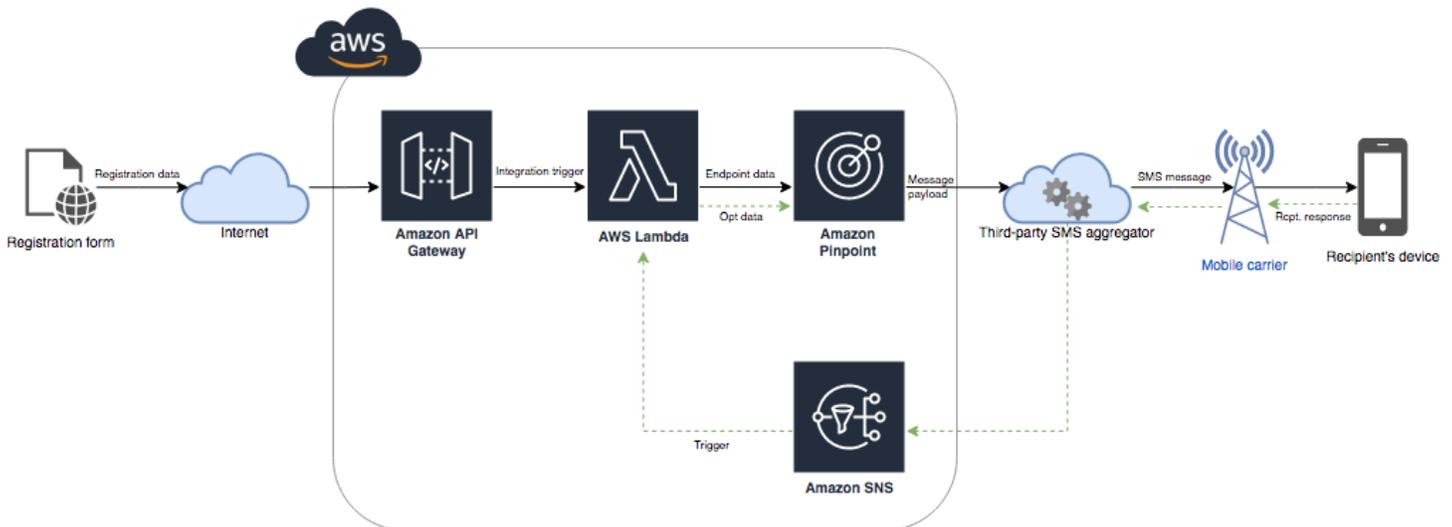
## Tutorial: configurazione di un sistema di registrazione SMS

I messaggi SMS (messaggi di testo) sono un ottimo modo per inviare messaggi in tempo reale ai clienti. Al giorno d'oggi molte persone hanno sempre il telefono a portata di mano. Inoltre, i messaggi SMS catturano l'attenzione della gente in misura maggiore rispetto alle notifiche push, alle e-mail o alle telefonate.

Un metodo comune per acquisire i numeri di smartphone dei clienti consiste nell'utilizzare un modulo Web. Dopo aver verificato il numero di telefono del cliente e aver ottenuto la conferma della sottoscrizione, puoi iniziare a inviargli messaggi SMS promozionali, transazionali e informativi.

Questo tutorial illustra come configurare un modulo Web per acquisire le informazioni di contatto dei clienti. Il modulo Web invia queste informazioni ad Amazon Pinpoint. Quindi, Amazon Pinpoint verifica la validità del numero di telefono e acquisisce altri metadati correlati al numero. Successivamente, Amazon Pinpoint invia un messaggio al cliente per chiedere conferma della sottoscrizione. Una volta che il cliente conferma la sottoscrizione, Amazon Pinpoint lo abilita mediante opt-in alla ricezione dei messaggi.

Il seguente diagramma di architettura mostra il flusso di dati in questa soluzione.



## Informazioni sulla doppia attivazione

In questo tutorial viene illustrato come configurare un sistema di doppio opt-in in Amazon Pinpoint basato sulla messaggistica SMS bidirezionale.

In un sistema di doppia attivazione SMS, il cliente ti fornisce il suo numero di telefono inviandolo in un modulo Web o all'interno della tua app. Quando ricevi la richiesta del cliente, crei un nuovo endpoint in Amazon Pinpoint. Il nuovo endpoint deve essere escluso dalle comunicazioni. Quindi, invii un messaggio a quel numero di telefono. Nel messaggio richiedi al destinatario la conferma della sottoscrizione con una parola o frase specifica (ad esempio "Sì" o "Conferma"). Se il cliente risponde al messaggio con la parola o frase specificata, modifichi lo stato dell'endpoint in attivato. In caso contrario, se il cliente non risponde o risponde con una parola o frase diversa, puoi lasciare l'endpoint disattivato.

## Informazioni su questa soluzione

Questa sezione contiene informazioni sulla soluzione che stai creando in questo tutorial.

### Destinatari principali

Questo tutorial è rivolto a un pubblico di sviluppatori e di implementatori di sistema. Non è necessario avere familiarità con Amazon Pinpoint per completare le fasi descritte in questo tutorial. Tuttavia, è consigliabile avere una certa dimestichezza con la gestione delle policy IAM, la creazione di funzioni Lambda in Node.js e l'implementazione di contenuti Web.

### Caratteristiche utilizzate

Questo tutorial include esempi di utilizzo delle seguenti funzionalità di Amazon Pinpoint:

- Invio di SMS transazionali
- Acquisizione di informazioni sui numeri di telefono utilizzando la funzione di convalida del numero di telefono
- Ricezione di messaggi SMS in entrata utilizzando la messaggistica SMS bidirezionale
- Creazione di segmenti dinamici
- Creazione di campagne
- Interazione con l'API Amazon Pinpoint utilizzando AWS Lambda

### Tempo richiesto

Il completamento di questo tutorial richiede circa un'ora. Dopo aver implementato questa soluzione, esistono fasi aggiuntive che puoi utilizzare per adeguarla al tuo caso d'uso specifico.

### Limitazioni regionali

Questo tutorial richiede il noleggio di un codice lungo tramite la console Amazon Pinpoint. Puoi utilizzare la console Amazon Pinpoint per noleggiare codici lunghi dedicati ubicati in vari paesi. Tuttavia, per inviare messaggi SMS puoi utilizzare solo codici lunghi con sede in Canada. (Per inviare messaggi vocali è possibile utilizzare codici lunghi con sede in altri paesi e regioni).

Abbiamo sviluppato i codici di esempio di questo tutorial tenendo presente questa limitazione. Ad esempio, i codici di esempio presuppongono che il numero di telefono sia sempre di 10 cifre e che il codice paese sia 1. Se implementi la soluzione in paesi o regioni che non siano Stati Uniti o Canada, devi modificare i codici di esempio in modo appropriato.

### Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, implementando questa soluzione potresti incorrere nei seguenti costi:

- Costo di noleggio del codice lungo: per completare questo tutorial, devi noleggiare un codice lungo. I codici lunghi con sede in Canada costano 1,00 USD al mese.
- Utilizzo della convalida del numero di telefono: la soluzione in questo tutorial utilizza la funzionalità di convalida del numero di telefono di Amazon Pinpoint per verificare che ogni numero ricevuto sia valido e nel formato corretto, nonché per ottenere ulteriori informazioni sul numero di telefono. Per ogni richiesta di convalida del numero di telefono ti vengono addebitati 0,006 USD.

- **Costi per l'invio dei messaggi:** la soluzione in questo tutorial invia messaggi SMS in uscita. Ti viene addebitato ogni messaggio inviato tramite Amazon Pinpoint. Il prezzo addebitato per ogni messaggio dipende dal paese o dalla regione del destinatario. Se invii messaggi a destinatari negli Stati Uniti (esclusi i territori non incorporati negli Stati Uniti), ti vengono addebitati 0,00645 USD a messaggio. Se invii messaggi a destinatari in Canada, l'addebito è nell'intervallo 0,00109–0,02 USD, a seconda dell'operatore e della posizione del destinatario.
- **Costi per la ricezione dei messaggi:** questa soluzione riceve ed elabora anche i messaggi SMS in entrata. Ti viene addebitato ogni messaggio in entrata inviato ai numeri di telefono associati all'account Amazon Pinpoint. Il prezzo dipende da dove si trova il telefono che riceve i messaggi. Se il numero ricevente è negli Stati Uniti (esclusi i territori non incorporati negli Stati Uniti), ti vengono addebitati 0,0075 USD per ogni messaggio in entrata. Se il numero è in Canada, ti vengono addebitati 0,00155 USD per ogni messaggio in entrata.
- **Utilizzo di Lambda:** questa soluzione utilizza due funzioni Lambda che interagiscono con l'API Amazon Pinpoint. Quando chiami una funzione Lambda, i costi addebitati dipendono dal numero di richieste di funzioni, dal tempo necessario all'esecuzione del codice e dalla quantità di memoria utilizzata dalla funzione. In questo tutorial, le funzioni utilizzano una quantità ridotta di memoria e di solito vengono eseguite per 1-3 secondi. L'utilizzo totale o parziale della soluzione può rientrare nei limiti del piano di utilizzo gratuito di Lambda. Per ulteriori informazioni, consulta [Prezzi di Lambda](#).
- **Utilizzo di Gateway API:** in questa soluzione, il modulo Web chiama un'API gestita da Gateway API. Per ogni milione di chiamate ad API Gateway, paghi da 3,50 a 3,70 USD, a seconda della regione in cui AWS utilizzi Amazon Pinpoint. Per ulteriori informazioni, consulta [Prezzi di Gateway Amazon API](#).
- **Costi per l'hosting Web:** questa soluzione include un modulo Web che devi ospitare nel tuo sito Web. Il prezzo per l'hosting di questi contenuti dipende dal provider dell'hosting Web.

#### Note

Tutti i prezzi indicati in questo elenco sono in dollari USA (USD).

Successivo: [Prerequisiti](#)

## Prerequisiti

Prima di iniziare questo tutorial, è necessario completare i seguenti prerequisiti:

- Devi disporre di un account AWS. Per creare un account AWS, vai su <https://console.aws.amazon.com/> e scegli Crea un nuovo account AWS.
- L'account che utilizzi per accedere alla AWS Management Console deve essere in grado di eseguire le attività seguenti:
  - Creazione di nuovi ruoli e policy IAM
  - Creazione di nuovi progetti Amazon Pinpoint
  - Creazione di nuove funzioni Lambda
  - Creazione di nuove API in Gateway API
- È necessario disporre di un metodo per l'hosting di pagine Web e dovresti sapere come pubblicare le pagine Web. Sebbene sia possibile utilizzare i servizi AWS per l'hosting di pagine Web, non è necessario.

 Tip

Per ulteriori informazioni sull'hosting di pagine Web utilizzando i servizi AWS, consulta [Hosting di un sito Web statico](#).

Successivo: [Configurazione di Amazon Pinpoint](#)

## Passaggio 1: configurare Amazon Pinpoint

Il primo passaggio di implementazione di questa soluzione prevede la configurazione di Amazon Pinpoint. In questa sezione effettuerai le operazioni seguenti:

- Creare un progetto Amazon Pinpoint
- Abilitare il canale SMS e noleggiare un numero di telefono
- Configurare la messaggistica SMS bidirezionale

Prima di iniziare con questo tutorial, è necessario verificare i [prerequisiti](#).

### Creare un progetto Amazon Pinpoint

Per iniziare, è necessario creare un progetto Amazon Pinpoint. In Amazon Pinpoint, un progetto è composto da segmenti, campagne, configurazioni e dati che condividono uno scopo comune. Ad esempio, è possibile utilizzare un progetto per contenere tutti i contenuti relativi a una particolare app

o a un determinato brand o iniziativa di marketing. Quando si aggiungono le informazioni sui clienti ad Amazon Pinpoint, quelle informazioni sono associate a un progetto.

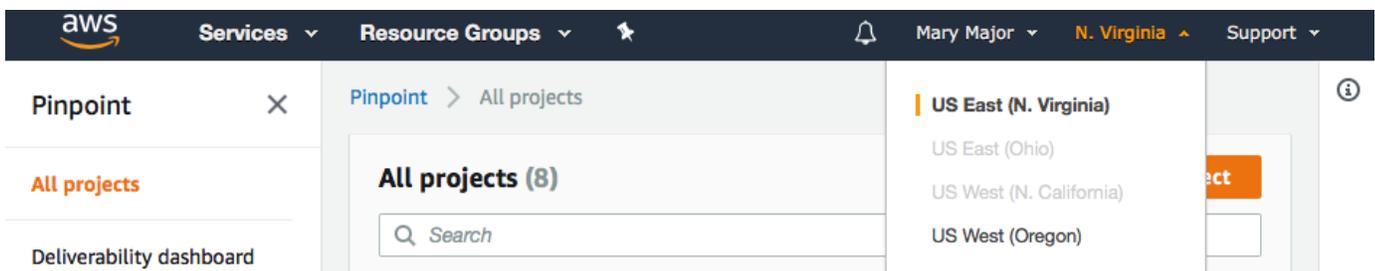
I passi necessari per la creazione di un nuovo progetto variano a seconda che in precedenza sia stato creato un progetto in Amazon Pinpoint o meno.

### Creazione di un progetto (nuovi utenti Amazon Pinpoint)

Questi passaggi descrivono il processo di creazione di un nuovo progetto Amazon Pinpoint se non hai mai creato un progetto nella regione corrente AWS .

Per creare un progetto

1. [Accedi AWS Management Console e apri la console Amazon Pinpoint all'indirizzo https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
2. Usa il selettore della regione per scegliere la AWS regione che desideri utilizzare, come mostrato nell'immagine seguente. Se non si è sicuri, scegliere la regione più prossima.



3. In Get started (Inizia), nel campo Name (Nome), inserire un nome per la campagna (come **SMSRegistration**), quindi selezionare Create project (Crea progetto)..
4. Nella pagina Configure features (Configura caratteristiche), selezionare Skip this step (Salta questa fase).
5. Nel riquadro di navigazione, selezionare All projects (Tutti i progetti).
6. Nella pagina All projects (Tutti i progetti), accanto al progetto appena creato, copiare il valore che compare nella colonna Project ID (ID progetto).

#### Tip

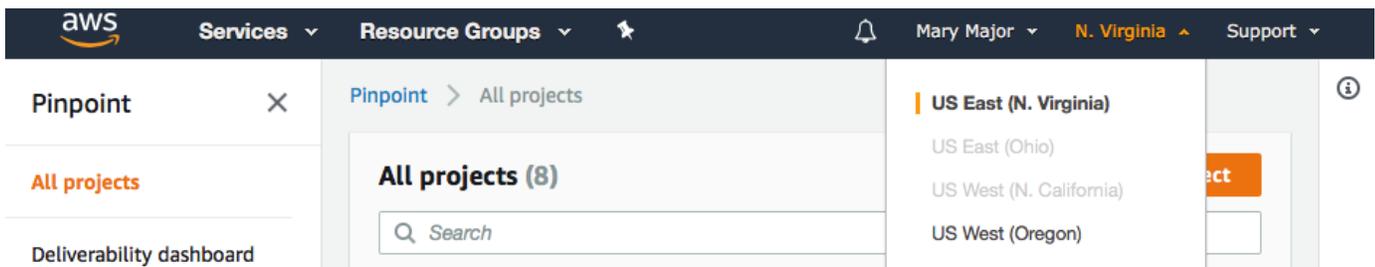
Questo ID servirà in pochi altri punti di questo tutorial. Conservare l'ID progetto in un percorso sicuro per poterlo copiare in un secondo momento.

## Creazione di un progetto (utenti Amazon Pinpoint esistenti)

Questi passaggi descrivono il processo di creazione di un nuovo progetto Amazon Pinpoint se hai già creato progetti nella regione corrente AWS .

Per creare un progetto

1. [Accedi AWS Management Console e apri la console Amazon Pinpoint all'indirizzo https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
2. Usa il selettore della regione per scegliere la AWS regione che desideri utilizzare, come mostrato nell'immagine seguente. Se non si è sicuri, scegliere la regione più prossima.



3. Nella pagina All projects (Tutti i progetti) scegliere Create a project (Crea un progetto).
4. Nella finestra Create a project (Crea un progetto), nel campo Project name (Nome progetto), immettere un nome per il progetto (come **SMSRegistration**). Scegli Crea.
5. Nella pagina Configure features (Configura caratteristiche), selezionare Skip this step (Salta questa fase).
6. Nel riquadro di navigazione, selezionare All projects (Tutti i progetti).
7. Nella pagina All projects (Tutti i progetti), accanto al progetto appena creato, copiare il valore che compare nella colonna Project ID (ID progetto).

### Tip

Questo ID servirà in pochi altri punti di questo tutorial. Conservare l'ID progetto in un percorso sicuro per poterlo copiare in un secondo momento.

## Ottenere un numero di telefono dedicato

### Note

Amazon Pinpoint ha aggiornato la documentazione relativa alla guida per l'utente. Per ottenere le informazioni più recenti su come creare, configurare e gestire le risorse vocali e SMS di Amazon Pinpoint, consulta la nuova [Guida per l'utente per Amazon Pinpoint SMS](#).

Dopo aver creato un progetto, è possibile iniziare a configurare le caratteristiche all'interno del progetto. In questa sezione, puoi abilitare il canale SMS e ottenere un numero di telefono dedicato da utilizzare per l'invio di messaggi SMS.

### Note

Questa sezione presuppone che tu stia noleggiando un numero di telefono 10DLC statunitense dopo avere registrato il marchio e la campagna, un numero verde statunitense o un codice lungo canadese. Se segui le procedure di questa sezione, ma scegli un paese diverso da Stati Uniti o Canada, non potrai utilizzare tale numero per inviare messaggi SMS. Per ulteriori informazioni sul leasing di codici lunghi compatibili con SMS in paesi diversi dagli Stati Uniti o dal Canada, consulta [Regioni e paesi supportati \(canale SMS\)](#) nella Guida per l'utente per Amazon Pinpoint SMS.

Per abilitare il canale SMS utilizzando la console Amazon Pinpoint, attieniti alla seguente procedura:

### Abilita il canale SMS

1. [Accedi AWS Management Console e apri la console Amazon Pinpoint all'indirizzo https://console.aws.amazon.com/pinpoint/](https://console.aws.amazon.com/pinpoint/).
2. Nel riquadro di navigazione, in Settings (Impostazioni), scegliere SMS and voice (Messaggi SMS e vocali).
3. Accanto a SMS settings (Impostazioni SMS), scegliere Edit (Modifica).
4. In General settings (Impostazioni generali), scegliere Enable the SMS channel for this project (Abilita il canale SMS per questo progetto), quindi scegliere Save changes (Salva modifiche).

Per richiedere un numero di telefono tramite la console Amazon Pinpoint SMS, attieniti alla seguente procedura:

### Richiesta di un numero di telefono (Console)

1. Apri la console Amazon Pinpoint SMS all'indirizzo <https://console.aws.amazon.com/sms-voice/>.

#### Note

Assicurati di richiedere un numero di telefono nello stesso indirizzo in Regione AWS cui hai creato il tuo progetto Amazon Pinpoint.

2. Nel riquadro di navigazione, in Configurazioni, scegli Numeri di telefono, quindi Origine della richiesta.
3. Nella pagina Seleziona paese per Paese di destinazione del messaggio scegli Stati Uniti o Canada. Seleziona Successivo.
4. Nella sezione Caso d'uso della messaggistica inserisci quanto segue:
  - In Funzionalità numeriche, scegli SMS

#### Important

Le funzionalità per SMS e voce non possono essere modificate una volta acquistato il numero di telefono.

- Per Messaggistica bidirezionale scegli Sì.
5. Seleziona Successivo.
  6. In Seleziona il tipo di origine scegli Codice lungo o 10DLC.

Se scegli 10DLC e hai già una campagna registrata, puoi scegliere la campagna da Associa a campagna registrata.
  7. Seleziona Successivo.
  8. In Verifica e richiedi puoi verificare e modificare la richiesta prima di inviarla. Scegli Richiedi.
  9. A seconda del tipo di numero di telefono richiesto, potrebbe apparire una finestra Registrazione obbligatoria. Il tuo numero di telefono è associato a questa registrazione e non puoi inviare messaggi finché la registrazione non viene approvata. Per ulteriori informazioni sui requisiti di registrazione, consulta [Registrazioni](#).

- a. Per Nome del modulo di registrazione, inserisci un nome descrittivo.
- b. Scegli Inizia la registrazione per completare la registrazione del numero di telefono o Registrati più tardi.

**⚠ Important**

Il tuo numero di telefono non può inviare messaggi finché la registrazione non viene approvata.

Ti verrà comunque addebitato il canone di locazione mensile ricorrente per il numero di telefono indipendentemente dallo stato della registrazione. Per ulteriori informazioni sui requisiti di registrazione, consulta [Registrazioni](#).

## Abilitazione della messaggistica SMS bidirezionale

Ora che disponi di un numero di telefono dedicato, puoi configurare la messaggistica SMS bidirezionale. Tale abilitazione consente ai clienti di rispondere ai messaggi SMS che hai inviato loro. In questa soluzione, puoi usare la messaggistica SMS bidirezionale per offrire ai clienti un modo per confermare il desiderio di iscriversi al programma SMS.

Per abilitare gli SMS bidirezionali utilizzando la console Amazon Pinpoint SMS, attieniti alla seguente procedura:

### Abilitazione della messaggistica SMS bidirezionale

1. Apri la console Amazon Pinpoint SMS all'indirizzo <https://console.aws.amazon.com/sms-voice/>.
2. Nel pannello di navigazione, in Configurazioni, scegli Numeri di telefono.
3. Nella pagina Numeri di telefono scegli un numero di telefono.
4. Nella scheda SMS bidirezionali, scegli il pulsante Modifica impostazioni.
5. Nella pagina Modifica impostazioni scegli Abilita messaggio bidirezionale.
6. Per Tipo di destinazione, scegli Amazon SNS.
  - Nuovo argomento Amazon SNS: Amazon Pinpoint SMS crea un argomento nell'account. L'argomento viene creato automaticamente con tutte le autorizzazioni richieste. Per ulteriori informazioni sugli argomenti Amazon SNS, consulta [Configurazione Amazon SNS](#) nella Guida per gli sviluppatori di Amazon Simple Notification Service.

- Per Destinazione del messaggio in entrata, inserisci il nome di un argomento, ad esempio, **SMSRegistrationFormTopic**.
7. Per Ruolo del canale bidirezionale, scegli Usa le policy sugli argomenti SNS.
  8. Seleziona Salvataggio delle modifiche.

Usa la console Amazon Pinpoint SMS per aggiungere al numero di telefono le parole chiave che i clienti ti inviano per confermare i loro abbonamenti (ad esempio, **Yes** o **Confirm**).

#### Aggiunta di una parola chiave

1. Apri la console Amazon Pinpoint SMS all'indirizzo <https://console.aws.amazon.com/sms-voice/>.
2. Nel pannello di navigazione, in Configurazioni, scegli Numero di telefono.
3. Nella pagina Numero di telefono scegli un numero di telefono a cui aggiungere una parola chiave.
4. Nella scheda Parole chiave, scegli il pulsante Aggiungi parola chiave.
5. Nel riquadro Parole chiave personalizzate aggiungi quanto segue:
  - Parola chiave: la nuova parola chiave da aggiungere (ad esempio **Yes** o **Confirm**).
  - Messaggio di risposta: il messaggio da inviare al destinatario.
  - Azione con parole chiave: l'operazione da eseguire quando si riceve la parola chiave. Scegli Risposta automatica.
6. Scegli Aggiungi parola chiave.

Successivo: [Creazione di policy e ruoli IAM](#)

## Passaggio 2: creare policy e ruoli IAM

Il passaggio successivo nell'implementazione della soluzione di registrazione SMS consiste nella configurazione di una policy e di un ruolo in AWS Identity and Access Management (IAM). Per questa soluzione, è necessario creare una policy che fornisca l'accesso a determinate risorse correlate ad Amazon Pinpoint. Creerai quindi un ruolo a cui collegare la policy. Più avanti in questo tutorial, creerai una AWS Lambda funzione che utilizza questo ruolo per richiamare determinate operazioni nell'API Amazon Pinpoint.

## Creazione di una policy IAM

Questa sezione illustra come creare una policy IAM. Gli utenti e i ruoli che usano questa policy possono eseguire le seguenti operazioni:

- Utilizzare la funzione di convalida del numero di telefono
- È possibile visualizzare, creare e aggiornare gli endpoint Amazon Pinpoint
- Invio di messaggi agli endpoint Amazon Pinpoint

In questo tutorial, vuoi consentire a Lambda di eseguire queste attività. Tuttavia, per una maggiore sicurezza, questa policy utilizza il principale dei privilegi minimi di concessione. In altre parole, concede solo le autorizzazioni necessarie per completare questa soluzione, e nient'altro. Questa policy è limitata nei seguenti modi:

- È possibile utilizzarla solo per chiamare l'API di convalida del numero di telefono in una regione specifica..
- È possibile utilizzarla solo per visualizzare, creare o aggiornare gli endpoint associati a un progetto Amazon Pinpoint specifico.
- È possibile utilizzarla solo per inviare messaggi agli endpoint associati a un progetto Amazon Pinpoint specifico.

### Come creare la policy

1. Accedi AWS Management Console e apri la console IAM all'[indirizzo https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
3. Nella scheda JSON, incollare il codice seguente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:CreateLogGroup"
      ]
    }
  ],
}
```

```

    "Resource": "arn:aws:logs:*:*:*"
  },
  {
    "Effect": "Allow",
    "Action": "mobiletargeting:SendMessage",
    "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mobiletargeting:GetEndpoint",
      "mobiletargeting:UpdateEndpoint",
      "mobiletargeting:PutEvents"
    ],
    "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/
endpoints/*"
  },
  {
    "Effect": "Allow",
    "Action": "mobiletargeting:PhoneNumberValidate",
    "Resource": "arn:aws:mobiletargeting:region:accountId:phone/number/
validate"
  }
]
}

```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci la **AWS regione** con la regione in cui utilizzi Amazon Pinpoint, ad esempio us-east-1 o eu-central-1

#### Tip

Per un elenco completo delle AWS regioni in cui è disponibile Amazon Pinpoint, consulta [AWS le regioni e gli endpoint](#) nel. Riferimenti generali di AWS

- Sostituisci **AccountID** con l'ID univoco del tuo AWS account.
- Sostituisci **ProjectID** con l'ID univoco del progetto che hai creato in [Crea un progetto Amazon Pinpoint](#) di questo tutorial.

**Note**

Le logs azioni consentono a Lambda di registrare l'output in CloudWatch Logs.

4. Seleziona Successivo.
5. Per Nome della politica, inserisci un nome per la politica, ad esempio.  
**RegistrationFormPolicy** Scegli Crea policy.

## Creazione di un ruolo IAM

### Come creare il ruolo

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione della console IAM, scegli Ruoli, quindi Crea ruolo.
3. In Tipo di entità affidabile, scegli AWS servizio, quindi per Servizio o caso utente scegli Lambda dall'elenco a discesa.
4. Seleziona Successivo.
5. In Politiche di autorizzazione, scegli o cerca la politica che hai creato nella sezione precedente, quindi scegli Avanti.
6. In Dettagli del ruolo, in Nome del ruolo, inserisci un nome per il ruolo, ad esempio **SMSRegistrationForm**. Scegli Crea ruolo.

Successivo: [Creazione della funzione Lambda](#)

## Passaggio 1: creare funzioni Lambda

Questa soluzione utilizza due funzioni Lambda. In questa sezione viene descritto come creare e configurare queste funzioni. Più avanti, Gateway API e Amazon Pinpoint verranno configurati per eseguire queste funzioni quando si verificano determinati eventi. Entrambe le funzioni creano e aggiornano gli endpoint nel progetto Amazon Pinpoint specificato. La prima funzione utilizza anche la funzione di convalida dei numeri di telefono.

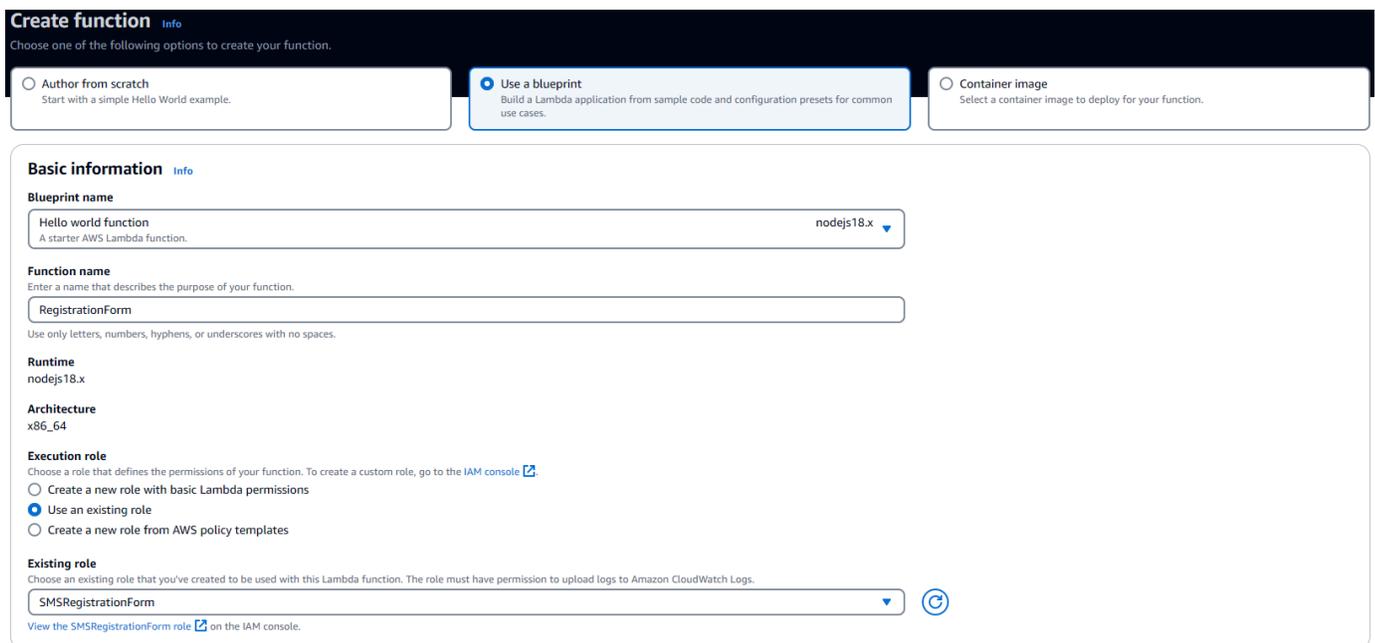
Creazione della funzione che convalida le informazioni sul cliente e crea gli endpoint.

La prima funzione viene inserita dal modulo di registrazione, che riceve da Amazon API Gateway. Utilizza queste informazioni per ottenere informazioni sul numero di telefono del cliente utilizzando

la funzione di [convalida del numero di telefono](#) di Amazon Pinpoint. La funzione utilizza quindi i dati convalidati per creare un nuovo endpoint nel progetto Amazon Pinpoint specificato. Per impostazione predefinita, l'endpoint creato dalla funzione è escluso dalle future comunicazioni dell'utente, ma questa condizione può essere modificata dalla seconda funzione. Infine, questa funzione invia al cliente un messaggio per verificare la volontà di ricevere SMS dall'utente.

## Creazione della funzione Lambda

1. [Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. Scegli Crea funzione.
3. In Crea una funzione, scegli Usa un blueprint.
4. Nel campo di ricerca, digitare **hello** e premere Invio. Nell'elenco dei risultati, scegliere la funzione Node.js hello-world, come illustrato nella seguente immagine.



**Create function** Info

Choose one of the following options to create your function.

- Author from scratch  
Start with a simple Hello World example.
- Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image  
Select a container image to deploy for your function.

**Basic information** Info

**Blueprint name**  
Hello world function A starter AWS Lambda function. nodejs18.x ▾

**Function name**  
Enter a name that describes the purpose of your function.  
RegistrationForm  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime**  
nodejs18.x

**Architecture**  
x86\_64

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
 Create a new role with basic Lambda permissions  
 Use an existing role  
 Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
SMSRegistrationForm ▾ 🔍  
[View the SMSRegistrationForm role](#) on the IAM console.

5. In Basic information (Informazioni di base) eseguire queste operazioni:
  - In Name (Nome), inserire un nome per la funzione, ad esempio **RegistrationForm**.
  - Per Role (Ruolo) selezionare Choose an existing role (Scegli un ruolo esistente).
  - Per Ruolo esistente, scegli il RegistrationForm ruolo SMS che hai creato in [Crea un ruolo IAM](#).

Al termine, selezionare Create function (Crea funzione).

6. Per Code source, elimina la funzione di esempio nell'editor di codice, quindi incolla il codice seguente:

```
import { PinpointClient, PhoneNumberValidateCommand, UpdateEndpointCommand,
  SendMessagesCommand } from "@aws-sdk/client-pinpoint"; // ES Modules import
const pinClient = new PinpointClient({region: process.env.region});

// Make sure the SMS channel is enabled for the projectId that you specify.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-setup.html
var projectId = process.env.projectId;

// You need a dedicated long code in order to use two-way SMS.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-voice-manage.html#channels-voice-manage-request-phone-numbers
var originationNumber = process.env.originationNumber;

// This message is spread across multiple lines for improved readability.
var message = "ExampleCorp: Reply YES to confirm your subscription. 2 msgs per "
  + "month. No purchase req'd. Msg&data rates may apply. Terms: "
  + "example.com/terms-sms";

var messageType = "TRANSACTIONAL";

export const handler = async (event, context) => {
  console.log('Received event:', event);
  await validateNumber(event);
};

async function validateNumber (event) {
  var destinationNumber = event.destinationNumber;
  if (destinationNumber.length == 10) {
    destinationNumber = "+1" + destinationNumber;
  }
  var params = {
    NumberValidateRequest: {
      IsoCountryCode: 'US',
      PhoneNumber: destinationNumber
    }
  };
  try{
    const PhoneNumberValidatorresponse = await pinClient.send( new
    PhoneNumberValidateCommand(params));
    console.log(PhoneNumberValidatorresponse);
    if (PhoneNumberValidatorresponse['NumberValidateResponse']['PhoneTypeCode'] ==
    0) {
```

```
    await createEndpoint(PhoneNumberValidatorResponse, event.firstName,
event.lastName, event.source);

    } else {
        console.log("Received a phone number that isn't capable of receiving "
            +"SMS messages. No endpoint created.");
    }
} catch(err){
    console.log(err);
}
}

async function createEndpoint(data, firstName, lastName, source) {
    var destinationNumber = data['NumberValidateResponse']
['CleansedPhoneNumberE164'];
    var endpointId = data['NumberValidateResponse']
['CleansedPhoneNumberE164'].substring(1);

    var params = {
        ApplicationId: projectId,
        // The Endpoint ID is equal to the cleansed phone number minus the leading
        // plus sign. This makes it easier to easily update the endpoint later.
        EndpointId: endpointId,
        EndpointRequest: {
            ChannelType: 'SMS',
            Address: destinationNumber,
            // OptOut is set to ALL (that is, endpoint is opted out of all messages)
            // because the recipient hasn't confirmed their subscription at this
            // point. When they confirm, a different Lambda function changes this
            // value to NONE (not opted out).
            OptOut: 'ALL',
            Location: {
                PostalCode: data['NumberValidateResponse']['ZipCode'],
                City: data['NumberValidateResponse']['City'],
                Country: data['NumberValidateResponse']['CountryCodeIso2'],
            },
            Demographic: {
                Timezone: data['NumberValidateResponse']['Timezone']
            },
            Attributes: {
                Source: [
                    source
                ]
            },
        },
    },
```

```
    User: {
      UserAttributes: {
        FirstName: [
          firstName
        ],
        LastName: [
          lastName
        ]
      }
    }
  };
  try{
    const UpdateEndpointresponse = await pinClient.send(new
UpdateEndpointCommand(params));
    console.log(UpdateEndpointresponse);
    await sendConfirmation(destinationNumber);
  }catch(err){
    console.log(err);
  }
}

async function sendConfirmation(destinationNumber) {
  var params = {
    ApplicationId: projectId,
    MessageRequest: {
      Addresses: {
        [destinationNumber]: {
          ChannelType: 'SMS'
        }
      },
      MessageConfiguration: {
        SMSMessage: {
          Body: message,
          MessageType: messageType,
          OriginationNumber: originationNumber
        }
      }
    }
  };
  try{
    const SendMessagesCommandresponse = await pinClient.send(new
SendMessagesCommand(params));
    console.log("Message sent! ")
```

```

    + SendMessagesCommandresponse['MessageResponse']['Result']
[destinationNumber]['StatusMessage']);
  }catch(err){
    console.log(err);
  }
}

```

7. Nella scheda Configurazione per le variabili di ambiente, scegliete Modifica, quindi Aggiungi variabile di ambiente, effettuate le seguenti operazioni:

- Nella prima riga, creare una variabile con una chiave di **originationNumber**. Quindi, impostare il valore sul numero di telefono del codice lungo dedicato ricevuto nella [Fase 1.2](#).

#### Note

Ricordare di includere il segno (+) e il codice del paese nel numero di telefono. Non includere altri caratteri speciali, ad esempio trattini (-), punti (.) o parentesi.

- Nella seconda riga, creare una variabile con una chiave di **projectId**. Quindi, impostare il valore sull'ID univoco del progetto creato nella [Fase 1.1](#).
- Nella terza riga, creare una variabile con una chiave di **region**. Quindi, imposta il valore sulla regione in cui viene utilizzato Amazon Pinpoint, ad esempio **us-east-1** o **us-west-2**.

Al termine, la sezione Environment Variables (Variabili di ambiente) dovrebbe essere simile all'esempio illustrato nella seguente immagine.

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more.](#)

originationNumber	+12065550199	Remove
projectId	33d643d9bexample9a5e726f6c4	Remove
region	us-east-1	Remove
Key	Value	Remove

► Encryption configuration

8. Nella parte superiore della pagina, scegli Save (Salva).

## Test della funzione

Una volta creata la funzione, è opportuno testarla per verificarne la corretta configurazione. Verifica inoltre che il ruolo IAM creato disponga delle autorizzazioni appropriate.

### Per testare la funzione

1. Seleziona la scheda Test.
2. Scegliete Crea nuovo evento, effettuate le seguenti operazioni:
  - Nel campo Event name (Nome evento), inserire un nome per l'evento di test, ad esempio **MyPhoneNumber**.
  - Cancellare il codice di esempio nell'editor di codici. Incolla il codice seguente:

```
{
  "destinationNumber": "+12065550142",
  "firstName": "Carlos",
  "lastName": "Salazar",
  "source": "Registration form test"
}
```

- Nel precedente codice di esempio, sostituire i valori degli attributi `destinationNumber`, `lastName` e `firstName`, con i valori da utilizzare per il test, ad esempio le informazioni di contatto personali. Quando si testa la funzione, questa invia un SMS al numero di telefono specificato nell'attributo `destinationNumber`. Verificare che il numero di telefono specificato possa ricevere gli SMS.
  - Scegliere Create (Crea).
3. Scegliere Test (Esegui test).
  4. In Execution result: succeeded (Risultato esecuzione: riuscita), selezionare Details (Dettagli). Nella sezione Log output (Output di log), riesaminare l'output della funzione. Verificare che la funzione sia stata eseguita senza errori.

Controllare il dispositivo associato al `destinationNumber` specificato per verificare che abbia ricevuto il messaggio di test.

5. Apri la console Amazon Pinpoint all'indirizzo <https://console.aws.amazon.com/pinpoint/>.
6. Nella pagina Tutti i progetti, scegli il progetto che hai creato in [Crea un progetto Amazon Pinpoint](#).

7. Nel riquadro di navigazione, selezionare Segments (Segmenti). Nella Segments page (Pagina segmenti), selezionare Create a segment (Crea un segmento).
8. In Segment group 1 (Gruppo segmenti 1), nel campo Add filters to refine your segment (Aggiungi filtri per rifinire i segmenti), selezionare Filter by user (Filtra per utente).
9. Per Scegli un attributo utente, scegli FirstName. Quindi, in Choose values (Scegli valori), selezionare il nome specificato nell'evento di prova.

La sezione Segment estimate (Stima segmenti) dovrebbe indicare che non ci sono endpoint idonei e un endpoint totale, come illustrato nella seguente immagine. Questo risultato è previsto. Quando la funzione crea un nuovo endpoint, l'endpoint viene disattivato. I segmenti in Amazon Pinpoint escludono automaticamente gli endpoint interessati da un'azione di opt-out.

The screenshot displays the Amazon Pinpoint console interface for creating and estimating a segment. On the left, the 'Segment group 1' configuration is shown, including a filter for 'User' with the attribute 'FirstName' set to 'Carlos'. On the right, the 'Segment estimate' section shows '0 endpoints' and a message indicating that no matches were found, suggesting the segment filters need to be modified.

## Creazione della funzione che abilita i clienti alle comunicazioni

La seconda funzione viene eseguita solo quando un cliente risponde al messaggio inviato dalla prima funzione. Se la risposta del cliente include la parola chiave specificata in [Abilita SMS bidirezionali](#), la funzione aggiorna il record dell'endpoint per consentirgli di ricevere comunicazioni future. Amazon Pinpoint risponde inoltre automaticamente con il messaggio specificato in [Abilita SMS bidirezionali](#).

Se il cliente non risponde o risponde con una parola chiave diversa da quella designata, non succede nulla. L'endpoint del cliente resta in Amazon Pinpoint, ma non può essere scelto come destinazione dei segmenti.

## Creazione della funzione Lambda

1. [Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. Scegli Crea funzione.
3. In Create a function (Crea una funzione), selezionare Blueprints (Piani).
4. Nel campo di ricerca, digitare **hello** e premere Invio. Nell'elenco dei risultati, scegliere la funzione Node.js hello-world, come illustrato nella seguente immagine. Scegliere Configure (Configura).
5. In Basic information (Informazioni di base) eseguire queste operazioni:
  - In Name (Nome), inserire un nome per la funzione, ad esempio **RegistrationForm\_OptIn**.
  - Per Role (Ruolo) selezionare Choose an existing role (Scegli un ruolo esistente).
  - Per Ruolo esistente, scegli il RegistrationForm ruolo SMS che hai creato in [Crea un ruolo IAM](#).

Al termine, selezionare Create function (Crea funzione).

6. Eliminare la funzione di esempio nell'editor di codici, quindi incollare il codice seguente:

```
import { PinpointClient, UpdateEndpointCommand } from "@aws-sdk/client-pinpoint"; // ES Modules import

// Create a new Pinpoint client instance with the region specified in the
// environment variables
const pinClient = new PinpointClient({ region: process.env.region });

// Get the Pinpoint project ID and the confirm keyword from environment variables
const projectId = process.env.projectId;
const confirmKeyword = process.env.confirmKeyword.toLowerCase();

// This is the main handler function that is invoked when the Lambda function is
// triggered
export const handler = async (event, context) => {
  console.log('Received event:', event);

  try {
    // Extract the timestamp, message, and origination number from the SNS
    event
      const timestamp = event.Records[0].Sns.Timestamp;
      const message = JSON.parse(event.Records[0].Sns.Message);
      const originationNumber = message.originationNumber;
```

```
const response = message.messageBody.toLowerCase();

// Check if the response message contains the confirm keyword
if (response.includes(confirmKeyword)) {
    // If the confirm keyword is found, update the endpoint's opt-in status
    await updateEndpointOptIn(originationNumber, timestamp);
}
} catch (error) {
    console.error('An error occurred:', error);
    throw error; // Rethrow the error to handle it upstream
}
};

// This function updates the opt-in status of a Pinpoint endpoint
async function updateEndpointOptIn(originationNumber, timestamp) {
    // Extract the endpoint ID from the origination number
    const endpointId = originationNumber.substring(1);

    // Prepare the parameters for the UpdateEndpointCommand
    const params = {
        ApplicationId: projectId,
        EndpointId: endpointId,
        EndpointRequest: {
            Address: originationNumber,
            ChannelType: 'SMS',
            OptOut: 'NONE',
            Attributes: {
                OptInTimestamp: [timestamp]
            },
        },
    };

    try {
        // Send the UpdateEndpointCommand to update the endpoint's opt-in status
        const updateEndpointResponse = await pinClient.send(new
UpdateEndpointCommand(params));
        console.log(updateEndpointResponse);
        console.log(`Successfully changed the opt status of endpoint ID
${endpointId}`);
    } catch (error) {
        console.error('An error occurred while updating endpoint:', error);
        throw error; // Rethrow the error to handle it upstream
    }
}
```

```
}
```

- In Environment variables (Variabili di ambiente), effettuare le seguenti operazioni:
  - Nella prima riga, creare una variabile con una chiave di **projectId**. Successivamente, imposta il valore sull'ID univoco del progetto che hai creato in [Crea un progetto Amazon Pinpoint](#).
  - Nella seconda riga, creare una variabile con una chiave di **region**. Quindi, imposta il valore sulla regione in cui viene utilizzato Amazon Pinpoint, ad esempio **us-east-1** o **us-west-2**.
  - Nella terza riga, creare una variabile con una chiave di **confirmKeyword**. Successivamente, imposta il valore sulla parola chiave di conferma che hai creato in [Abilita SMS bidirezionali](#).

#### Note

La parola chiave non fa distinzione tra le lettere maiuscole e minuscole. Questa funzione converte il messaggio in entrata in lettere minuscole.

Al termine, la sezione Environment Variables (Variabili di ambiente) dovrebbe essere simile all'esempio illustrato nella seguente immagine.

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more.](#)

projectId	33d643d9example9a5e726f6c4a	Remove
confirmKeyword	Yes	Remove
region	us-east-1	Remove
Key	Value	Remove

► Encryption configuration

- Nella parte superiore della pagina, scegli Save (Salva).

## Test della funzione

Una volta creata la funzione, è opportuno testarla per verificarne la corretta configurazione. Verifica inoltre che il ruolo IAM creato disponga delle autorizzazioni appropriate.

### Per testare la funzione

1. Scegli Test (Esegui test).
2. Nella finestra Configure test event (Configura evento di test), seguire la procedura riportata di seguito:
  - a. Scegliere Create new test event (Creare nuovo evento di test).
  - b. Nel campo Event name (Nome evento), inserire un nome per l'evento di test, ad esempio **MyResponse**.
  - c. Cancellare il codice di esempio nell'editor di codici. Incolla il codice seguente:

```
{
  "Records": [
    {
      "Sns": {
        "Message": "{\"originationNumber\": \"+12065550142\", \"messageBody\": \"Yes\"}",
        "Timestamp": "2019-02-20T17:47:44.147Z"
      }
    }
  ]
}
```

Nel codice di esempio precedente, sostituisci i valori dell'attributo `originationNumber` con il numero di telefono utilizzato durante il test della funzione Lambda precedente. Sostituisci il valore di `messageBody` con la parola chiave SMS bidirezionale specificata in [Abilità SMS bidirezionali](#). Opzionalmente è possibile sostituire il valore di `Timestamp` con la data e l'ora attuale.

- d. Scegli Crea.
3. Scegliere Test di nuovo.
  4. In Execution result: succeeded (Risultato esecuzione: riuscita), selezionare Details (Dettagli). Nella sezione Log output (Output di log), riesaminare l'output della funzione. Verificare che la funzione sia stata eseguita senza errori.

5. Apri la console Amazon Pinpoint all'indirizzo <https://console.aws.amazon.com/pinpoint/>.
6. Nella pagina Tutti i progetti, scegli il progetto che hai creato in [Crea un progetto Amazon Pinpoint](#).
7. Nel riquadro di navigazione, selezionare Segments (Segmenti). Nella Segments page (Pagina segmenti), selezionare Create a segment (Crea un segmento).
8. In Segment group 1 (Gruppo segmenti 1), nel campo Add filters to refine your segment (Aggiungi filtri per rifinire i segmenti), selezionare Filter by user (Filtra per utente).
9. Per Scegli un attributo utente, scegli FirstName. Quindi, in Choose values (Scegli valori), selezionare il nome specificato nell'evento di prova.

La sezione Segment estimate (Stima segmenti) dovrebbe indicare che c'è un endpoint idoneo e un endpoint totale.

Successivo: [Configurazione di Gateway Amazon API](#)

## Passaggio 4: configurare Gateway Amazon API

Questa sezione illustra la creazione di una nuova API tramite Gateway Amazon API. Il modulo di registrazione implementato in questa soluzione chiama questa API. API Gateway passa quindi le informazioni acquisite nel modulo di registrazione alla funzione Lambda creata in [Create Lambda functions](#).

### Creazione dell'API

In primo luogo, è necessario creare una nuova API in Gateway API. Le procedure seguenti mostrano come creare una nuova API REST.

Per creare una nuova API

1. Aprire la console Gateway API all'indirizzo <https://console.aws.amazon.com/apigateway/>.
2. Seleziona Create API (Crea API). Operare le seguenti selezioni:
  - In Choose the protocol (Scegliere il protocollo) scegliere REST.
  - In Create new API (Crea nuova API) scegliere New API (Nuova API).
  - In Settings (Impostazioni), per Name (Nome), immettere un nome, come **RegistrationForm**. Per Description (Descrizione), è possibile immettere un testo che descriva lo scopo dell'API. In Endpoint Type (Tipo di endpoint), scegliere Regional (Regionale). Quindi, scegliere Create API (Crea API).

Un esempio di queste impostazioni è illustrato nella seguente immagine.

### Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

REST  WebSocket

### Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

New API  Clone from existing API  Import from Swagger or Open API 3  Example API

### Settings

Choose a friendly name and description for your API.

API name*	<input type="text" value="RegistrationForm"/>
Description	<input type="text" value="Collects input from a registration form. which is passed on to a"/>
Endpoint Type	<input type="text" value="Regional"/> ⓘ

\* Required

Create API

## Creazione di una risorsa

Dopo aver creato un'API, puoi iniziare ad aggiungere risorse al suo interno. In seguito, puoi aggiungere un metodo POST alla risorsa e indicare a Gateway API di trasferire i dati ricevuti da questo metodo alla funzione Lambda.

1. Nel menu Actions (Operazioni) scegliere Create Resource (Crea risorsa). Nel riquadro New Child Resource (Nuova risorsa figlio), per Resource Name (Nome risorsa), immettere **register**, come illustrato nella seguente immagine. Scegliere Create Resource (Crea risorsa).

## New Child Resource

Use this page to create a new child resource for your resource. 

Configure as [proxy resource](#)

Resource Name\*

Resource Path\*

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring **/{proxy+}** as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

Enable API Gateway CORS

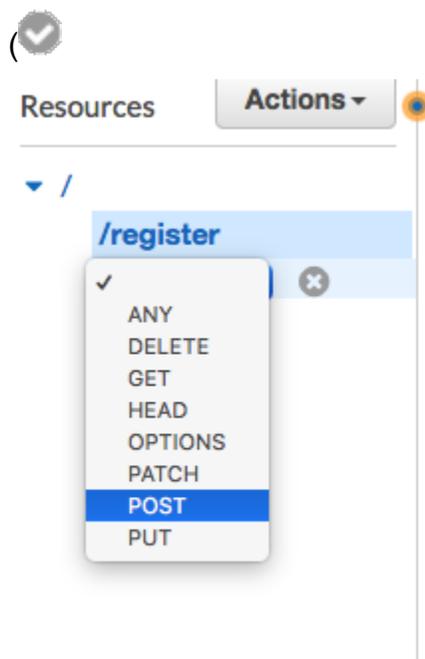
 

\* Required

Cancel

Create Resource

- Nel menu Actions (Operazioni) scegliere Create method (Crea metodo). Dal menu visualizzato, scegliere POST, come illustrato nella seguente immagine. Quindi scegliere il pulsante del segno di spunta



- Nel riquadro /register - POST - Setup (registra - POST - Configura), effettua le seguenti selezioni:

- Per Integration type (Tipo di integrazione), scegliere Lambda Function (Funzione Lambda).
- Scegli Usa integrazione proxy Lambda.
- In Regione Lambda, scegli la regione in cui è stata creata la funzione Lambda.
- Per Funzione Lambda, scegli la RegisterEndpoint funzione che hai creato in [Crea funzioni Lambda](#).

Un esempio di queste impostazioni è illustrato nella seguente immagine.

## /register - POST - Setup



Choose the integration point for your new method.

**Integration type**  Lambda Function ⓘ

HTTP ⓘ

Mock ⓘ

AWS Service ⓘ

VPC Link ⓘ

**Use Lambda Proxy integration**  ⓘ

**Lambda Region**

**Lambda Function**

ⓘ

**Use Default Timeout**  ⓘ

**Save**

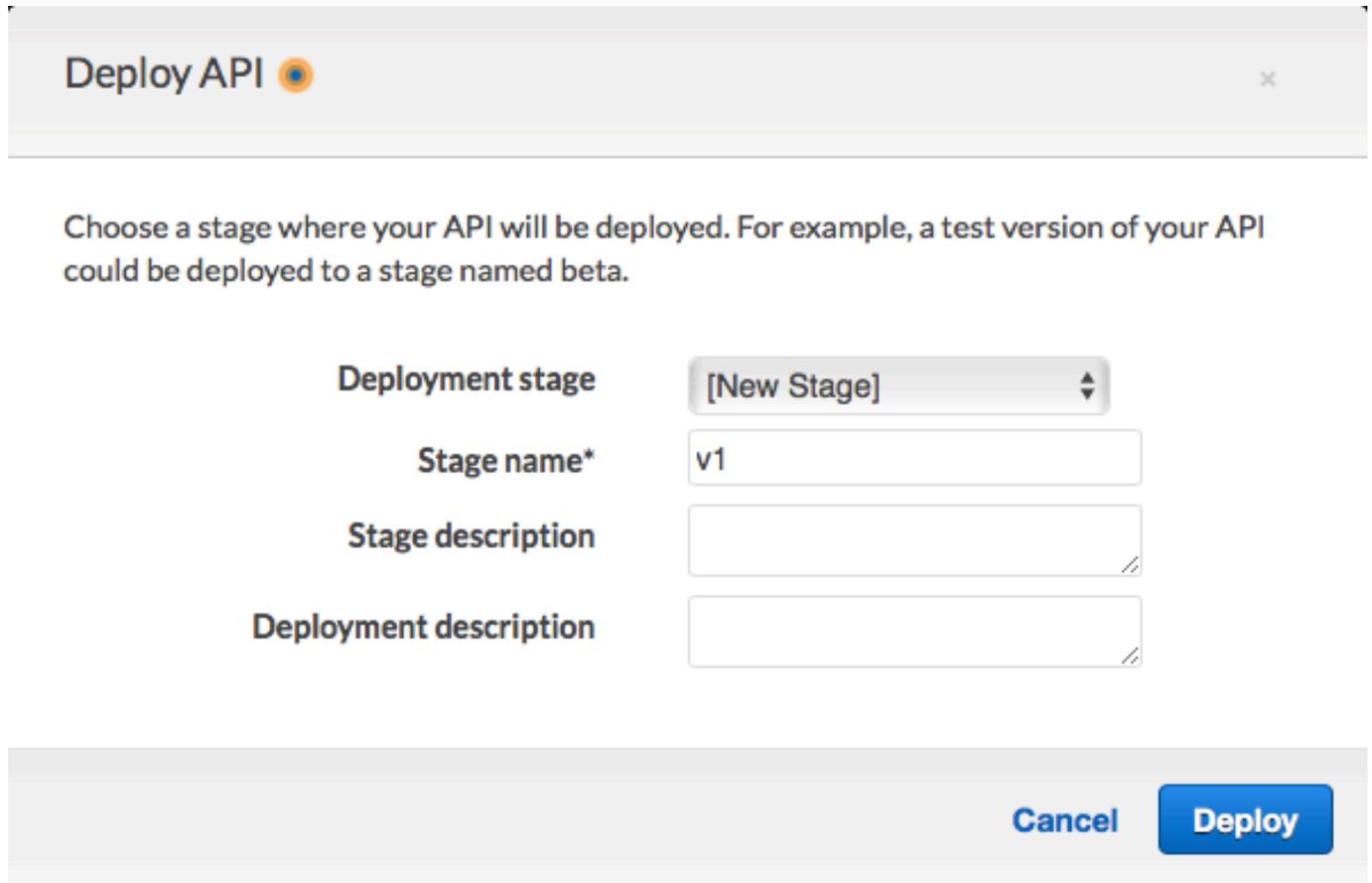
Selezionare Salva. Nella finestra visualizzata, scegli OK per autorizzare Gateway API a eseguire la funzione Lambda.

## Distribuzione dell'API

L'API è ora pronta per l'uso. A questo punto, è necessario distribuirla per creare un endpoint accessibile pubblicamente.

1. Nel menu Actions (Operazioni), scegliere Deploy API (Distribuisci API). Nella finestra Deploy API (Distribuisci API), effettuare le seguenti selezioni:
  - In Deployment stage (Fase di distribuzione), scegliere [New Stage] ([Nuova fase]).
  - In Stage name (Nome fase) immettere **v1**.
  - Selezionare Deploy (Distribuisci).

Un esempio di queste selezioni è illustrato nella seguente immagine.



Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage	[New Stage]
Stage name*	v1
Stage description	
Deployment description	

Cancel Deploy

2. Nel riquadro v1 Stage Editor (Editor fasi v1), scegliere la risorsa /register (/registra) e poi scegliere il metodo POST. Copiare l'indirizzo che viene visualizzato accanto a Invoke URL (URL chiamata), come illustrato nella seguente immagine.

## v1 - POST - /register

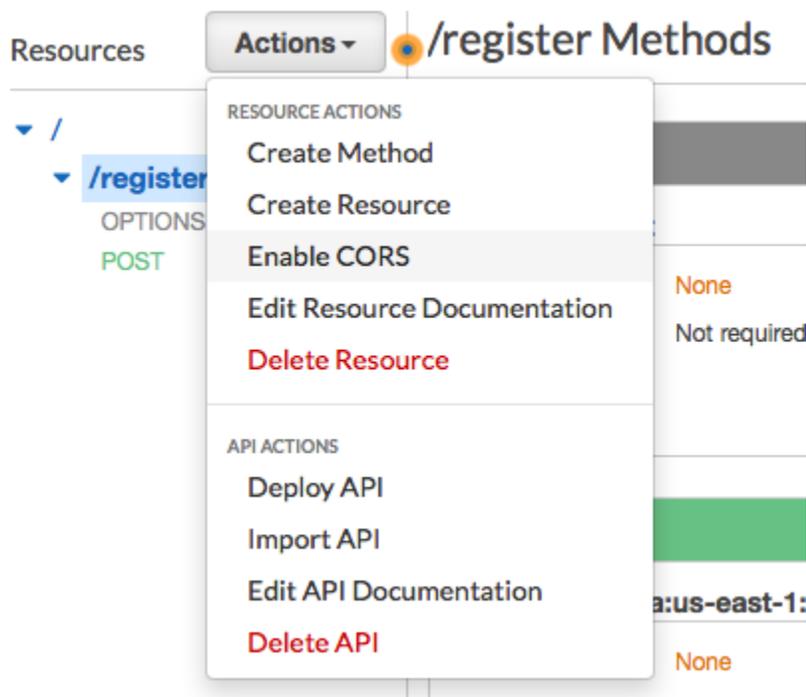
Invoke URL: <https://example.execute-api.us-east-1.amazonaws.com/v1/register>

Use this page to override the v1 stage settings for the POST to /register method.

Settings  Inherit from stage  
 Override for this method

[Save Changes](#)

3. Nel riquadro di spostamento seleziona Resources (Risorse). Nell'elenco delle risorse, scegliere la risorsa /register (/registra). Infine, nel menu Actions (Operazioni), scegliere Enable CORS (Abilita CORS), come illustrato nella seguente immagine.



4. Nel riquadro Enable CORS (Abilita CORS), scegliere Enable CORS and replace existing CORS headers (Abilita CORS e sostituisci intestazioni CORS esistenti).

Successivo: [Creazione e distribuzione del modulo Web](#)

## Fase 5: creazione e distribuzione del modulo Web

Tutti i componenti di questa soluzione che utilizzano AWS i servizi sono ora disponibili. L'ultima fase prevede la creazione e la distribuzione del modulo Web che acquisisce i dati del cliente.

## Crea il gestore del JavaScript modulo

In questa sezione, crei una JavaScript funzione che analizza il contenuto del modulo Web creato nella sezione successiva. Dopo aver analizzato il contenuto, questa funzione invia i dati all'API che hai creato in [Configurare Amazon API Gateway](#).

Per creare il gestore del modulo

1. In un editor di testo, crea un nuovo file.
2. Nell'editor, incollare il seguente codice.

```
$(document).ready(function() {  
  
    // Handle form submission.  
    $("#submit").click(function(e) {  
  
        var firstName = $("#firstName").val(),  
            lastName = $("#lastName").val(),  
            source = window.location.pathname,  
            optTimestamp = undefined,  
            utcSeconds = Date.now() / 1000,  
            timestamp = new Date(0),  
            phone = $("#areaCode").val()  
                + $("#phone1").val()  
                + $("#phone2").val();  
  
        e.preventDefault();  
  
        if (firstName == "") {  
            $('#form-response').html('<div class="mt-3 alert alert-info"  
role="alert">Please enter your first name.</div>');  
        } else if (lastName == "") {  
            $('#form-response').html('<div class="mt-3 alert alert-info"  
role="alert">Please enter your last name.</div>');  
        } else if (phone.match(/[^0-9]/gi)) {  
            $('#form-response').html('<div class="mt-3 alert alert-info"  
role="alert">Your phone number contains invalid characters. Please check the phone  
number that you supplied.</div>');  
        } else if (phone.length < 10) {  
            $('#form-response').html('<div class="mt-3 alert alert-info"  
role="alert">Please enter your phone number.</div>');  
        } else if (phone.length > 10) {
```

```
$('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Your phone number contains too many digits. Please check the phone
number that you supplied.</div>');
} else {
    $('#submit').prop('disabled', true);
    $('#submit').html('<span class="spinner-border spinner-border-sm"
role="status" aria-hidden="true"></span> Saving your preferences</button>');

    timestamp.setUTCSeconds(utcSeconds);

    var data = JSON.stringify({
        'destinationNumber': phone,
        'firstName': firstName,
        'lastName': lastName,
        'source': source,
        'optTimestamp': timestamp.toString()
    });

    $.ajax({
        type: 'POST',
        url: 'https://example.execute-api.us-east-1.amazonaws.com/v1/register',
        contentType: 'application/json',
        data: data,
        success: function(res) {
            $('#form-response').html('<div class="mt-3 alert alert-success"
role="alert"><p>Congratulations! You've successfully registered for SMS
Alerts from ExampleCorp.</p><p>We just sent you a message. Follow the instructions
in the message to confirm your subscription. We won't send any additional
messages until we receive your confirmation.</p><p>If you decide you don't
want to receive any additional messages from us, just reply to one of our messages
with the keyword STOP.</p></div>');
            $('#submit').prop('hidden', true);
            $('#unsubAll').prop('hidden', true);
            $('#submit').text('Preferences saved!');
        },
        error: function(jqxhr, status, exception) {
            $('#form-response').html('<div class="mt-3 alert alert-danger"
role="alert">An error occurred. Please try again later.</div>');
            $('#submit').text('Save preferences');
            $('#submit').prop('disabled', false);
        }
    });
}
});
```

```
});
```

3. Nell'esempio precedente, sostituisci `https://example.execute-api.us-east-1.amazonaws.com/v1/register` con l'URL Invoke ottenuto in [Deploy the API](#).
4. Salvare il file.

## Crea il file del modulo

In questa sezione, viene creato un file HTML contenente il modulo utilizzato dai clienti per effettuare la registrazione al tuo programma SMS. Questo file utilizza il gestore di JavaScript moduli creato nella sezione precedente per trasmettere i dati del modulo alla funzione Lambda.

### Important

L'invio del modulo da parte di un'attiva una funzione Lambda che chiama varie operazioni dell'API Amazon Pinpoint. Utenti malintenzionati potrebbero lanciare un attacco al modulo, che potrebbe provocare un elevato numero di richieste. Se prevedi di utilizzare questa soluzione per un caso d'uso di produzione, è consigliabile proteggerla tramite un sistema come [Google reCAPTCHA](#).

## Per creare il modulo

1. In un editor di testo, crea un nuovo file.
2. Nell'editor, incollare il seguente codice.

```
<!doctype html>
<html lang="en">

<head>
  <!-- Meta tags required by Bootstrap -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gg0yR0iXcBMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

```

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/
popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
U02eT0CpHqdSJK6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

<script type="text/javascript" src="SMSFormHandler.js"></script>
<title>SMS Registration Form</title>
</head>

<body>
<div class="container">
<div class="row justify-content-center mt-3">
<div class="col-md-6">
<h1>Register for SMS Alerts</h1>
<p>Enter your phone number below to sign up for PromotionName messages from
ExampleCorp.</p>
<p>We don't share your contact information with anyone else. For more
information, see our <a href="http://example.com/privacy">Privacy Policy</a>.</p>
<p>ExampleCorp alerts are only available to recipients in the United
States.</p>
</div>
</div>
<div class="row justify-content-center">
<div class="col-md-6">
<form>
<div class="form-group">
<label for="firstName" class="font-weight-bold">First name</label>
<input type="text" class="form-control" id="firstName"
placeholder="Your first name" required>
</div>
<div class="form-group">
<label for="lastName" class="font-weight-bold">Last name</label>
<input type="text" class="form-control" id="lastName" placeholder="Your
last name" required>
</div>
<label for="areaCode" class="font-weight-bold">Phone number</label>

```

```

        <div class="input-group">
            <span class="h3">(&nbsp;</span>
                <input type="tel" class="form-control" id="areaCode" placeholder="Area
code" required>
                <span class="h3">&nbsp;</span>)&nbsp;</span>
                <input type="tel" class="form-control" id="phone1" placeholder="555"
required>
                <span class="h3">&nbsp;</span>)&nbsp;</span>
                <input type="tel" class="form-control" id="phone2" placeholder="0199"
required>
            </div>
            <div id="form-response"></div>
            <button id="submit" type="submit" class="btn btn-primary btn-block
mt-3">Submit</button>
        </form>
    </div>
</div>
<div class="row mt-3">
    <div class="col-md-12 text-center">
        <small class="text-muted">Copyright © 2019, ExampleCorp or its
affiliates.</small>
    </div>
</div>
</div>
</body>
</html>

```

3. Nell'esempio precedente, sostituisci *SMS FormHandler .js* con il percorso completo del JavaScript file del gestore del modulo creato nella sezione precedente.
4. Salvare il file.

### Carica i file del modulo

Ora che hai creato il modulo HTML e il gestore del JavaScript modulo, l'ultimo passaggio consiste nel pubblicare questi file su Internet. In questa sezione si presuppone la disponibilità di un provider di hosting Web. Se non disponi di un provider di hosting esistente, puoi avviare un sito Web utilizzando Amazon Route 53, Amazon Simple Storage Service (Amazon S3) e Amazon CloudFront. Per ulteriori informazioni, consulta [Hosting di un sito Web statico](#).

Se utilizzi un altro provider di hosting Web, consulta la documentazione di tale provider per ulteriori informazioni sulla pubblicazione di pagine Web.

## Test del modulo

Una volta pubblicato il modulo, è necessario inviare alcuni eventi di test per verificare che funzioni come previsto.

Per eseguire il test del modulo di registrazione

1. In un browser Web, andare alla posizione in cui è stato caricato il modulo di registrazione. Se hai utilizzato l'esempio di codice di [Create the JavaScript form handler](#), vedrai un modulo simile all'esempio nell'immagine seguente.

# Register for SMS Alerts

Enter your phone number below to sign up for PromotionName messages from ExampleCorp.

We don't share your contact information with anyone else. For more information, see our [Privacy Policy](#).

ExampleCorp alerts are only available to recipients in the United States.

### First name

### Last name

### Phone number

(  )  -

Copyright © 2019, ExampleCorp or its affiliates.

2. Inserire le proprie informazioni di contatto nei campi First name (Nome), Last name (Cognome) e Phone number (Numero di telefono).

**Note**

Quando si invia il modulo, Amazon Pinpoint prova a inviare un messaggio al numero di telefono specificato. A causa di questa funzionalità, è necessario utilizzare un numero di telefono reale per eseguire il test della soluzione dall'inizio alla fine.

Se hai testato la funzione Lambda in Create [Lambda functions](#), il tuo progetto Amazon Pinpoint contiene già almeno un endpoint. Quando provi questo modulo, devi inviare un numero di telefono diverso nel modulo o eliminare l'endpoint esistente utilizzando l'operazione API. [DeleteEndpoint](#)

3. Controllare il dispositivo associato al numero di telefono specificato per verificare che abbia ricevuto il messaggio.
4. Apri la console Amazon Pinpoint all'indirizzo <https://console.aws.amazon.com/pinpoint/>.
5. Nella pagina Tutti i progetti, scegli il progetto che hai creato in [Crea un progetto Amazon Pinpoint](#).
6. Nel riquadro di navigazione, selezionare Segments (Segmenti). Nella Segments page (Pagina segmenti), selezionare Create a segment (Crea un segmento).
7. In Segment group 1 (Gruppo segmenti 1), nel campo Add filters to refine your segment (Aggiungi filtri per rifinire i segmenti), selezionare Filter by user (Filtra per utente).
8. Per Scegli un attributo utente, scegli FirstName. Quindi, in Choose values (Scegli valori), scegliere il nome specificato al momento dell'invio del modulo.

La sezione Segment estimate (Stima segmenti) dovrebbe indicare che non ci sono endpoint idonei e un endpoint (in Total endpoints (Endpoint totali)), come illustrato nella seguente immagine. Questo risultato è previsto. Quando la funzione Lambda crea un nuovo endpoint, per tale endpoint viene eseguito l'opt-out per impostazione predefinita.

The screenshot displays the Amazon Pinpoint console interface. On the left, the 'Segment group 1' configuration is shown. It includes a description, a dropdown for 'Include endpoints that are in' (set to 'any'), and a dropdown for 'of the following segments' (set to 'All segments'). Below this, there's a section for 'Endpoints that match any of the following filters:'. A filter named 'Filter 1: User' is active, with 'FirstName' selected and the value 'Carlos' entered. There are options to 'Add more attributes or metrics to this filter' and 'Add filters to refine your segment'. On the right, the 'Segment estimate' section shows '0 endpoints' and a red warning box stating 'No matches found'. Below the warning, it indicates '1 endpoints'.

9. Sul dispositivo che ha ricevuto il messaggio, rispondi al messaggio con la parola chiave SMS bidirezionale che hai specificato in [Abilita SMS bidirezionali](#). Amazon Pinpoint invia immediatamente un messaggio di risposta.
10. Nella console Amazon Pinpoint, ripeti i passaggi da 4 a 8. Questa volta, quando si crea il segmento, vengono visualizzati un endpoint idoneo e un endpoint totale. Questo risultato è previsto, poiché è stato fornito il consenso.

## Fasi successive

Completando questo tutorial hai effettuato quanto segue:

- Creazione di un progetto Amazon Pinpoint, configurazione del canale SMS e recupero di un codice lungo dedicato.
- Creazione di una policy IAM che utilizza il principale del privilegio minimo per concedere i diritti di accesso e associazione di tale policy a un ruolo.
- Creazione di due funzioni Lambda che utilizzano le operazioni PhoneNumberValidate, UpdateEndpoint e SendMessages nell'API Amazon Pinpoint.
- Creazione di una REST API utilizzando Gateway API
- Create e distribuito un modulo basato sul Web che consente di raccogliere informazioni di contatto dei clienti.
- Eseguito test sulla soluzione per verificare che funzioni come previsto.

In questa sezione vengono illustrati alcuni modi in cui è possibile utilizzare le informazioni raccolte utilizzando questa soluzione. Include anche alcuni suggerimenti su come è possibile personalizzare questa soluzione per adattarla al caso d'uso unico specifico.

## Creazione di segmenti di clientela

Tutti i dettagli del cliente raccolti tramite questo modulo vengono memorizzati come endpoint. Questa soluzione consente di creare endpoint che contengono numerosi attributi utilizzabili a scopo di segmentazione.

Ad esempio, questa soluzione acquisisce un attributo di endpoint chiamato `Source`. Questo attributo contiene il percorso completo alla posizione in cui il modulo è stato ospitato. Quando crei un segmento, puoi filtrarlo per endpoint, quindi perfezionare ulteriormente il filtro scegliendo un attributo `Source`.

La creazione di segmenti in base all'attributo `Source` può essere utile in diversi modi. In primo luogo, ti consente di creare rapidamente un segmento di clienti che già si sono registrati per ricevere i tuoi SMS. Inoltre, lo strumento di segmentazione in Amazon Pinpoint esclude automaticamente gli endpoint che non hanno eseguito l'opt-in per la ricezione dei messaggi.

L'attributo `Source` è utile anche se decidi di ospitare il modulo di registrazione in diverse posizioni. Ad esempio, il materiale di marketing potrebbe fare riferimento a un modulo ospitato in un'unica posizione, mentre i clienti che trovano il modulo mentre navigano sul tuo sito Web potrebbero vedere una versione ospitata altrove. In questo modo, gli attributi di origine per i clienti che completano il modulo dopo aver visto i materiali di marketing sono differenti da quelli che completano il modulo dopo averlo trovato sul tuo sito Web. Puoi usare questa differenza per creare segmenti distinti e quindi inviare comunicazioni personalizzate a ognuno di questi gruppi di destinatari.

## Invio di messaggi personalizzati per campagne

Dopo aver creato i segmenti, puoi iniziare a inviare campagne a tali segmenti. Quando si creano messaggi per le campagne, è possibile personalizzarli specificando gli attributi di endpoint da includere nel messaggio. Ad esempio, il modulo Web utilizzato in questa soluzione richiede al cliente di immettere il proprio nome e cognome. Questi valori vengono memorizzati nel record utente associato all'endpoint.

Ad esempio, se utilizzi l'operazione API `GetEndpoint` per recuperare informazioni su un endpoint che è stato creato utilizzando questa soluzione, vedrai una sezione simile alla seguente:

...

```
"User": {
  "UserAttributes": {
    "FirstName": [
      "Carlos"
    ],
    "LastName": [
      "Salazar"
    ]
  }
}
...
```

Se vuoi includere i valori di questi attributi nel messaggio della tua campagna, puoi utilizzare la notazione punto per fare riferimento all'attributo. Quindi racchiudere l'intero riferimento in parentesi graffe doppie. Ad esempio, per inserire il nome di ciascun destinatario nel messaggio di una campagna, includere la seguente stringa nel messaggio: `{{User.UserAttributes.FirstName}}`. Quando Amazon Pinpoint invia il messaggio, sostituisce la stringa con il valore dell'attributo `FirstName`.

## Uso del modulo per raccogliere informazioni aggiuntive

È possibile modificare questa soluzione per raccogliere informazioni aggiuntive per il modulo di registrazione. Ad esempio, è possibile chiedere al cliente di fornire il proprio indirizzo e quindi utilizzare i dati dell'indirizzo per popolare i campi `Location.City`, `Location.Country`, `Location.Region` e `Location.PostalCode` nella risorsa `Endpoint`. La raccolta di informazioni relative all'indirizzo nel modulo di registrazione potrebbe far sì che gli endpoint contengano informazioni più dettagliate. Per apportare questa modifica, è necessario aggiungere i campi appropriati al modulo Web. È inoltre necessario modificare il codice JavaScript per il modulo per trasferire i nuovi valori. Infine, è necessario modificare la funzione Lambda che crea l'endpoint per gestire le nuove informazioni in entrata.

È anche possibile modificare il modulo in modo che raccolga le informazioni di contatto in altri canali. Ad esempio, è possibile usare il modulo per raccogliere l'indirizzo e-mail del cliente oltre al suo numero di telefono. Per apportare questa modifica, è necessario modificare il codice HTML e JavaScript per il modulo Web. È inoltre necessario modificare la funzione Lambda in modo che crei due endpoint distinti, ovvero uno per l'endpoint e-mail e uno per l'endpoint SMS. È inoltre necessario modificare la funzione Lambda in modo che generi un valore univoco per l'attributo `User.UserId` e quindi associ tale valore a entrambi gli endpoint.

## Registrazione di attributi aggiuntivi per finalità di audit

Questa soluzione registra due preziosi attributi quando crea e aggiorna gli endpoint. Innanzitutto, quando la prima funzione Lambda crea inizialmente l'endpoint, registra l'URL del modulo stesso nell'attributo `Attributes.Source`. Se il cliente risponde al messaggio, la seconda funzione Lambda crea un attributo `Attributes.OptInTimestamp`. Questo attributo contiene la data e l'ora esatte del momento in cui il cliente ha fornito il suo consenso a ricevere messaggi da parte tua.

Entrambi questi campi possono essere utili se un operatore telefonico o un'agenzia di regolamentazione dovesse richiedere di fornire prova del consenso del cliente. È possibile recuperare queste informazioni in qualsiasi momento utilizzando l'operazione dell'API [GetEndpoint](#).

È anche possibile modificare le funzioni Lambda per la registrazione di dati aggiuntivi che possono essere utili per finalità di audit, ad esempio l'indirizzo IP da cui è stata inviata la richiesta di registrazione.

# Integrazione di Amazon Pinpoint con le applicazioni

L'integrazione di Amazon Pinpoint con il codice client è utile per comprendere e coinvolgere gli utenti.

Dopo l'integrazione, quando gli utenti avviano l'applicazione, questa si connette al servizio Amazon Pinpoint per aggiungere o aggiornare gli endpoint. Gli endpoint rappresentano le destinazioni a cui è possibile inviare messaggi, ad esempio i dispositivi degli utenti, gli indirizzi e-mail o i numeri di telefono.

Inoltre, la tua applicazione fornisce dati relativi all'utilizzo o eventi. Visualizza i dati sugli eventi nella console Amazon Pinpoint per scoprire quanti utenti hai, con quale frequenza usano la tua applicazione, quando la utilizzano e molto altro.

Quando l'applicazione fornisce endpoint ed eventi, puoi usare queste informazioni per creare campagne di messaggistica mirate per specifici destinatari o segmenti. (È anche possibile inviare messaggi direttamente a semplici elenchi di destinatari senza creare campagne.)

Usa gli argomenti in questa sezione per integrare Amazon Pinpoint con un'applicazione Web o per dispositivi mobili. Questi argomenti includono esempi di codice e procedure per l'integrazione con un' JavaScriptapplicazione Android, Swift o Flutter. Per iniziare a integrare le tue app, consulta [the section called “Connessione dell'applicazione front-end con AWS Amplify”](#).

Al di fuori del client, puoi utilizzare le versioni di [AWS SDK supportate](#) o l'[API Amazon Pinpoint](#) per importare endpoint, esportare dati sugli eventi, definire segmenti di clienti, creare ed eseguire campagne e altro ancora.

## Argomenti

- [Utilizzo di Amazon Pinpoint con un SDK AWS](#)
- [Connessione dell'applicazione front-end ad Amazon Pinpoint con AWS Amplify](#)
- [Registrazione di endpoint nell'applicazione](#)
- [Segnalazione degli eventi nell'applicazione](#)
- [Gestione delle notifiche push](#)

## Utilizzo di Amazon Pinpoint con un SDK AWS

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ esempi di codice</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI esempi di codice</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go esempi di codice</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java esempi di codice</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript esempi di codice</a>
<a href="#">SDK AWS for Kotlin</a>	<a href="#">SDK AWS for Kotlin esempi di codice</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET esempi di codice</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP esempi di codice</a>
<a href="#">AWS Tools for PowerShell</a>	<a href="#">Strumenti per esempi di PowerShell codice</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) esempi di codice</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby esempi di codice</a>
<a href="#">AWS SDK for Rust</a>	<a href="#">AWS SDK for Rust esempi di codice</a>
<a href="#">SDK AWS per SAP ABAP</a>	<a href="#">SDK AWS per SAP ABAP esempi di codice</a>
<a href="#">SDK AWS per Swift</a>	<a href="#">SDK AWS per Swift esempi di codice</a>

Per esempi specifici di Amazon Pinpoint, consulta [Esempi di codice per Amazon Pinpoint con AWS SDK](#).

### Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link [Provide feedback \(Fornisci un feedback\)](#) nella parte inferiore di questa pagina.

## Connessione dell'applicazione front-end ad Amazon Pinpoint con AWS Amplify

Usa AWS Amplify per integrare l'app con AWS. Per le app Swift, consulta la sezione relativa alle [nozioni di base](#) nella documentazione di Amplify per Swift. Per le app Android, consulta la sezione relativa alle [nozioni di base](#) nella documentazione di Amplify per SDK per Android. Per l'app React Native, consulta la sezione relativa alle [nozioni di base](#) nella documentazione di Amplify per JavaScript. Per le app Flutter, consulta la sezione relativa alle [nozioni di base](#) nella documentazione dell'SDK per Flutter. Questi argomenti consentono di:

- Configurare le risorse di back-end.
- Connetti la app alle risorse di back-end utilizzando le librerie Amplify.

Per ulteriori informazioni su come connettere un'app front-end ad Amazon Pinpoint per l'integrazione di funzionalità di analisi, messaggistica in-app e notifiche push, consulta [AWS Amplify](#).

## Approfondimenti

Hai integrato la libreria AWS Amplify con la tua applicazione. Aggiorna quindi il codice per registrare i dispositivi degli utenti come endpoint. Per informazioni, consultare [Registrazione di endpoint nell'applicazione](#).

## Registrazione di endpoint nell'applicazione

Quando un utente avvia una sessione, ad esempio lanciando l'app per dispositivi mobili, l'applicazione Web o per dispositivi mobili può registrare (o aggiornare) automaticamente un endpoint in Amazon Pinpoint. L'endpoint rappresenta il dispositivo con cui l'utente avvia la sessione. Include gli attributi che descrivono il dispositivo e può includere anche attributi personalizzati. Gli endpoint possono rappresentare anche altri metodi per comunicare con i clienti, ad esempio indirizzi e-mail o numero di cellulare.

Quando l'applicazione ha registrato endpoint, è possibile segmentare i destinatari in base agli attributi degli endpoint. Puoi quindi coinvolgere questi segmenti con campagne di messaggistica mirate. Puoi anche utilizzare la pagina **Analisi** nella console Amazon Pinpoint per visualizzare grafici relativi alla registrazione e all'attività degli endpoint, ad esempio **Nuovi endpoint** e **Endpoint attivi al giorno**.

Puoi assegnare un unico ID utente a più endpoint. Un ID utente rappresenta un singolo utente, mentre ciascun endpoint a cui viene assegnato l'ID utente rappresenta uno dei dispositivi dell'utente. Dopo avere assegnato ID utente agli endpoint, puoi visualizzare i grafici relativi all'attività degli utenti nella console, ad esempio **Utenti attivi al giorno** e **Utenti attivi mensili**.

## Prima di iniziare

Se non l'hai ancora fatto, integra l'SDK AWS Mobile per Android o iOS oppure la libreria JavaScript di AWS con l'applicazione. Per informazioni, consultare [Connessione dell'applicazione front-end ad Amazon Pinpoint con AWS Amplify](#).

## Registrazione di endpoint con gli SDK AWS per dispositivi mobili per Android o iOS

Puoi usare gli SDK AWS Mobile per Android o iOS per registrare e personalizzare gli endpoint. Per maggiori informazioni ed esempi di codice, consulta i seguenti documenti:

- [Registrazione di endpoint nell'applicazione](#) nella documentazione dell'SDK per Android.
- [Registrazione di endpoint nell'applicazione](#) nella documentazione dell'SDK per iOS.

## Registrazione di endpoint con la libreria JavaScript di AWS Amplify

È possibile utilizzare la libreria JavaScript di AWS Amplify per registrare e aggiornare gli endpoint nelle app. Per informazioni ed esempi di codice, consulta l'argomento relativo all'[aggiornamento di endpoint](#) nella documentazione di JavaScript di AWS Amplify.

## Fasi successive

Hai aggiornato l'app per registrare gli endpoint. A questo punto, quando gli utenti avviano la tua app, le informazioni sul dispositivo e gli attributi personalizzati vengono inviati ad Amazon Pinpoint. Queste informazioni possono essere utilizzate per definire i segmenti di destinatari. Nella console sono riportati i parametri relativi agli endpoint e, se del caso, agli utenti a cui sono assegnati ID utente.

Completa quindi la procedura indicata in [Segnalazione degli eventi nell'applicazione](#) per aggiornare l'app per la segnalazione dei dati di utilizzo.

## Segnalazione degli eventi nell'applicazione

Nell'applicazione Web o per dispositivi mobili, è possibile utilizzare SDK mobile AWS o l'[API Eventi di Amazon Pinpoint](#) per segnalare i dati di utilizzo o gli eventi ad Amazon Pinpoint. È possibile segnalare gli eventi per acquisire informazioni come i tempi delle sessioni, il comportamento di acquisto degli utenti, i tentativi di accesso o qualsiasi tipo di evento personalizzato sia necessario.

Quando l'applicazione segnala gli eventi, puoi visualizzare i dati analitici nella console Amazon Pinpoint. I grafici nella pagina Analytics (Analisi) offrono parametri per molti aspetti del comportamento degli utenti. Per ulteriori informazioni, consulta la sezione [Grafico di riferimento per l'analisi di Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

Per analizzare e archiviare i dati sugli eventi all'esterno di Amazon Pinpoint, puoi configurare Amazon Pinpoint per lo streaming dei dati a Amazon Kinesis. Per ulteriori informazioni, consulta [Streaming di eventi Amazon Pinpoint su Kinesis](#).

Con SDK mobile AWS e le librerie JavaScript di AWS Amplify è possibile chiamare l'API Amazon Pinpoint per segnalare i seguenti tipi di eventi:

### Eventi di sessione

Indicano quando e con quale frequenza gli utenti aprono e chiudono l'app.

Quando l'applicazione segnala gli eventi relativi alle sessioni, nella pagina Analisi della console Amazon Pinpoint vengono visualizzati i grafici Sessioni, Endpoint attivi al giorno, Percentuale di conservazione su 7 giorni e altri ancora.

### Eventi personalizzati

Si tratta di eventi non standard vengono definiti assegnando un tipo di evento personalizzato. È possibile aggiungere attributi e parametri personalizzati a un evento personalizzato.

Nella pagina Analytics (Analisi) della console, la scheda Events (Eventi) mostra i parametri per tutti gli eventi personalizzati segnalati dall'app.

### Eventi di monetizzazione

Segnalano il fatturato generato dall'applicazione e il numero di articoli acquistati dagli utenti.

Nella pagina Analytics (Analisi), la scheda Revenue (Fatturato) mostra i grafici Revenue (Fatturato), Paying users (Utenti paganti), Units sold (Unità vendute) e altri ancora.

## Eventi di autenticazione

Indicano la frequenza con cui gli utenti effettuano l'autenticazione nell'applicazione.

Nella pagina Analytics (Analisi), la scheda Users (Utenti) mostra i grafici Sign-ins (Accessi), Sign-ups (Registrazioni) e Authentication failures (Errori di autenticazione).

## Prima di iniziare

Se non lo hai ancora fatto, esegui queste operazioni:

- Integra l'app con AWS Amplify. Per informazioni, consultare [Connessione dell'applicazione front-end ad Amazon Pinpoint con AWS Amplify](#).
- Aggiorna l'applicazione per registrare gli endpoint. Per informazioni, consultare [Registrazione di endpoint nell'applicazione](#).

## Segnalazione di eventi con gli SDK AWS Mobile per Android o iOS

Per consentire a un'app per dispositivi mobili di segnalare gli eventi ad Amazon Pinpoint, è possibile utilizzare SDK mobile AWS per iOS e Android.

Per ulteriori informazioni su come aggiornare l'app per registrare e inviare eventi ad Amazon Pinpoint, consulta le pagine seguenti nella documentazione di AWS Amplify:

- [Analytics](#) nella documentazione dell'SDK per iOS
- [Analytics](#) nella documentazione dell'SDK per Android

## Segnalazione di eventi con la libreria JavaScript di AWS Amplify

Per consentire alle app JavaScript e React Native di segnalare gli eventi di utilizzo dell'applicazione ad Amazon Pinpoint, è possibile utilizzare la libreria JavaScript di AWS Amplify. Per ulteriori informazioni su come aggiornare l'app per inviare eventi ad Amazon Pinpoint, consulta l'argomento relativo alle [funzionalità di analisi](#) nella documentazione di JavaScript di AWS Amplify.

## Segnalazione di eventi utilizzando l'API Amazon Pinpoint

È possibile utilizzare l'API Amazon Pinpoint o AWS SDK per inviare eventi ad Amazon Pinpoint in blocco. Per ulteriori informazioni, consulta [Eventi](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

### Approfondimenti

Hai aggiornato l'app per segnalare gli eventi. A questo punto, quando gli utenti interagiscono con l'app, quest'ultima invia i dati di utilizzo ad Amazon Pinpoint. Questi dati sono disponibili nella console e possono essere inviati in streaming ad Amazon Kinesis.

Aggiorna quindi la tua app per gestire le notifiche push inviate con Amazon Pinpoint. Per informazioni, consultare [Gestione delle notifiche push](#).

## Gestione delle notifiche push

Gli argomenti seguenti illustrano come modificare un'app Swift, Android, React Native o Flutter affinché riceva le notifiche push inviate utilizzando Amazon Pinpoint.

### Argomenti

- [Configurazione delle notifiche push per Amazon Pinpoint](#)
- [Gestione delle notifiche push](#)

## Configurazione delle notifiche push per Amazon Pinpoint

Per configurare Amazon Pinpoint in modo da poter inviare notifiche push alle app, devi prima di tutto fornire le credenziali necessarie per autorizzare Amazon Pinpoint a inviare messaggi alla tua app. Le credenziali fornite dipendono dal sistema di notifica push che utilizzi:

- Per le app iOS, fornisci un certificato SSL, che puoi ottenere dal portale Apple Developer. Il certificato autorizza Amazon Pinpoint a inviare messaggi alla tua app tramite il servizio di notifiche push Apple (APN).
- Per le app Android, fornisci una chiave API Web, che puoi ottenere dalla console Firebase. Queste credenziali autorizzano Amazon Pinpoint a inviare messaggi alla tua app tramite Firebase Cloud Messaging.

Dopo aver recuperato le credenziali per un canale di notifiche push, è necessario creare un progetto in Amazon Pinpoint e associarlo alle credenziali per il servizio di notifiche push.

## Argomenti

- [Configurazione delle notifiche push Swift](#)
- [Configurazione delle notifiche push per Android](#)
- [Configurazione delle notifiche push di Flutter](#)
- [Configurazione delle notifiche push per React Native](#)
- [Creazione di un progetto in Amazon Pinpoint](#)

## Configurazione delle notifiche push Swift

Le notifiche push per le app iOS vengono inviate tramite il servizio di notifiche push Apple (APN). Per poter inviare notifiche push ai dispositivi iOS, è necessario creare un ID app nel portale Apple Developer e creare i certificati richiesti. Per ulteriori informazioni sull'esecuzione di questa procedura, consulta l'argomento relativo alla [configurazione dei servizi di notifiche push](#) nella documentazione di AWS Amplify.

### Utilizzo dei token APNs

Come best practice, è consigliabile sviluppare l'app in modo che i token di dispositivo dei clienti vengano rigenerati quando l'app viene reinstallata.

Se un destinatario aggiorna il dispositivo a una nuova versione principale di iOS (ad esempio, da iOS 12 a iOS 13) e successivamente reinstalla l'app, questa genera un nuovo token. Se l'app non aggiorna il token, per inviare la notifica viene utilizzato il token precedente. Di conseguenza, il servizio APNs (Apple Push Notification Service) rifiuta la notifica, perché il token ora non è valido. Quando tenti di inviare la notifica, ricevi una notifica di errore del messaggio da parte degli APNs.

## Configurazione delle notifiche push per Android

Le notifiche push per le app Android vengono inviate utilizzando Firebase Cloud Messaging (FCM), che sostituisce Google Cloud Messaging (GCM). Prima di poter inviare notifiche push ai dispositivi Android, è necessario ottenere le credenziali FCM. È quindi possibile utilizzare quelle credenziali per creare un progetto Android e avviare un'app di esempio in grado di ricevere notifiche push. Per ulteriori informazioni sull'esecuzione di questa procedura, consulta la sezione relativa alle [notifiche push](#) nella documentazione di AWS Amplify.

## Configurazione delle notifiche push di Flutter

Le notifiche push per le app Flutter vengono inviate utilizzando Firebase Cloud Messaging (FCM) per Android e il servizio APN per iOS. Per ulteriori informazioni sull'esecuzione di questa procedura, consulta la sezione relativa alle notifiche push nella [documentazione di AWS Amplify Flutter](#).

## Configurazione delle notifiche push per React Native

Le notifiche push per le app React Native vengono inviate utilizzando Firebase Cloud Messaging (FCM) per Android e il servizio APN per iOS. Per ulteriori informazioni sull'esecuzione di questa procedura, consulta la sezione relativa alle notifiche push nella documentazione di [AWS Amplify JavaScript](#).

## Creazione di un progetto in Amazon Pinpoint

In Amazon Pinpoint, un progetto è una raccolta di impostazioni, dati, campagne e segmenti che condividono uno scopo comune. Nell'API Amazon Pinpoint, i progetti sono anche definiti applicazioni. In questa sezione viene utilizzata esclusivamente la parola "progetto" per fare riferimento a questo concetto.

Per iniziare a inviare notifiche push in Amazon Pinpoint, è necessario creare un progetto. Quindi, è necessario abilitare i canali delle notifiche push da utilizzare fornendo le credenziali appropriate.

Per creare nuovi progetti e configurare i canali delle notifiche push, è possibile utilizzare la console Amazon Pinpoint. Per ulteriori informazioni, consulta [Configurazione dei canali push Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

È anche possibile creare e configurare progetti utilizzando l'[API Amazon Pinpoint](#), [AWS SDK](#) o l'[AWS Command Line Interface](#) (AWS CLI). Per creare un progetto, usa la risorsa Apps. Per configurare i canali delle notifiche push, usa le risorse seguenti:

- [Canale APN](#) per l'invio di messaggi agli utenti di dispositivi iOS mediante il servizio di notifiche push Apple (APN).
- [Canale ADM](#) per l'invio di messaggi agli utenti di Amazon Kindle Fire.
- [Canale Baidu](#) per l'invio di messaggi agli utenti di Baidu.
- [Canale GCM](#) per l'invio di messaggi a dispositivi Android mediante Firebase Cloud Messaging (FCM), che sostituisce Google Cloud Messaging (GCM).

## Gestione delle notifiche push

Dopo aver ottenuto le credenziali necessarie per inviare notifiche push, puoi aggiornare le tue app in modo che possano ricevere notifiche push. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base sulle notifiche push](#) nella documentazione AWS Amplify.

# Definizione del pubblico in Amazon Pinpoint

In Amazon Pinpoint, ogni membro del pubblico è rappresentato da uno o più endpoint. Quando usi Amazon Pinpoint per inviare un messaggio, indirizzi il messaggio agli endpoint che rappresentano i membri del pubblico di destinazione. Ogni definizione di endpoint include una destinazione del messaggio, come un token del dispositivo, un indirizzo e-mail o un numero di telefono. Include anche dati relativi agli utenti e ai loro dispositivi. Per analizzare, segmentare o coinvolgere il pubblico, occorre prima di tutto aggiungere endpoint al progetto Amazon Pinpoint.

Per aggiungere endpoint, puoi:

- Integrare Amazon Pinpoint con il client Android, iOS o JavaScript, in modo che gli endpoint vengano aggiunti automaticamente quando gli utenti visitano l'applicazione.
- Usare l'API Amazon Pinpoint per aggiungere gli endpoint singolarmente o in batch.
- Importare le definizioni di endpoint archiviate esternamente ad Amazon Pinpoint.

Dopo aver aggiunto gli endpoint, puoi:

- Visualizzare le analisi relative al pubblico nella console Amazon Pinpoint.
- Ottenere maggiori informazioni sui destinatari cercando o esportando i dati degli endpoint.
- Definire segmenti di destinatari in base agli attributi degli endpoint, ad esempio dati demografici o interessi degli utenti.
- Coinvolgere i destinatari con campagne di messaggistica mirate.
- Inviare messaggi direttamente a elenchi di endpoint.

Utilizza gli argomenti in questa sezione per aggiungere, aggiornare ed eliminare endpoint tramite l'API Amazon Pinpoint. Se vuoi aggiungere endpoint automaticamente dal tuo client Android, iOS o JavaScript, consulta [Registrazione di endpoint nell'applicazione](#) .

Argomenti

- [Aggiunta di endpoint ad Amazon Pinpoint](#)
- [Associazione di utenti agli endpoint Amazon Pinpoint](#)
- [Aggiunta di un batch di endpoint ad Amazon Pinpoint](#)
- [Importazione di endpoint in Amazon Pinpoint](#)

- [Eliminazione degli endpoint da Amazon Pinpoint](#)
- [Gestione del numero massimo di membri di un pubblico dell'endpoint](#)

## Aggiunta di endpoint ad Amazon Pinpoint

Un endpoint rappresenta una destinazione alla quale puoi inviare messaggi, ad esempio un dispositivo mobile, un numero di telefono o un indirizzo e-mail. Prima di inviare messaggi a un membro del gruppo di destinatari, è necessario definire uno o più endpoint per quella persona.

Per definire un endpoint, devi specificare il canale e l'indirizzo. Il canale è il tipo di piattaforma usata per inviare messaggi all'endpoint, ad esempio un servizio di notifica push, SMS o e-mail. L'indirizzo specifica dove inviare il messaggio all'endpoint, ad esempio un token di dispositivo, un numero di telefono o un indirizzo e-mail.

Per aggiungere ulteriori dettagli sui destinatari, puoi arricchire gli endpoint con attributi personalizzati o standard. Questi attributi possono includere dati relativi agli utenti, alle loro preferenze, ai loro dispositivi, alle versioni del client che utilizzano o alle loro posizioni. L'aggiunta di dati di questo tipo agli endpoint ti consente di:

- Visualizzare grafici relativi al pubblico nella console Amazon Pinpoint.
- Segmentare il gruppo di destinatari in base agli attributi degli endpoint, in modo da inviare messaggi al target appropriato.
- Personalizzare i messaggi incorporando variabili dei messaggi che vengono sostituite con i valori degli attributi degli endpoint.

Un'applicazione client JavaScript o per dispositivi mobili registra automaticamente gli endpoint se si integra Amazon Pinpoint utilizzando SDK mobile AWS o la libreria JavaScript di AWS Amplify. Il client registra un endpoint per ogni nuovo utente e aggiorna gli endpoint per gli utenti già registrati. Per registrare gli endpoint da un client JavaScript o per dispositivi mobili, consulta [Registrazione di endpoint nell'applicazione](#).

## Esempi

Gli esempi seguenti illustrano come aggiungere un endpoint a un progetto Amazon Pinpoint. L'endpoint rappresenta un membro del gruppo di destinatari che abita a Seattle e usa un iPhone. A questa persona è possibile inviare messaggi tramite il servizio APNs (Apple Push Notification service). L'indirizzo dell'endpoint è il token del dispositivo fornito dal servizio APNs.

## AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando per l'aggiornamento di endpoint

Per aggiungere o aggiornare un endpoint, usa il comando [update-endpoint](#):

```
$ aws pinpoint update-endpoint \  
> --application-id application-id \  
> --endpoint-id endpoint-id \  
> --endpoint-request file://endpoint-request-file.json
```

Dove:

- `application-id` è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.
- `example-endpoint` è l'ID da assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente da aggiornare.
- `endpoint-request-file.json` è il percorso di un file JSON locale che contiene l'input per il parametro `--endpoint-request`.

Example File di richiesta endpoint

Il comando `update-endpoint` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene una definizione di endpoint come la seguente:

```
{  
  "ChannelType": "APNS",  
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",  
  "Attributes": {  
    "Interests": [  
      "Technology",  
      "Music",  
      "Travel"  
    ]  
  },  
  "Metrics": {  
    "technology_interest_level": 9.0,  
    "music_interest_level": 6.0,  
    "travel_interest_level": 4.0  
  }  
}
```

```
},
"Demographic": {
  "AppVersion": "1.0",
  "Make": "apple",
  "Model": "iPhone",
  "ModelVersion": "8",
  "Platform": "ios",
  "PlatformVersion": "11.3.1",
  "Timezone": "America/Los_Angeles"
},
"Location": {
  "Country": "US",
  "City": "Seattle",
  "PostalCode": "98121",
  "Latitude": 47.61,
  "Longitude": -122.33
}
}
```

Per gli attributi che è possibile utilizzare per definire un endpoint, consulta lo schema [EndpointRequest](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per aggiungere un endpoint, inizializza un oggetto [EndpointRequest](#) e passalo al metodo [updateEndpoint](#) del client AmazonPinpoint:

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.*;
import java.util.Arrays;

public class AddExampleEndpoint {

    public static void main(String[] args) {

        final String USAGE = "\n" +
```

```
"AddExampleEndpoint - Adds an example endpoint to an Amazon Pinpoint
application." +
"Usage: AddExampleEndpoint <applicationId>" +
"Where:\n" +
"  applicationId - The ID of the Amazon Pinpoint application to add the example
" +
"endpoint to.";

if (args.length < 1) {
    System.out.println(USAGE);
    System.exit(1);
}

String applicationId = args[0];

// The device token assigned to the user's device by Apple Push Notification
// service (APNs).
String deviceToken =
"1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f";

// Initializes an endpoint definition with channel type and address.
EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
    .withChannelType(ChannelType.APNS)
    .withAddress(deviceToken);

// Adds custom attributes to the endpoint.
wangXiulansIphoneEndpoint.addAttributeEntry("interests", Arrays.asList(
    "technology",
    "music",
    "travel"));

// Adds custom metrics to the endpoint.
wangXiulansIphoneEndpoint.addMetricEntry("technology_interest_level", 9.0);
wangXiulansIphoneEndpoint.addMetricEntry("music_interest_level", 6.0);
wangXiulansIphoneEndpoint.addMetricEntry("travel_interest_level", 4.0);

// Adds standard demographic attributes.
wangXiulansIphoneEndpoint.setDemographic(new EndpointDemographic()
    .withAppVersion("1.0")
    .withMake("apple")
    .withModel("iPhone")
    .withModelVersion("8")
    .withPlatform("ios")
    .withPlatformVersion("11.3.1"))
```

```
.withTimezone("America/Los_Angeles"));

// Adds standard location attributes.
wangXiulansIphoneEndpoint.setLocation(new EndpointLocation()
    .withCountry("US")
    .withCity("Seattle")
    .withPostalCode("98121")
    .withLatitude(47.61)
    .withLongitude(-122.33));

// Initializes the Amazon Pinpoint client.
AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
    .withRegion(Regions.US_EAST_1).build();

// Updates or creates the endpoint with Amazon Pinpoint.
UpdateEndpointResult result = pinpointClient.updateEndpoint(new
UpdateEndpointRequest()
    .withApplicationId(applicationId)
    .withEndpointId("example_endpoint")
    .withEndpointRequest(wangXiulansIphoneEndpoint));

System.out.format("Update endpoint result: %s\n",
result.getMessageBody().getMessage());

}
}
```

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta PUT per l'endpoint

Per aggiungere una richiesta di endpoint, invia una richiesta PUT alla risorsa [Endpoint](#) all'URI seguente:

```
/v1/apps/application-id/endpoints/endpoint-id
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.

- endpoint-id è l'ID che vuoi assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente che vuoi aggiornare.

Nella richiesta includi le intestazioni necessarie e specifica lo schema [EndpointRequest](#) in formato JSON come corpo.

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
}
```

```
"Location": {
  "Country": "US",
  "City": "Seattle",
  "PostalCode": "98121",
  "Latitude": 47.61,
  "Longitude": -122.33
}
```

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
  "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
  "Message": "Accepted"
}
```

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta [Endpoint](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

Per ulteriori informazioni sulla personalizzazione dei messaggi con variabili, consulta la sezione relativa alle [variabili dei messaggi](#) nella Guida per l'utente di Amazon Pinpoint.

Per informazioni sulle quote applicabili agli endpoint, ad esempio il numero di attributi che è possibile assegnare, consulta [the section called "Quote per gli endpoint"](#).

## Associazione di utenti agli endpoint Amazon Pinpoint

Un endpoint può includere attributi che definiscono un utente, che rappresenta una persona nel gruppo di destinatari. Ad esempio, un utente può rappresentare una persona che ha installato la tua app per dispositivi mobili o che ha un account sul tuo sito Web.

Per definire un utente è necessario specificare un ID utente univoco e, facoltativamente, gli attributi utente personalizzati. Se una persona usa la tua app su più dispositivi o se è contattabile tramite messaggio a più indirizzi, puoi assegnare lo stesso ID utente a più endpoint. In questo caso, Amazon Pinpoint sincronizza gli attributi utente negli endpoint. Quindi, se aggiungi un attributo utente a un

endpoint, Amazon Pinpoint aggiunge tale attributo a tutti gli endpoint che condividono lo stesso ID utente.

Puoi aggiungere attributi utente per tenere traccia dei dati applicabili a una persona che non variano in base al dispositivo usato da quella persona. Ad esempio, è possibile aggiungere attributi per il nome, l'età o lo stato dell'account di una persona.

### Tip

Se l'applicazione usa pool di utenti di Amazon Cognito per gestire l'autenticazione degli utenti, Amazon Cognito può aggiungere automaticamente gli ID utente e gli attributi agli endpoint. Per il valore dell'ID utente degli endpoint, Amazon Cognito assegna il valore sub assegnato all'utente nel pool di utenti. Per ulteriori informazioni sull'aggiunta di utenti con [, consulta](#) Utilizzo dell'analisi dei dati di Amazon Pinpoint con i pool di utenti di Amazon Cognito nella Guida per gli sviluppatori di Amazon Cognito.

Dopo aver aggiunto le definizioni degli utenti agli endpoint, puoi segmentare il gruppo di destinatari in altri modi. Puoi definire un segmento in base agli attributi utente oppure importando un elenco di ID utente. Quando invii un messaggio a un segmento basato sugli utenti, le potenziali destinazioni includono tutti gli endpoint associati a ogni utente nel segmento.

Puoi anche definire in altre opzioni per l'invio di messaggi ai destinatari. Puoi usare una campagna per inviare messaggi a un segmento di utenti oppure inviare un messaggio direttamente a un elenco di ID utente. Per personalizzare il messaggio, puoi includere variabili dei messaggi che vengono sostituite con i valori degli attributi utente.

## Esempi

Gli esempi seguenti illustrano come aggiungere una definizione utente a un endpoint.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando per l'aggiornamento di endpoint

Per aggiungere un utente a un endpoint, usa il comando [update-endpoint](#). Per il parametro `--endpoint-request` è possibile definire un nuovo endpoint, che può includere un utente. In

alternativa, per aggiornare un endpoint esistente, è possibile specificare solo gli attributi da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente specificando solo gli attributi utente:

```
$ aws pinpoint update-endpoint \  
> --application-id application-id \  
> --endpoint-id endpoint-id \  
> --endpoint-request file://endpoint-request-file.json
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.
- *endpoint-id* è l'ID che vuoi assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente che vuoi aggiornare.
- *endpoint-request-file.json* è il percorso di un file JSON locale che contiene l'input per il parametro `--endpoint-request`.

#### Example File di richiesta endpoint

Il comando `update-endpoint` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene una definizione utente come la seguente:

```
{  
  "User":{  
    "UserId":"example_user",  
    "UserAttributes":{  
      "FirstName":["Wang"],  
      "LastName":["Xiulan"],  
      "Gender":["Female"],  
      "Age":["39"]  
    }  
  }  
}
```

Per gli attributi che è possibile utilizzare per definire un utente, fai riferimento all'oggetto `User` nello schema [EndpointRequest](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per aggiungere un utente, inicializza un oggetto [EndpointRequest](#) e passalo al metodo [updateEndpoint](#) del client AmazonPinpoint. Puoi usare questo oggetto per definire un nuovo endpoint, che può includere un utente. In alternativa, per aggiornare un endpoint esistente, è possibile aggiornare solo le proprietà da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente aggiungendo un oggetto [EndpointUser](#) all'oggetto EndpointRequest:

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
public static void updatePinpointEndpoint(PinpointClient pinpoint, String
applicationId, String endPointId) {
    try {
        List<String> wangXiList = new ArrayList<>();
        wangXiList.add("cooking");
        wangXiList.add("running");
        wangXiList.add("swimming");

        Map myMapWang = new HashMap<>();
        myMapWang.put("interests", wangXiList);

        List<String> myNameWang = new ArrayList<>();
        myNameWang.add("Wang ");
        myNameWang.add("Xiulan");

        Map wangName = new HashMap<>();
```

```
wangName.put("name", myNameWang);

EndpointUser wangMajor = EndpointUser.builder()
    .userId("example_user_10")
    .userAttributes(wangName)
    .build();

// Create an EndpointBatchItem object for Mary Major.
EndpointRequest wangXiulanEndpoint = EndpointRequest.builder()
    .channelType(ChannelType.EMAIL)
    .address("wang_xiulan@example.com")
    .attributes(myMapWang)
    .user(wangMajor)
    .build();

// Adds multiple endpoint definitions to a single request object.
UpdateEndpointRequest endpointList = UpdateEndpointRequest.builder()
    .applicationId(applicationId)
    .endpointRequest(wangXiulanEndpoint)
    .endpointId(endpointId)
    .build();

UpdateEndpointResponse result = pinpoint.updateEndpoint(endpointList);
System.out.format("Update endpoint result: %s\n",
result.messageBody().message());

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Per l'esempio completo dell'SDK, consulta [AddExampleUser.java](#) su [GitHub](#).

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta Put per endpoint con definizione utente

Per aggiungere un utente a un endpoint, invia una richiesta PUT alla risorsa [Endpoint](#) all'URI seguente:

```
/v1/apps/application-id/endpoints/endpoint-id
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.
- *endpoint-id* è l'ID che vuoi assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente che vuoi aggiornare.

Nella richiesta includi le intestazioni necessarie e specifica lo schema [EndpointRequest](#) in formato JSON come corpo. Il corpo della richiesta può definire un nuovo endpoint, che può includere un utente. In alternativa, per aggiornare un endpoint esistente, è possibile specificare solo gli attributi da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente specificando solo gli attributi utente:

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-type;host;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "User":{
    "UserId":"example_user",
    "UserAttributes":{
      "FirstName":"Wang",
      "LastName":"Xiulan",
      "Gender":"Female",
      "Age":"39"
    }
  }
}
```

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
  "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
  "Message": "Accepted"
```

```
}
```

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta [Endpoint](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

Per ulteriori informazioni sulla personalizzazione dei messaggi con variabili, consulta la sezione relativa alle [variabili dei messaggi](#) nella Guida per l'utente di Amazon Pinpoint.

Per informazioni su come definire un segmento importando un elenco di ID utente, consulta [Importazione di segmenti](#) nella Guida per l'utente di Amazon Pinpoint.

Per informazioni sull'invio di un messaggio diretto a un massimo di 100 ID utente, consulta la sezione relativa ai [messaggi utenti](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

Per informazioni sulle quote applicabili agli endpoint, ad esempio il numero di attributi utente che è possibile assegnare, consulta [the section called "Quote per gli endpoint"](#).

## Aggiunta di un batch di endpoint ad Amazon Pinpoint

È possibile aggiungere o aggiornare più endpoint in un'unica operazione fornendo gli endpoint in batch. Ogni richiesta batch può includere fino a 100 definizioni di endpoint.

Se vuoi aggiungere o aggiornare più di 100 endpoint in un'unica operazione, consulta [Importazione di endpoint in Amazon Pinpoint](#).

## Esempi

Gli esempi seguenti illustrano come aggiungere due endpoint contemporaneamente includendoli in una richiesta batch.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando per l'aggiornamento di endpoint in batch

Per inviare una richiesta di endpoint in batch, usa il comando [update-endpoints-batch](#):

```
$ aws pinpoint update-endpoints-batch \  
> --application-id application-id \  
> --endpoint-batch-request file://endpoint_batch_request_file.json
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare gli endpoint.
- *endpoint\_batch\_request\_file.json* è il percorso di un file JSON locale che contiene l'input per il parametro `--endpoint-batch-request`.

### Example File di richiesta di batch di endpoint

Il comando `update-endpoints-batch` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene un batch di definizioni di endpoint come il seguente:

```
{  
  "Item": [  
    {  
      "ChannelType": "EMAIL",  
      "Address": "richard_roe@example.com",  
      "Attributes": {  
        "Interests": [  
          "Music",  
          "Books"  
        ]  
      },  
      "Metrics": {  
        "music_interest_level": 3.0,  
        "books_interest_level": 7.0  
      },  
      "Id": "example_endpoint_1",  
      "User": {  
        "UserId": "example_user_1",  
        "UserAttributes": {  
          "FirstName": "Richard",  
          "LastName": "Roe"  
        }  
      }  
    }  
  ],  
}
```

```
{
  "ChannelType": "SMS",
  "Address": "+16145550100",
  "Attributes": {
    "Interests": [
      "Cooking",
      "Politics",
      "Finance"
    ]
  },
  "Metrics": {
    "cooking_interest_level": 5.0,
    "politics_interest_level": 8.0,
    "finance_interest_level": 4.0
  },
  "Id": "example_endpoint_2",
  "User": {
    "UserId": "example_user_2",
    "UserAttributes": {
      "FirstName": "Mary",
      "LastName": "Major"
    }
  }
}
]
```

Per gli attributi che è possibile utilizzare per definire un batch di endpoint, consulta lo schema [EndpointBatchRequest](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per inviare una richiesta di endpoint in batch, inizializza un oggetto [EndpointBatchRequest](#) e passalo al metodo [updateEndpointsBatch](#) del client AmazonPinpoint. L'esempio seguente popola un oggetto EndpointBatchRequest con due oggetti EndpointBatchItem:

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
```

```
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

    public static void main(String[] args) {
        final String usage = ""

            Usage:    <appId>

            Where:
                appId - The ID of the application.
```

```
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String applicationId = args[0];
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    updateEndpointsViaBatch(pinpoint, applicationId);
    pinpoint.close();
}

public static void updateEndpointsViaBatch(PinpointClient pinpoint, String
applicationId) {
    try {
        List<String> myList = new ArrayList<>();
        myList.add("music");
        myList.add("books");

        Map myMap = new HashMap<String, List>();
        myMap.put("attributes", myList);

        List<String> myNames = new ArrayList<String>();
        myList.add("Richard");
        myList.add("Roe");

        Map myMap2 = new HashMap<String, List>();
        myMap2.put("name", myNames);

        EndpointUser richardRoe = EndpointUser.builder()
            .userId("example_user_1")
            .userAttributes(myMap2)
            .build();

        // Create an EndpointBatchItem object for Richard Roe.
        EndpointBatchItem richardRoesEmailEndpoint =
EndpointBatchItem.builder()
            .channelType(ChannelType.EMAIL)
            .address("richard_roe@example.com")
```

```
                .id("example_endpoint_1")
                .attributes(myMap)
                .user(richardRoe)
                .build();

List<String> myListMary = new ArrayList<String>();
myListMary.add("cooking");
myListMary.add("politics");
myListMary.add("finance");

Map myMapMary = new HashMap<String, List>();
myMapMary.put("interests", myListMary);

List<String> myNameMary = new ArrayList<String>();
myNameMary.add("Mary ");
myNameMary.add("Major");

Map maryName = new HashMap<String, List>();
myMapMary.put("name", myNameMary);

EndpointUser maryMajor = EndpointUser.builder()
    .userId("example_user_2")
    .userAttributes(maryName)
    .build();

// Create an EndpointBatchItem object for Mary Major.
EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
    .channelType(ChannelType.SMS)
    .address("+16145550100")
    .id("example_endpoint_2")
    .attributes(myMapMary)
    .user(maryMajor)
    .build();

// Adds multiple endpoint definitions to a single request
object.

EndpointBatchRequest endpointList =
EndpointBatchRequest.builder()
    .item(richardRoesEmailEndpoint)
    .item(maryMajorsSmsEndpoint)
    .build();

// Create the UpdateEndpointsBatchRequest.
```

```
UpdateEndpointsBatchRequest batchRequest =
UpdateEndpointsBatchRequest.builder()
    .applicationId(applicationId)
    .endpointBatchRequest(endpointList)
    .build();

// Updates the endpoints with Amazon Pinpoint.
UpdateEndpointsBatchResponse result =
pinpoint.updateEndpointsBatch(batchRequest);
    System.out.format("Update endpoints batch result: %s\n",
result.messageBody().message());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Per l'esempio completo dell'SDK, consulta [AddExampleEndpoints.java](#) su [GitHub](#).

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

### Example Richiesta Put per endpoint

Per inviare una richiesta di endpoint in batch, invia una richiesta PUT alla risorsa [Endpoints](#) all'URI seguente:

```
/v1/apps/application-id/endpoints
```

Dove *application-id* è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare gli endpoint.

Nella richiesta includi le intestazioni necessarie e specifica lo schema [EndpointBatchRequest](#) in formato JSON come corpo:

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-type;host;x-amz-date,
Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

```
{
  "Item": [
    {
      "ChannelType": "EMAIL",
      "Address": "richard_roe@example.com",
      "Attributes": {
        "Interests": [
          "Music",
          "Books"
        ]
      },
      "Metrics": {
        "music_interest_level": 3.0,
        "books_interest_level": 7.0
      },
      "Id": "example_endpoint_1",
      "User": {
        "UserId": "example_user_1",
        "UserAttributes": {
          "FirstName": "Richard",
          "LastName": "Roe"
        }
      }
    },
    {
      "ChannelType": "SMS",
      "Address": "+16145550100",
      "Attributes": {
        "Interests": [
          "Cooking",
          "Politics",
          "Finance"
        ]
      },
      "Metrics": {
        "cooking_interest_level": 5.0,
        "politics_interest_level": 8.0,
        "finance_interest_level": 4.0
      }
    }
  ]
}
```

```
    },
    "Id": "example_endpoint_2",
    "User": {
      "UserId": "example_user_2",
      "UserAttributes": {
        "FirstName": "Mary",
        "LastName": "Major"
      }
    }
  ]
}
```

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
  "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
  "Message": "Accepted"
}
```

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta [Endpoint](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## Importazione di endpoint in Amazon Pinpoint

È possibile aggiungere o aggiornare numerosi endpoint importandoli da un bucket Amazon S3. L'importazione di endpoint è utile se disponi di record relativi al pubblico all'esterno di Amazon Pinpoint e vuoi aggiungere queste informazioni a un progetto Amazon Pinpoint. In questo caso, è necessario:

1. Creare definizioni di endpoint basate sui dati dei destinatari.
2. Salvare queste definizioni di endpoint in uno o più file e caricare i file in un bucket Amazon S3.
3. Aggiungere gli endpoint a un progetto Amazon Pinpoint importandoli da un bucket.

Ogni processo di importazione può trasferire fino a 1 GB di dati. In un processo tipico, in cui ogni endpoint è da 4 KB o meno, è possibile importare circa 250.000 endpoint. Si possono eseguire fino a due processi di importazione simultanei per ogni account AWS. Se ti serve più larghezza di banda per i processi di importazione, puoi inviare una richiesta di aumento della quota di servizio a AWS Support. Per ulteriori informazioni, consulta [Richiesta di aumento delle quote](#).

## Prima di iniziare

Prima di importare endpoint, devi disporre delle seguenti risorse nel tuo account AWS:

- Un bucket Amazon S3. Per creare un bucket, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
- Un ruolo AWS Identity and Access Management (IAM) che conceda ad Amazon Pinpoint le autorizzazioni di lettura per il bucket Amazon S3. Per creare il ruolo, consulta [Ruolo IAM per l'importazione di endpoint o segmenti](#).

## Esempi

Gli esempi seguenti illustrano come aggiungere definizioni di endpoint a un bucket Amazon S3 e quindi importare tali endpoint in un progetto Amazon Pinpoint.

### File con definizioni di endpoint

I file che aggiungi al tuo bucket Amazon S3 possono contenere definizioni di endpoint in formato CSV o in formato JSON delimitato da nuove righe. Per gli attributi che è possibile utilizzare per definire un endpoint, consulta lo schema JSON [EndpointRequest](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

### CSV

È possibile importare gli endpoint definiti in un file CSV, come nell'esempio seguente:

```
ChannelType,Address,Location.Country,Demographic.Platform,Demographic.Make,User.UserId
SMS,12065550182,CN,Android,LG,example-user-id-1
APNS,1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f,US,iOS,Apple,example-user-id-2
EMAIL,john.stiles@example.com,US,iOS,Apple,example-user-id-2
```

La prima riga è l'intestazione, che contiene gli attributi degli endpoint. Specifica gli attributi nidificati usando la notazione punto, come in `Location.Country`.

Le righe successive definiscono gli endpoint fornendo valori per ognuno degli attributi nell'intestazione.

Per includere una virgola o le virgolette in un valore, racchiudi il valore tra virgolette, come in "aaa,bbb".

Le interruzioni di riga non sono supportate all'interno di un valore nel file CSV.

## JSON

È possibile importare gli endpoint definiti in un file JSON delimitato da nuove righe, come nell'esempio seguente:

```
{"ChannelType":"SMS","Address":"12065550182","Location":
{"Country":"CN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

In questo formato, ogni riga è un oggetto JSON completo che contiene una singola definizione di endpoint.

## Richieste al processo di importazione

I seguenti esempi illustrano come aggiungere definizioni di endpoint a Amazon S3 caricando un file locale in un bucket. Quindi, gli esempi importano le definizioni di endpoint in un progetto Amazon Pinpoint.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

#### Example Comando S3 CP

Per caricare un file locale in un bucket Amazon S3, utilizza il comando Amazon S3: [cp](#)

```
$ aws s3 cp ./endpoints-file s3://bucket-name/prefix/
```

Dove:

- `./endpoints-file` è il percorso di un file locale che contiene le definizioni di endpoint.
- `bucket-name/prefix/` è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/imports/endpoints/`.

Example Comando per la creazione di un processo di importazione

Per importare le definizioni di endpoint da un bucket Amazon S3, usa il comando [create-import-job](#):

```
$ aws pinpoint create-import-job \  
> --application-id application-id \  
> --import-job-request \  
> S3Url=s3://bucket-name/prefix/key,\  
> RoleArn=iam-import-role-arn,\  
> Format=format,\  
> RegisterEndpoints=true
```

Dove:

- `application-id` è l'ID del progetto Amazon Pinpoint per cui stai importando gli endpoint.
- `bucket-name/prefix/key` è la posizione in Amazon S3 contenente uno o più oggetti da importare. La posizione può terminare con la chiave per un singolo oggetto oppure con un prefisso che qualifica più oggetti.
- `iam-import-role-arn` è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.
- `format` può essere JSON o CSV, a seconda del formato utilizzato per definire gli endpoint. Se la posizione Amazon S3 include più oggetti di formati misti, Amazon Pinpoint importa solo gli oggetti che corrispondono al formato specificato.
- `RegisterEndpoints` può essere `true` o `false`. Se impostato su `true`, il processo di importazione registra gli endpoint con Amazon Pinpoint quando vengono importate le definizioni di endpoint.

## Combinazioni RegisterEndpoints e DefineSegments

RegisterEndpoints	DefineSegments	Descrizione
true	true	Amazon Pinpoint importerà gli endpoint e creerà un segmento che li contiene.
true	false	Amazon Pinpoint importerà gli endpoint e non creerà un segmento.
false	true	Amazon Pinpoint importerà gli endpoint e creerà un segmento che li contiene. Gli endpoint non verranno salvati e non sovrascriveranno gli endpoint esistenti.
false	false	Amazon Pinpoint rifiuterà questa richiesta.

La risposta include i dettagli sul processo di importazione:

```
{
  "ImportJobResponse": {
    "CreationDate": "2018-05-24T21:26:33.995Z",
    "Definition": {
      "DefineSegment": false,
      "ExternalId": "463709046829",
      "Format": "JSON",
      "RegisterEndpoints": true,
      "RoleArn": "iam-import-role-arn",
      "S3Url": "s3://bucket-name/prefix/key"
    },
    "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
    "JobStatus": "CREATED",
    "Type": "IMPORT"
  }
}
```

```
}  
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di importazione.

Example Comando Get per processo di importazione

Per verificare lo stato corrente di un processo di importazione, usa il comando `get-import-job`:

```
$ aws pinpoint get-import-job \  
> --application-id application-id \  
> --job-id job-id
```

Dove:

- `application-id` è l'ID del progetto Amazon Pinpoint per cui è stato avviato il processo di importazione.
- `job-id` è l'ID del processo di importazione da verificare.

La risposta a questo comando fornisce lo stato corrente del processo di importazione:

```
{  
  "ImportJobResponse": {  
    "ApplicationId": "application-id",  
    "CompletedPieces": 1,  
    "CompletionDate": "2018-05-24T21:26:45.308Z",  
    "CreationDate": "2018-05-24T21:26:33.995Z",  
    "Definition": {  
      "DefineSegment": false,  
      "ExternalId": "463709046829",  
      "Format": "JSON",  
      "RegisterEndpoints": true,  
      "RoleArn": "iam-import-role-arn",  
      "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"  
    },  
    "FailedPieces": 0,  
    "Id": "job-id",  
    "JobStatus": "COMPLETED",  
    "TotalFailures": 0,  
    "TotalPieces": 1,  
    "TotalProcessed": 3,  
  },  
}
```

```
        "Type": "IMPORT"  
    }  
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`.

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per caricare un file con le definizioni di endpoint in Amazon S3, usa il metodo [putObject](#) del client AmazonS3.

Per importare endpoint in un progetto Amazon Pinpoint, inizializza un oggetto [CreateImportJobRequest](#). Passa quindi questo oggetto al metodo [createImportJob](#) del client `AmazonPinpoint`.

```
package com.amazonaws.examples.pinpoint;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.regions.Regions;  
import com.amazonaws.services.pinpoint.AmazonPinpoint;  
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;  
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;  
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;  
import com.amazonaws.services.pinpoint.model.Format;  
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;  
import com.amazonaws.services.pinpoint.model.GetImportJobResult;  
import com.amazonaws.services.pinpoint.model.ImportJobRequest;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.AmazonS3Exception;  
import java.io.File;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.util.List;  
import java.util.concurrent.TimeUnit;  
  
public class ImportEndpoints {
```

```
public static void main(String[] args) {

    final String USAGE = "\n" +
        "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
by: \n" +
        "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
+
        "2.) Importing the endpoint definitions from the bucket to an Amazon
Pinpoint " +
        "application.\n\n" +
        "Usage: ImportEndpoints <endpointsFileLocation> <s3BucketName>
<iamImportRoleArn> " +
        "<applicationId>\n\n" +
        "Where:\n" +
        " endpointsFileLocation - The relative location of the JSON file
that contains the " +
        "endpoint definitions.\n" +
        " s3BucketName - The name of the Amazon S3 bucket to upload the
JSON file to. If the " +
        "bucket doesn't exist, a new bucket is created.\n" +
        " iamImportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint read " +
        "permissions to the S3 bucket.\n" +
        " applicationId - The ID of the Amazon Pinpoint application to add
the endpoints to.";

    if (args.length < 1) {
        System.out.println(USAGE);
        System.exit(1);
    }

    String endpointsFileLocation = args[0];
    String s3BucketName = args[1];
    String iamImportRoleArn = args[2];
    String applicationId = args[3];

    Path endpointsFilePath = Paths.get(endpointsFileLocation);
    File endpointsFile = new
File(endpointsFilePath.toAbsolutePath().toString());
    uploadToS3(endpointsFile, s3BucketName);

    importToPinpoint(endpointsFile.getName(), s3BucketName, iamImportRoleArn,
applicationId);
}
```

```
}

private static void uploadToS3(File endpointsFile, String s3BucketName) {

    // Initializes Amazon S3 client.
    final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

    // Checks whether the specified bucket exists. If not, attempts to create
one.
    if (!s3.doesBucketExistV2(s3BucketName)) {
        try {
            s3.createBucket(s3BucketName);
            System.out.format("Created S3 bucket %s.\n", s3BucketName);
        } catch (AmazonS3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Uploads the endpoints file to the bucket.
    String endpointsFileName = endpointsFile.getName();
    System.out.format("Uploading %s to S3 bucket %s . . .\n", endpointsFileName,
s3BucketName);
    try {
        s3.putObject(s3BucketName, "imports/" + endpointsFileName,
endpointsFile);
        System.out.println("Finished uploading to S3.");
    } catch (AmazonServiceException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}

private static void importToPinpoint(String endpointsFileName, String
s3BucketName,
    String iamImportRoleArn, String applicationId) {

    // The S3 URL that Amazon Pinpoint requires to find the endpoints file.
    String s3Url = "s3://" + s3BucketName + "/imports/" + endpointsFileName;

    // Defines the import job that Amazon Pinpoint runs.
    ImportJobRequest importJobRequest = new ImportJobRequest()
        .withS3Url(s3Url)
        .withRegisterEndpoints(true)
}
```

```
        .withRoleArn(iamImportRoleArn)
        .withFormat(Format.JSON);
CreateImportJobRequest createImportJobRequest = new CreateImportJobRequest()
        .withApplicationId(applicationId)
        .withImportJobRequest(importJobRequest);

// Initializes the Amazon Pinpoint client.
AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
        .withRegion(Regions.US_EAST_1).build();

System.out.format("Importing endpoints in %s to Amazon Pinpoint application
%s . . .\n",
        endpointsFileName, applicationId);

try {

    // Runs the import job with Amazon Pinpoint.
    CreateImportJobResult importResult =
pinpointClient.createImportJob(createImportJobRequest);

    String jobId = importResult.getImportJobResponse().getId();
    GetImportJobResult getImportJobResult = null;
    String jobStatus = null;

    // Checks the job status until the job completes or fails.
    do {
        getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                .withJobId(jobId)
                .withApplicationId(applicationId));
        jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
        System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
        TimeUnit.SECONDS.sleep(3);
    } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));

    if (jobStatus.equals("COMPLETED")) {
        System.out.println("Finished importing endpoints.");
    } else {
        System.err.println("Failed to import endpoints.");
        System.exit(1);
    }

    // Checks for entries that failed to import.
```

```

        // getFailures provides up to 100 of the first failed entries for the
job, if
        // any exist.
        List<String> failedEndpoints =
getImportJobResult.getImportJobResponse().getFailures();
        if (failedEndpoints != null) {
            System.out.println("Failed to import the following entries:");
            for (String failedEndpoint : failedEndpoints) {
                System.out.println(failedEndpoint);
            }
        }

    } catch (AmazonServiceException | InterruptedException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }

}
}
}
}

```

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

### Example Richiesta PUT Object di S3

Per aggiungere le definizioni di endpoint a un bucket, utilizza l'operazione [oggetto PUT](#) di Amazon S3 e fornisci le definizioni di endpoint come corpo:

```

PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
    c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
    Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

```

```
{
  "ChannelType": "SMS", "Address": "2065550182", "Location": {
    "Country": "CAN"}, "Demographic": {
    "Platform": "Android", "Make": "LG"}, "User": {
    "UserId": "example-user-id-1"}}
{
  "ChannelType": "APNS", "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f", "Location": {
    "Country": "USA"}, "Demographic": {
    "Platform": "iOS", "Make": "Apple"}, "User": {
    "UserId": "example-user-id-2"}}
{
  "ChannelType": "EMAIL", "Address": "john.stiles@example.com", "Location": {
    "Country": "USA"}, "Demographic": {
    "Platform": "iOS", "Make": "Apple"}, "User": {
    "UserId": "example-user-id-2"}}
```

Dove:

- /prefix/key è il prefisso e il nome della chiave per l'oggetto che conterrà le definizioni di endpoint dopo il caricamento. È possibile utilizzare il prefisso per organizzare gerarchicamente gli oggetti. Ad esempio, un prefisso utile potrebbe essere pinpoint/imports/endpoints/.
- bucket-name è il nome del bucket Amazon S3 a cui aggiungi le definizioni di endpoint.

Example Richiesta POST per processo di importazione

Per importare le definizioni di endpoint da un bucket Amazon S3, invia una richiesta POST alla risorsa [Processi di importazione](#). Nella richiesta includi le intestazioni necessarie e specifica lo schema [ImportJobRequest](#) in formato JSON come corpo:

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3Url": "s3://bucket-name/prefix/key",
  "RoleArn": "iam-import-role-arn",
  "Format": "format",
  "RegisterEndpoints": true
}
```

**Dove:**

- `application-id` è l'ID del progetto Amazon Pinpoint per cui stai importando gli endpoint.
- `bucket-name/prefix/key` è la posizione in Amazon S3 contenente uno o più oggetti da importare. La posizione può terminare con la chiave per un singolo oggetto oppure con un prefisso che qualifica più oggetti.
- `iam-import-role-arn` è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.
- `format` può essere JSON o CSV, a seconda del formato utilizzato per definire gli endpoint. Se la posizione Amazon S3 include più file di formati misti, Amazon Pinpoint importa solo i file corrispondenti al formato specificato.

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
  "Id": "a995ce5d70fa44adb563b7d0e3f6c6f5",
  "JobStatus": "CREATED",
  "CreationDate": "2018-06-05T21:49:15.288Z",
  "Type": "IMPORT",
  "Definition": {
    "S3Url": "s3://bucket-name/prefix/key",
    "RoleArn": "iam-import-role-arn",
    "ExternalId": "external-id",
    "Format": "JSON",
    "RegisterEndpoints": true,
    "DefineSegment": false
  }
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di importazione.

**Example Richiesta GET per processo di importazione**

Per controllare lo stato corrente di un processo di importazione, invia una richiesta GET alla risorsa [Import Job \(Processo di importazione\)](#):

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
```

```
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

Dove:

- `application_id` è l'ID del progetto Amazon Pinpoint per cui è stato avviato il processo di importazione.
- `job_id` è l'ID del processo di importazione da verificare.

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
  "ApplicationId": "application_id",
  "Id": "70a51b2cf442447492d2c8e50336a9e8",
  "JobStatus": "COMPLETED",
  "CompletedPieces": 1,
  "FailedPieces": 0,
  "TotalPieces": 1,
  "CreationDate": "2018-06-05T22:04:49.213Z",
  "CompletionDate": "2018-06-05T22:04:58.034Z",
  "Type": "IMPORT",
  "TotalFailures": 0,
  "TotalProcessed": 3,
  "Definition": {
    "S3Url": "s3://bucket-name/prefix/key.json",
    "RoleArn": "iam-import-role-arn",
    "ExternalId": "external-id",
    "Format": "JSON",
    "RegisterEndpoints": true,
    "DefineSegment": false
  }
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`.

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Processi di importazione nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta l'articolo relativo ai [processi di importazione](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## Eliminazione degli endpoint da Amazon Pinpoint

Puoi eliminare gli endpoint quando non desideri più inviare messaggi a una determinata destinazione, ad esempio quando la destinazione diventa irraggiungibile o quando un cliente chiude un account.

## Esempi

Gli esempi seguenti illustrano come eliminare un endpoint.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando di eliminazione endpoint

Per eliminare un endpoint, usa il comando [delete-endpoint](#):

```
$ aws pinpoint delete-endpoint \  
> --application-id application-id \  
> --endpoint-id endpoint-id
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- *endpoint-id* è l'ID dell'endpoint da eliminare.

La risposta a questo comando è la definizione JSON dell'endpoint eliminato.

### AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

## Example Codice

Per eliminare un endpoint, usa il metodo [deleteEndpoint](#) del client AmazonPinpoint. Specifica un oggetto [DeleteEndpointRequest](#) come argomento del metodo:

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = ""

            Usage:  <appName> <endpointId >

            Where:
                appId - The id of the application to delete.
                endpointId - The id of the endpoint to delete.
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
```

```
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        deletePinEndpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEndpoint(PinpointClient pinpoint, String appId,
String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                .applicationId(appId)
                .endpointId(endpointId)
                .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Per l'esempio completo dell'SDK, consulta [DeleteEndpoint.java](#) su [GitHub](#).

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

### Example Richiesta di eliminazione endpoint

Per eliminare un endpoint, invia una richiesta DELETE alla risorsa [Endpoint](#):

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Dove:

- application-id è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- endpoint-id è l'ID dell'endpoint da eliminare.

La risposta a questa richiesta è la definizione JSON dell'endpoint eliminato.

## Gestione del numero massimo di membri di un pubblico dell'endpoint

Ogni membro del pubblico può avere un massimo di 15 endpoint associati al proprio UserID.

Consulta [Quote per gli endpoint](#). Se provi ad aggiungere un sedicesimo endpoint, a seconda del valore dell'attributo ChannelType, otterrai BadRequestException o l'operazione avrà esito positivo se rimuovi l'endpoint con il valore EffectiveDate meno recente.

Aggiunta del sedicesimo endpoint

- Se il nuovo tipo di canale per l'endpoint è SMS, PUSH, VOICE, EMAIL, CUSTOM o IN\_APP, viene restituita un'eccezione BadRequestException perché il membro del pubblico ha raggiunto il numero massimo di endpoint. Devi rimuovere un endpoint associato al membro del pubblico e riprovare. Consulta [Eliminazione degli endpoint da Amazon Pinpoint](#).
- Se il nuovo tipo di canale per l'endpoint è ADM, GCM, APNS, APNS\_VOIP, APNS\_VOIP\_SANDBOX o BAIDU:
  - Verifica che almeno un endpoint attualmente associato al membro del pubblico abbia l'attributo ChannelType impostato su ADM, GCM, APNS, APNS\_VOICE, APNS\_VOIP\_SANDBOX o BAIDU. In caso contrario, viene restituita l'eccezione BadRequestException e un endpoint deve essere rimosso prima di riprovare. In questo caso, consulta [Eliminazione degli endpoint da Amazon Pinpoint](#).
  - In caso contrario, l'endpoint con l'attributo EffectiveDate meno recente è impostato su INACTIVE, dove l'attributo ChannelType è impostato su ADM, GCM, APNS, APNS\_VOIP, APNS\_VOIP\_SANDBOX or BAIDU.
    - L'attributo userID viene rimosso dall'endpoint meno recente.
    - Il nuovo endpoint è associato al membro del pubblico che dispone ancora del numero massimo di endpoint.

L'endpoint può essere riabilitato impostando l'attributo Status su ACTIVE e aggiungendo nuovamente l'attributo UserId all'endpoint.

# Accesso ai dati del pubblico in Amazon Pinpoint

Man mano che aggiungi endpoint in Amazon Pinpoint, viene creato un repository sempre più ricco di dati relativi al pubblico. Questi dati sono composti da:

- Endpoint aggiunti da te o aggiornati usando l'API Amazon Pinpoint.
- Endpoint aggiunti o aggiornati dal codice client man mano che si aggiungono utenti alla tua applicazione.

I dati degli endpoint crescono e cambiano di pari passo con il tuo pubblico. Per visualizzare le più recenti informazioni sul pubblico disponibili in Amazon Pinpoint, puoi cercare singoli endpoint o esportare tutti gli endpoint per un progetto Amazon Pinpoint. Visualizzando i dati degli endpoint puoi acquisire informazioni sulle caratteristiche dei destinatari registrati negli endpoint, ad esempio:

- Dispositivi e piattaforme degli utenti.
- Fusi orari degli utenti.
- Versioni delle tue app installate sui dispositivi degli utenti.
- Posizione dei clienti, ad esempio le città o i paesi in cui si trovano.
- Tutti gli attributi o i parametri personalizzati che registri.

La console Amazon Pinpoint fornisce anche l'analisi dei dati demografici e degli attributi personalizzati acquisiti negli endpoint.

Per cercare gli endpoint, devi aggiungerli a un progetto Amazon Pinpoint. Per aggiungere gli endpoint, consulta [Definizione del pubblico in Amazon Pinpoint](#).

Utilizza gli argomenti in questa sezione per cercare o esportare endpoint utilizzando l'API Amazon Pinpoint.

## Argomenti

- [Ricerca degli endpoint con Amazon Pinpoint](#)
- [Esportazione di endpoint da Amazon Pinpoint](#)
- [Elenco degli ID endpoint con Amazon Pinpoint](#)

# Ricerca degli endpoint con Amazon Pinpoint

È possibile cercare i dettagli relativi a qualsiasi endpoint aggiunto a un progetto Amazon Pinpoint. Questi dettagli possono includere l'indirizzo di destinazione per i messaggi, il canale di messaggistica, i dati relativi al dispositivo dell'utente, i dati relativi alla posizione dell'utente e qualsiasi attributo personalizzato registrato negli endpoint.

Per cercare un endpoint, è necessario l'ID endpoint. Se non conosci l'ID, puoi recuperare i dati dell'endpoint tramite l'esportazione. Per esportare gli endpoint, consulta [the section called “Esportazione di endpoint”](#).

## Esempi

Gli esempi seguenti illustrano come cercare un singolo endpoint specificando il relativo ID.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando get-endpoint

Per cercare un endpoint, usa il comando [get-endpoint](#):

```
$ aws pinpoint get-endpoint \  
> --application-id application-id \  
> --endpoint-id endpoint-id
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- *endpoint-id* è l'ID dell'endpoint da cercare.

La risposta a questo comando è la definizione JSON dell'endpoint, come nell'esempio seguente:

```
{  
  "EndpointResponse": {  
    "Address":  
    "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",  
    "ApplicationId": "application-id",  
    "Attributes": {  
      "Interests": [  

```

```
        "Technology",
        "Music",
        "Travel"
    ]
},
"ChannelType": "APNS",
"CohortId": "63",
"CreationDate": "2018-05-01T17:31:01.046Z",
"Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
},
"EffectiveDate": "2018-05-07T19:03:29.963Z",
"EndpointStatus": "ACTIVE",
"Id": "example_endpoint",
"Location": {
    "City": "Seattle",
    "Country": "US",
    "Latitude": 47.6,
    "Longitude": -122.3,
    "PostalCode": "98121"
},
"Metrics": {
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0,
    "technology_interest_level": 9.0
},
"OptOut": "ALL",
"RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
"User": {
    "UserAttributes": {
        "Gender": "Female",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "Age": "39"
    },
    "UserId": "example_user"
}
}
```

```
}
```

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per cercare un endpoint, inizializza un oggetto [GetEndpointRequest](#). Passa quindi questo oggetto al metodo [getEndpoint](#) del client AmazonPinpoint:

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
```

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = ""
```

```
Usage:  <appId> <endpoint>

Where:
  appId - The ID of the application to delete.
  endpoint - The ID of the endpoint.\s
        """;

if (args.length != 2) {
    System.out.println(usage);
    System.exit(1);
}

String appId = args[0];
String endpoint = args[1];
System.out.println("Looking up an endpoint point with ID: " + endpoint);
PinpointClient pinpoint = PinpointClient.builder()
    .region(Region.US_EAST_1)
    .build();

lookupPinpointEndpoint(pinpoint, appId, endpoint);
pinpoint.close();
}

public static void lookupPinpointEndpoint(PinpointClient pinpoint, String appId,
String endpoint) {
    try {
        GetEndpointRequest appRequest = GetEndpointRequest.builder()
            .applicationId(appId)
            .endpointId(endpoint)
            .build();

        GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
        EndpointResponse endResponse = result.endpointResponse();

        // Uses the Google Gson library to pretty print the endpoint JSON.
        Gson gson = new GsonBuilder()
            .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
            .setPrettyPrinting()
            .create();

        String endpointJson = gson.toJson(endResponse);
        System.out.println(endpointJson);
    }
}
```

```
    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    System.out.println("Done");
}
}
```

Per stampare i dati dell'endpoint in un formato leggibile, questo esempio usa la libreria Google GSON per convertire l'oggetto `EndpointResponse` in una stringa JSON.

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

### Example Richiesta GET per endpoint

Per cercare un endpoint, invia una richiesta GET alla risorsa [Endpoint](#):

```
GET /v1/apps/application_id/endpoints/endpoint_id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- *endpoint-id* è l'ID dell'endpoint da cercare.

La risposta a questa richiesta è la definizione JSON dell'endpoint, come nell'esempio seguente:

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "EndpointStatus": "ACTIVE",
  "OptOut": "NONE",
  "RequestId": "b720cfa8-6924-11e8-aeda-0b22e0b0fa59",
  "Location": {
    "Latitude": 47.6,
    "Longitude": -122.3,
    "PostalCode": "98121",
```

```
    "City": "Seattle",
    "Country": "US"
  },
  "Demographic": {
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Timezone": "America/Los_Angeles",
    "AppVersion": "1.0",
    "Platform": "ios",
    "PlatformVersion": "11.3.1"
  },
  "EffectiveDate": "2018-06-06T00:58:19.865Z",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "music_interest_level": 6,
    "travel_interest_level": 4,
    "technology_interest_level": 9
  },
  "User": {},
  "ApplicationId": "application_id",
  "Id": "example_endpoint",
  "CohortId": "39",
  "CreationDate": "2018-06-06T00:58:19.865Z"
}
```

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, consulta [Endpoint](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## Esportazione di endpoint da Amazon Pinpoint

Per ottenere tutte le informazioni di cui Amazon Pinpoint dispone sul pubblico, è possibile esportare le definizioni di endpoint appartenenti a un progetto. Quando esegui l'esportazione, Amazon Pinpoint

inserisce le definizioni di endpoint in un bucket Amazon S3 specificato. L'esportazione degli endpoint è utile per:

- Visualizzare i dati più recenti sugli endpoint nuovi ed esistenti che l'applicazione client ha registrato in Amazon Pinpoint.
- Sincronizzare i dati degli endpoint in Amazon Pinpoint con il sistema CRM (Customer Relationship Management) in uso.
- Creare report sui dati dei clienti o analizzarli.

## Prima di iniziare

Prima di esportare endpoint, devi disporre delle seguenti risorse nel tuo account AWS:

- Un bucket Amazon S3. Per creare un bucket, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
- Un ruolo AWS Identity and Access Management (IAM) che conceda ad Amazon Pinpoint le autorizzazioni di scrittura per il bucket Amazon S3. Per creare il ruolo, consulta [Ruolo IAM per l'esportazione di endpoint o segmenti](#).

## Esempi

Gli esempi seguenti illustrano come esportare gli endpoint da un progetto Amazon Pinpoint e quindi scaricare questi endpoint dal bucket Amazon S3.

### AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l'AWS CLI.

Example Comando per la creazione di un processo di esportazione

Per esportare gli endpoint nel tuo progetto Amazon Pinpoint, usa il comando [create-export-job](#):

```
$ aws pinpoint create-export-job \  
> --application-id application-id \  
> --export-job-request \  
> S3UrlPrefix=s3://bucket-name/prefix/\  
> RoleArn=iam-export-role-arn
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente gli endpoint.
- *bucket-name/prefix/* è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.

La risposta a questo comando fornisce dettagli sul processo di esportazione:

```
{
  "ExportJobResponse": {
    "CreationDate": "2018-06-04T22:04:20.585Z",
    "Definition": {
      "RoleArn": "iam-export-role-arn",
      "S3UrlPrefix": "s3://s3-bucket-name/prefix/"
    },
    "Id": "7390e0de8e0b462380603c5a4df90bc4",
    "JobStatus": "CREATED",
    "Type": "EXPORT"
  }
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di esportazione.

Example Comando per ottenere un processo di esportazione

Per verificare lo stato corrente di un processo di esportazione, usa il comando [get-export-job](#):

```
$ aws pinpoint get-export-job \
> --application-id application-id \
> --job-id job-id
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint da cui sono stati esportati gli endpoint.
- *job-id* è l'ID del processo da verificare.

La risposta a questo comando fornisce lo stato corrente del processo di esportazione:

```
{
  "ExportJobResponse": {
    "ApplicationId": "application-id",
    "CompletedPieces": 1,
    "CompletionDate": "2018-05-08T22:16:48.228Z",
    "CreationDate": "2018-05-08T22:16:44.812Z",
    "Definition": {},
    "FailedPieces": 0,
    "Id": "6c99c463f14f49caa87fa27a5798bef9",
    "JobStatus": "COMPLETED",
    "TotalFailures": 0,
    "TotalPieces": 1,
    "TotalProcessed": 215,
    "Type": "EXPORT"
  }
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`. Quando il valore di stato del processo è `COMPLETED`, puoi recuperare gli endpoint esportati dal bucket Amazon S3.

#### Example Comando S3 CP

Per scaricare gli endpoint esportati, usa il comando Amazon S3 [cp](#):

```
$ aws s3 cp s3://bucket-name/prefix/key.gz /local/directory/
```

Dove:

- *bucket-name/prefix/key* è la posizione del file `.gz` aggiunto da Amazon Pinpoint al bucket durante l'esportazione degli endpoint. Questo file contiene le definizioni degli endpoint esportati. Ad esempio, nell'URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv`, `PINPOINT-EXAMPLE-BUCKET` è il nome del bucket e `Exports/example.csv` è la chiave. Per ulteriori informazioni sulle Chiavi, consulta [Chiavi](#) nella Guida per l'utente di Amazon S3.
- */local/directory/* è il percorso della directory locale in cui scaricare gli endpoint.

## AWS SDK for Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK for Java.

### Example Codice

Per esportare endpoint da un progetto Amazon Pinpoint, inizializza un oggetto [CreateExportJobRequest](#). Passa quindi questo oggetto al metodo [createExportJob](#) del client `AmazonPinpoint`.

Per scaricare gli endpoint esportati da Amazon Pinpoint, usa il metodo [getObject](#) del client `AmazonS3`.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = ""

        This program performs the following steps:
```

1. Exports the endpoints to an Amazon S3 bucket.  
 2. Downloads the exported endpoints files from Amazon S3.  
 3. Parses the endpoints files to obtain the endpoint IDs and prints them.

Usage: ExportEndpoints <applicationId> <s3BucketName>  
 <iamExportRoleArn> <path>

Where:

applicationId - The ID of the Amazon Pinpoint application that has the endpoint.

s3BucketName - The name of the Amazon S3 bucket to export the JSON file to.\s

iamExportRoleArn - The ARN of an IAM role that grants Amazon Pinpoint write permissions to the S3 bucket. path - The path where the files downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).

```

    """;

    if (args.length != 4) {
        System.out.println(usage);
        System.exit(1);
    }

    String applicationId = args[0];
    String s3BucketName = args[1];
    String iamExportRoleArn = args[2];
    String path = args[3];
    System.out.println("Deleting an application with ID: " + applicationId);

    Region region = Region.US_EAST_1;
    PinpointClient pinpoint = PinpointClient.builder()
        .region(region)
        .build();

    S3Client s3Client = S3Client.builder()
        .region(region)
        .build();

    exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
    pinpoint.close();
    s3Client.close();
}

```

```
public static void exportAllEndpoints(PinpointClient pinpoint,
    S3Client s3Client,
    String applicationId,
    String s3BucketName,
    String path,
    String iamExportRoleArn) {

    try {
        List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
s3BucketName, iamExportRoleArn,
        applicationId);
        List<String> endpointFileKeys = objectKeys.stream().filter(o ->
o.endsWith(".gz"))
            .collect(Collectors.toList());
        downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static List<String> exportEndpointsToS3(PinpointClient pinpoint, S3Client
s3Client, String s3BucketName,
    String iamExportRoleArn, String applicationId) {

    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
    String endpointsKeyPrefix = "exports/" + applicationId + "_" +
dateFormat.format(new Date());
    String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
"/";

    List<String> objectKeys = new ArrayList<>();
    String key;

    try {
        // Defines the export job that Amazon Pinpoint runs.
        ExportJobRequest jobRequest = ExportJobRequest.builder()
            .roleArn(iamExportRoleArn)
            .s3UrlPrefix(s3UrlPrefix)
            .build();

        CreateExportJobRequest exportJobRequest =
CreateExportJobRequest.builder()
```

```

        .applicationId(applicationId)
        .exportJobRequest(jobRequest)
        .build();

    System.out.format("Exporting endpoints from Amazon Pinpoint application
%s to Amazon S3 " +
        "bucket %s . . .\n", applicationId, s3BucketName);

    CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
    String jobId = exportResult.exportJobResponse().id();
    System.out.println(jobId);
    printExportJobStatus(pinpoint, applicationId, jobId);

    ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
        .bucket(s3BucketName)
        .prefix(endpointsKeyPrefix)
        .build();

    // Create a list of object keys.
    ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
    List<S3Object> objects = v2Response.contents();
    for (S3Object object : objects) {
        key = object.key();
        objectKeys.add(key);
    }

    return objectKeys;

} catch (PinpointException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return null;
}

private static void printExportJobStatus(PinpointClient pinpointClient,
    String applicationId,
    String jobId) {

    GetExportJobResponse getExportJobResult;
    String status;

    try {

```

```
// Checks the job status until the job completes or fails.
GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
    .jobId(jobId)
    .applicationId(applicationId)
    .build();

do {
    getExportJobResult = pinpointClient.getExportJob(exportJobRequest);
    status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
    System.out.format("Export job %s . . .\n", status);
    TimeUnit.SECONDS.sleep(3);

    } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

    if (status.equals("COMPLETED")) {
        System.out.println("Finished exporting endpoints.");
    } else {
        System.err.println("Failed to export endpoints.");
        System.exit(1);
    }

} catch (PinpointException | InterruptedException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}

}

// Download files from an Amazon S3 bucket and write them to the path location.
public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

    String newPath;
    try {
        for (String key : objectKeys) {
            GetObjectRequest objectRequest = GetObjectRequest.builder()
                .bucket(s3BucketName)
                .key(key)
                .build();

            ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
            byte[] data = objectBytes.asByteArray();

```

```

        // Write the data to a local file.
        String fileSuffix = new
SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
        newPath = path + fileSuffix + ".gz";
        File myFile = new File(newPath);
        OutputStream os = new FileOutputStream(myFile);
        os.write(data);
    }
    System.out.println("Download finished.");

} catch (S3Exception | NullPointerException | IOException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}
}

```

Per l'esempio completo dell'SDK, consulta [ExportEndpoints.java](#) su [GitHub](#).

## HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

### Example Richiesta POST al processo di esportazione

Per esportare gli endpoint nel progetto Amazon Pinpoint, invia una richiesta POST alla risorsa [Processi di esportazione](#):

```

POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3UrlPrefix": "s3://bucket-name/prefix",
  "RoleArn": "iam-export-role-arn"
}

```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente gli endpoint.
- *bucket-name/prefix* è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.

La risposta a questa richiesta fornisce dettagli sul processo di esportazione:

```
{
  "Id": "611bdc54c75244bfa51fe7001ddb2e36",
  "JobStatus": "CREATED",
  "CreationDate": "2018-06-06T00:12:43.271Z",
  "Type": "EXPORT",
  "Definition": {
    "S3UrlPrefix": "s3://bucket-name/prefix",
    "RoleArn": "iam-export-role-arn"
  }
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di esportazione.

Example Richiesta GET al processo di esportazione

Per controllare lo stato corrente di un processo di esportazione, invia una richiesta GET alla risorsa [Export Job \(Processo di esportazione\)](#):

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint da cui sono stati esportati gli endpoint.
- *job-id* è l'ID del processo da verificare.

La risposta a questa richiesta fornisce lo stato corrente del processo di esportazione:

```
{
  "ApplicationId": "application_id",
  "Id": "job_id",
  "JobStatus": "COMPLETED",
  "CompletedPieces": 1,
  "FailedPieces": 0,
  "TotalPieces": 1,
  "CreationDate": "2018-06-06T00:12:43.271Z",
  "CompletionDate": "2018-06-06T00:13:01.141Z",
  "Type": "EXPORT",
  "TotalFailures": 0,
  "TotalProcessed": 217,
  "Definition": {}
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`. Quando il valore di stato del processo è `COMPLETED`, puoi recuperare gli endpoint esportati dal bucket Amazon S3.

## Informazioni correlate

Per ulteriori informazioni sulla risorsa Processi di esportazione nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta l'articolo relativo ai [processi di esportazione](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

## Elenco degli ID endpoint con Amazon Pinpoint

Per aggiornare o eliminare un endpoint, è necessario l'ID endpoint. Pertanto, se vuoi eseguire queste operazioni su tutti gli endpoint in un progetto Amazon Pinpoint, prima di tutto devi ottenere l'elenco di tutti gli ID endpoint appartenenti a tale progetto. In seguito, potrai eseguire iterazioni su questi ID, ad esempio per aggiungere un attributo a livello globale o eliminare tutti gli endpoint nel progetto.

L'esempio seguente usa l'AWS SDK for Java ed esegue queste operazioni:

1. Chiama il metodo `exportEndpointsToS3` di esempio dal codice di esempio in [Esportazione di endpoint da Amazon Pinpoint](#). Questo metodo esporta le definizioni di endpoint da un progetto Amazon Pinpoint. Le definizioni di endpoint vengono aggiunte come file gzip in un bucket Amazon S3.
2. Download dei file gzip esportati.
3. Legge i file gzip e ottiene l'ID endpoint della definizione JSON di ogni endpoint.
4. Stampa gli ID endpoint nella console.
5. Esegue la pulizia eliminando i file aggiunti da Amazon Pinpoint in Amazon S3.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = ""

            Usage:    <applicationId> <userId>

            Where:
```

applicationId - The ID of the Amazon Pinpoint application that has the endpoint.

userId - The user id applicable to the endpoints""";

```
    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String applicationId = args[0];
    String userId = args[1];
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    listAllEndpoints(pinpoint, applicationId, userId);
    pinpoint.close();
}

public static void listAllEndpoints(PinpointClient pinpoint,
    String applicationId,
    String userId) {

    try {
        GetUserEndpointsRequest endpointsRequest =
        GetUserEndpointsRequest.builder()
            .userId(userId)
            .applicationId(applicationId)
            .build();

        GetUserEndpointsResponse response =
        pinpoint.getUserEndpoints(endpointsRequest);
        List<EndpointResponse> endpoints = response.endpointsResponse().item();

        // Display the results.
        for (EndpointResponse endpoint : endpoints) {
            System.out.println("The channel type is: " + endpoint.channelType());
            System.out.println("The address is " + endpoint.address());
        }

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
}  
}
```

Per l'esempio completo dell'SDK, consulta [ListEndpoints.java](#) su [GitHub](#).

# Creazione di segmenti

Un segmento di utenti rappresenta un sottoinsieme degli utenti basato su caratteristiche condivise, ad esempio l'ultima volta in cui hanno utilizzato l'app o la piattaforma del dispositivo utilizzata. Un segmento indica quali utenti riceveranno i messaggi distribuiti da una campagna. La definizione di segmenti consente di raggiungere i destinatari appropriati quando si vuole invitare gli utenti a utilizzare di nuovo l'app, proporre offerte speciali o aumentare il coinvolgimento e gli acquisti degli utenti.

Dopo aver creato un segmento, è possibile usarlo in una o più campagne. Una campagna invia messaggi mirati agli utenti inclusi nel segmento.

Per ulteriori informazioni, consulta [Segmenti](#).

## Argomenti

- [Creazione di segmenti](#)
- [Importazione di segmenti](#)
- [Personalizzazione dei segmenti con AWS Lambda](#)

## Creazione di segmenti

Per raggiungere i destinatari previsti per una campagna, puoi creare un segmento basato sui dati riportati dalla tua app. Ad esempio, per raggiungere gli utenti che di recente non hanno utilizzato la tua app, puoi definire un segmento per gli utenti che non hanno utilizzato la tua app negli ultimi 30 giorni.

## Creazione di segmenti con l'AWS SDK for Java

L'esempio seguente mostra come creare un segmento con l'AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
```

```
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateSegment {
    public static void main(String[] args) {
        final String usage = ""

                Usage:  <appId>

                Where:
                    appId - The application ID to create a segment for.
```

```
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String appId = args[0];
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    SegmentResponse result = createSegment(pinpoint, appId);
    System.out.println("Segment " + result.name() + " created.");
    System.out.println(result.segmentType());
    pinpoint.close();
}

public static SegmentResponse createSegment(PinpointClient client, String
appId) {
    try {
        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();

        segmentAttributes.put("Team", AttributeDimension.builder()
            .attributeType(AttributeType.INCLUSIVE)
            .values("Lakers")
            .build());

        RecencyDimension recencyDimension = RecencyDimension.builder()
            .duration("DAY_30")
            .recencyType("ACTIVE")
            .build();

        SegmentBehaviors segmentBehaviors = SegmentBehaviors.builder()
            .recency(recencyDimension)
            .build();

        SegmentDemographics segmentDemographics = SegmentDemographics
            .builder()
            .build();

        SegmentLocation segmentLocation = SegmentLocation
            .builder()
```

```
        .build();

        SegmentDimensions dimensions = SegmentDimensions
            .builder()
            .attributes(segmentAttributes)
            .behavior(segmentBehaviors)
            .demographic(segmentDemographics)
            .location(segmentLocation)
            .build();

        WriteSegmentRequest writeSegmentRequest =
WriteSegmentRequest.builder()
            .name("MySegment")
            .dimensions(dimensions)
            .build();

        CreateSegmentRequest createSegmentRequest =
CreateSegmentRequest.builder()
            .applicationId(appId)
            .writeSegmentRequest(writeSegmentRequest)
            .build();

        CreateSegmentResponse createSegmentResult =
client.createSegment(createSegmentRequest);
        System.out.println("Segment ID: " +
createSegmentResult.segmentResponse().id());
        System.out.println("Done");
        return createSegmentResult.segmentResponse();

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return null;
}
}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Per l'esempio completo dell'SDK, consulta [CreateSegment.java](#) su [GitHub](#).

## Importazione di segmenti

Con Amazon Pinpoint, puoi definire un segmento di utenti importando le informazioni sugli endpoint appartenenti al segmento. Un endpoint è una destinazione di messaggistica, ad esempio un token di dispositivo mobile push, un numero di telefono o un indirizzo e-mail.

L'importazione di segmenti è utile se hai già creato segmenti di utenti al di fuori di Amazon Pinpoint ma desideri coinvolgere i tuoi utenti nelle campagne di Amazon Pinpoint.

Quando importi un segmento, Amazon Pinpoint recupera gli endpoint del segmento da Amazon Simple Storage Service (Amazon S3). Prima di eseguire l'importazione, devi aggiungere gli endpoint ad Amazon S3 e creare un ruolo IAM che consente ad Amazon Pinpoint di accedere ad Amazon S3. Devi quindi fornire ad Amazon Pinpoint il percorso in Amazon S3 in cui sono archiviati gli endpoint; Amazon Pinpoint aggiungerà ogni endpoint al segmento.

Per creare il ruolo IAM, consulta [Ruolo IAM per l'importazione di endpoint o segmenti](#). Per informazioni sull'importazione di un segmento utilizzando la console Amazon Pinpoint, consulta [Importazione di segmenti](#) nella Guida per l'utente di Amazon Pinpoint.

## Importazione di un segmento

L'esempio seguente illustra come importare un segmento utilizzando l'AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = ""

            Usage:  <appId> <bucket> <key> <roleArn>\s

            Where:
                appId - The application ID to create a segment for.
                bucket - The name of the Amazon S3 bucket that contains the segment
                key - The key of the S3 object.
                roleArn - ARN of the role that allows Amazon Pinpoint to access S3.
                You need to set trust management for this to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\_policies\_elements\_principal.html

            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket, key,
            roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for import
            job status.");
    }
}
```

```
        System.out.println("See application " + response.jobStatus() + " for import job
status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
        String appId,
        String bucket,
        String key,
        String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                .defineSegment(true)
                .registerEndpoints(true)
                .roleArn(roleArn)
                .format(Format.JSON)
                .s3Url("s3://" + bucket + "/" + key)
                .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                .importJobRequest(importRequest)
                .applicationId(appId)
                .build();

            CreateImportJobResponse jobResponse = client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

Per l'esempio completo dell'SDK, consulta [ImportingSegments.java](#) su [GitHub](#).

## Personalizzazione dei segmenti con AWS Lambda

Questa è la documentazione preliminare per una funzionalità nella versione beta pubblica. ed è soggetta a modifiche.

È possibile utilizzare AWS Lambda per personalizzare il modo in cui una campagna Amazon Pinpoint coinvolge i destinatari. Con AWS Lambda, è possibile modificare il segmento della campagna nel momento in cui Amazon Pinpoint invia il messaggio.

AWS Lambda è un servizio di calcolo che consente di eseguire il codice senza gestire i server o effettuare il provisioning. È possibile creare pacchetti di codice e quindi caricarli in Lambda come funzioni Lambda. Lambda esegue una funzione quando la funzione viene richiesta, manualmente o in modo automatico, in risposta a eventi. Per ulteriori informazioni, consulta la [Guida per gli sviluppatori di AWS Lambda](#).

Per assegnare una funzione Lambda a una campagna, definisci le impostazioni CampaignHook della campagna utilizzando la risorsa [Campagna](#) nell'API Amazon Pinpoint. Tali impostazioni includono il nome della funzione Lambda e la modalità CampaignHook, che specifica se Amazon Pinpoint riceve un valore restituito dalla funzione.

Una funzione Lambda assegnata a una campagna viene definita estensione di Amazon Pinpoint.

Dopo aver definito le impostazioni CampaignHook, Amazon Pinpoint richiama automaticamente la funzione Lambda quando esegue la campagna, prima di inviare il messaggio della campagna. Quando Amazon Pinpoint richiama la funzione, fornisce dati di evento che si riferiscono al recapito del messaggio. Questi dati includono il segmento della campagna, ovvero l'elenco degli endpoint a cui Amazon Pinpoint invia il messaggio.

Se la modalità CampaignHook è impostata su FILTER, Amazon Pinpoint consente alla funzione di modificare e restituire il segmento prima di inviare il messaggio. Ad esempio, è possibile che la funzione aggiorni le definizioni di endpoint con attributi contenenti i dati provenienti da un'origine esterna ad Amazon Pinpoint. Oppure potrebbe filtrare il segmento rimuovendo determinati endpoint, in base alle condizioni indicate nel tuo codice di funzione. Dopo che ha ricevuto il segmento modificato dalla funzione, Amazon Pinpoint invia il messaggio a ciascun endpoint del segmento utilizzando il canale di distribuzione della campagna.

Elaborando i segmenti con AWS Lambda, si dispone di maggiore controllo sugli utenti a cui inviare messaggi e sul contenuto dei messaggi stessi. È possibile personalizzare le campagne in tempo reale, nel momento in cui vengono inviati i messaggi. L'applicazione di filtri ai segmenti consente di coinvolgere sottoinsiemi di segmenti definiti in modo più specifico. L'aggiunta o l'aggiornamento di attributi di endpoint permette anche di rendere disponibili nuovi dati per le variabili dei messaggi.

### Note

È inoltre possibile utilizzare le impostazioni CampaignHook per assegnare una funzione Lambda che gestisca il recapito dei messaggi. Questo tipo di funzione è utile per distribuire messaggi tramite canali personalizzati non supportati da Amazon Pinpoint, ad esempio piattaforme di social media. Per ulteriori informazioni, consulta [Creazione di canali personalizzati in Amazon Pinpoint](#).

Quando si richiama un hook Lambda utilizzando Amazon Pinpoint, anche la funzione Lambda deve trovarsi nella stessa regione del progetto Amazon Pinpoint.

Per modificare i segmenti di una campagna con AWS Lambda, è prima necessario creare una funzione che elabori i dati relativi agli eventi inviati da Amazon Pinpoint e restituisca un segmento modificato. Quindi, autorizza Amazon Pinpoint a richiamare la funzione assegnando una policy di funzione Lambda. Infine, bisogna assegnare la funzione a una o più campagne definendo le impostazioni CampaignHook.

## Dati eventi

Quando richiama la funzione Lambda, Amazon Pinpoint fornisce il payload seguente come dati degli eventi:

```
{
  "MessageConfiguration": {Message configuration}
  "ApplicationId": ApplicationId,
  "CampaignId": CampaignId,
  "TreatmentId": TreatmentId,
  "ActivityId": ActivityId,
  "ScheduledTime": Scheduled Time,
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

AWS Lambda trasferisce i dati di evento al codice di funzione. I dati sugli eventi forniscono gli attributi seguenti:

- MessageConfiguration: ha la stessa struttura dell'oggetto DirectMessageConfiguration della risorsa [Messages](#) nell'API Amazon Pinpoint.

- **ApplicationId**: ID del progetto Amazon Pinpoint a cui appartiene la campagna.
- **CampaignId**: ID della campagna Amazon Pinpoint per cui viene richiamata la funzione.
- **TreatmentId**: ID di una variante di campagna utilizzata per il testing A/B.
- **ActivityId**: ID dell'attività eseguita dalla campagna.
- **ScheduledTime**: data e ora, in formato ISO 8601, di recapito dei messaggi della campagna.
- **Endpoints**: mappa che associa gli ID degli endpoint alle definizioni di endpoint. Ogni payload di dati sugli eventi contiene fino a 50 endpoint. Se il segmento di campagna contiene più di 50 endpoint, Amazon Pinpoint richiama più volte la funzione, con un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti gli endpoint.

## Creazione di una funzione Lambda

Per informazioni su come creare una funzione Lambda, consulta [Guida introduttiva](#) nella Guida per gli sviluppatori di AWS Lambda. Quando crei una funzione, tieni presente che la consegna del messaggio non riesce quando si verificano le seguenti condizioni:

- La funzione Lambda impiega più di 15 secondi per restituire il segmento modificato.
- Amazon Pinpoint non è in grado di decodificare il valore restituito della funzione.
- Per richiamare la funzione, Amazon Pinpoint deve fare più di 3 tentativi.

Amazon Pinpoint accetta solo definizioni di endpoint nel valore restituito della funzione. La funzione non può modificare altri elementi nei dati di evento.

### Funzione Lambda di esempio

La funzione Lambda elabora i dati degli eventi inviati da Amazon Pinpoint e restituisce l'endpoint modificato, come illustrato dal seguente gestore di esempio, scritto in Node.js:

```
'use strict';

exports.handler = (event, context, callback) => {
  for (var key in event.Endpoints) {
    if (event.Endpoints.hasOwnProperty(key)) {
      var endpoint = event.Endpoints[key];
      var attr = endpoint.Attributes;
      if (!attr) {
        attr = {};
      }
    }
  }
}
```

```
        endpoint.Attributes = attr;
    }
    attr["CreditScore"] = [ Math.floor(Math.random() * 200) + 650];
}
}
console.log("Received event:", JSON.stringify(event, null, 2));
callback(null, event.Endpoints);
};
```

Lambda trasferisce i dati degli eventi al gestore come parametro `event`.

In questo esempio il gestore scorre tutti gli endpoint nell'oggetto `event.Endpoints` e aggiunge un nuovo attributo, `CreditScore`, all'endpoint. Il valore dell'attributo `CreditScore` è semplicemente un numero casuale.

L'istruzione `console.log()` registra l'evento nei file di log CloudWatch.

L'istruzione `callback()` restituisce gli endpoint modificati ad Amazon Pinpoint. Di solito, il parametro `callback` è facoltativo nelle funzioni Lambda Node.js, ma in questo contesto è obbligatorio perché la funzione deve restituire ad Amazon Pinpoint gli endpoint aggiornati.

La funzione deve restituire gli endpoint nello stesso formato dei dati di evento, che fornisce una mappa di associazioni di ID e definizioni degli endpoint, come nell'esempio seguente:

```
{
  "eqmj8wpxszeqy/b3vch04sn41yw": {
    "ChannelType": "GCM",
    "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "Demographic": {
      "Make": "android"
    },
    "EffectiveDate": "2017-11-02T21:26:48.598Z",
    "User": {}
  },
  "idrexqqtn8sbwfex0ouscod0yto": {
    "ChannelType": "APNS",
    "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "Demographic": {
      "Make": "apple"
    }
  }
}
```

```

    },
    "EffectiveDate": "2017-11-02T21:26:48.598Z",
    "User": {}
  }
}

```

La funzione di esempio modifica e restituisce l'oggetto `event`. `Endpoints` ricevuto nei dati di evento.

Facoltativamente, è possibile includere gli attributi `TitleOverride` e `BodyOverride` nelle definizioni degli endpoint che vengono restituite.

### Note

Quando si utilizza questa soluzione per l'invio di messaggi, Amazon Pinpoint implementa gli attributi `TitleOverride` e `BodyOverride` solo per gli endpoint in cui l'attributo `ChannelType` ha uno dei seguenti valori: `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` o `SMS`.

Amazon Pinpoint non rispetta questi attributi per gli endpoint in cui il valore dell'attributo `ChannelType` è `EMAIL`.

## Assegnazione di una policy di funzione Lambda

Prima di utilizzare la funzione Lambda per elaborare gli endpoint, è necessario autorizzare Amazon Pinpoint a richiamare la funzione Lambda. Per concedere l'autorizzazione per l'invocazione, assegna una policy di funzione Lambda alla funzione. Una policy di funzione Lambda è una policy di autorizzazione basata su risorse che designa le entità che possono utilizzare la funzione e le azioni che tali entità possono eseguire.

Per ulteriori informazioni, consulta [Uso delle policy basate sulle risorse per AWS Lambda](#) nella Guida per gli sviluppatori di AWS Lambda.

### Policy di funzione di esempio

La policy seguente autorizza l'entità principale del servizio Amazon Pinpoint a utilizzare l'azione `lambda:InvokeFunction` per una campagna specifica (*campaign-id*):

```
{
```

```

"Sid": "sid",
"Effect": "Allow",
"Principal": {
  "Service": "pinpoint.us-east-1.amazonaws.com"
},
"Action": "lambda:InvokeFunction",
"Resource": "{arn:aws:lambda:us-east-1:account-id:function:function-name}",
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/campaigns/campaign-id"
  }
}
}

```

La policy di funzione richiede un blocco `Condition` che includa una chiave `AWS:SourceArn`. Questo codice indica la campagna Amazon Pinpoint a cui è consentito richiamare la funzione. In questo esempio, la policy concede l'autorizzazione per una sola campagna. Il blocco `Condition` deve includere anche una chiave `AWS:SourceAccount`, che controlla quale account AWS può richiamare l'azione.

Per scrivere una policy più generica, utilizza un carattere jolly di corrispondenza con più caratteri (\*). Ad esempio, puoi usare il seguente blocco `Condition` per consentire a qualsiasi campagna in un progetto Amazon Pinpoint specifico (*application-id*) di richiamare la funzione:

```

...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/campaigns/*"
  }
}
...

```

Se vuoi che la funzione Lambda sia la funzione predefinita utilizzata da tutte le campagne per un progetto, è consigliabile configurare il blocco `Condition` per la policy nel modo precedente. Per

informazioni sull'impostazione di una funzione Lambda come predefinita per tutte le campagne in un progetto, consulta [Assegnazione di una funzione Lambda a una campagna](#).

## Concessione dell'autorizzazione di invocazione di Amazon Pinpoint

È possibile usare l'AWS Command Line Interface (AWS CLI) per aggiungere autorizzazioni alla policy di funzione Lambda assegnata alla funzione Lambda stessa. Per consentire ad Amazon Pinpoint di richiamare una funzione per una determinata campagna, usa il comando Lambda [add-permission](#), come mostrato nell'esempio seguente:

```
$ aws lambda add-permission \  
> --function-name function-name \  
> --statement-id sid \  
> --action lambda:InvokeFunction \  
> --principal pinpoint.us-east-1.amazonaws.com \  
> --source-account 111122223333 \  
> --source-arn arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/  
campaigns/campaign-id
```

Puoi cercare gli ID delle campagne utilizzando il comando [get-campaigns](#) in AWS CLI. Puoi anche cercare l'ID dell'applicazione utilizzando il comando [get-apps](#).

Quando esegui il comando Lambda `add-permission`, viene restituito l'output seguente:

```
{  
  "Statement": "{\"Sid\":\"sid\",  
    \"Effect\":\"Allow\",  
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},  
    \"Action\":\"lambda:InvokeFunction\",  
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",  
    \"Condition\":  
      {\"ArnLike\":  
        {\"AWS:SourceArn\":  
          \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/application-id/  
campaigns/campaign-id\"}}  
        {\"StringEquals\":  
          {\"AWS:SourceAccount\":  
            \"111122223333\"}}}}  
}
```

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy della funzione Lambda.

## Assegnazione di una funzione Lambda a una campagna

È possibile assegnare una funzione Lambda a una campagna Amazon Pinpoint specifica. In alternativa, puoi impostare la funzione Lambda come predefinita per tutte le campagne in un progetto, tranne le campagne a cui assegni una funzione individualmente.

Per assegnare una funzione Lambda a una campagna specifica, utilizza l'API Amazon Pinpoint per creare o aggiornare un oggetto [Campaign](#) e definire il relativo attributo CampaignHook. Per impostare una funzione Lambda come predefinita per tutte le campagne in un progetto, crea o aggiorna la risorsa [Settings](#) per il progetto in questione e definisci il relativo oggetto CampaignHook.

In entrambi i casi, imposta gli attributi CampaignHook seguenti:

- **LambdaFunctionName**: nome o nome della risorsa Amazon (ARN) della funzione Lambda richiamata da Amazon Pinpoint prima di inviare i messaggi per la campagna.
- **Mode**: impostato su FILTER. Con questa modalità, Amazon Pinpoint richiama la funzione e attende che restituisca gli endpoint modificati. Dopo averli ricevuti, Amazon Pinpoint invia il messaggio. Amazon Pinpoint attende un massimo di 15 secondi prima che il recapito dei messaggi venga impostato come non riuscito.

Con le impostazioni CampaignHook definite per una campagna, Amazon Pinpoint richiama la funzione Lambda specificata prima di inviare i messaggi della campagna. Amazon Pinpoint attende di ricevere gli endpoint modificati dalla funzione. Se riceve gli endpoint aggiornati, Amazon Pinpoint procede con il recapito dei messaggi utilizzando i dati aggiornati.

# Creazione di campagne

Per aumentare il coinvolgimento tra le app e i relativi utenti, con Amazon Pinpoint puoi creare e gestire campagne tramite notifiche push per raggiungere specifici segmenti di utenti.

Ad esempio, una campagna può promuovere l'uso dell'app presso gli utenti che non l'hanno eseguita di recente oppure offrire promozioni speciali agli utenti che non effettuano acquisti da tempo.

Una campagna invia un messaggio mirato a un segmento di utenti specificato. La campagna può inviare il messaggio a tutti gli utenti nel segmento. In alternativa, puoi allocare un'esclusione, ovvero una percentuale di utenti che non ricevono i messaggi.

È possibile pianificare la campagna in modo che il messaggio venga inviato una sola volta o con una determinata frequenza, ad esempio una volta a settimana. Per evitare che gli utenti ricevano il messaggio in orari inopportuni, la pianificazione può includere un orario di silenzio in cui non vengono inviati messaggi.

Per sperimentare diverse strategie per le campagne, configura una campagna come test A/B. Un test A/B include due o più varianti per il messaggio o la pianificazione. Le varianti sono versioni diverse del messaggio o della pianificazione. Man mano che gli utenti rispondono alla campagna, puoi osservare i dati analitici della campagna per confrontare l'efficacia di ogni variante.

Per ulteriori informazioni, consulta l'argomento relativo alle [campagne](#).

## Creazione di campagne standard

Una campagna standard invia una notifica push personalizzata a un determinato segmento in base a una pianificazione definita in precedenza.

## Creazione di campagne con AWS SDK for Java

L'esempio seguente mostra come creare una campagna con l' AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
```

```
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = ""

            Usage:  <appId> <segmentId>

            Where:
                appId - The ID of the application to create the campaign in.
                segmentId - The ID of the segment to create the campaign from.
            "";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
    }
}
```

```
String appId = args[0];
String segmentId = args[1];
PinpointClient pinpoint = PinpointClient.builder()
    .region(Region.US_EAST_1)
    .build();

createPinCampaign(pinpoint, appId, segmentId);
pinpoint.close();
}

public static void createPinCampaign(PinpointClient pinpoint, String appId, String
segmentId) {
    CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
    System.out.println("Campaign " + result.name() + " created.");
    System.out.println(result.description());
}

public static CampaignResponse createCampaign(PinpointClient client, String appId,
String segmentID) {

    try {
        Schedule schedule = Schedule.builder()
            .startTime("IMMEDIATE")
            .build();

        Message defaultMessage = Message.builder()
            .action(Action.OPEN_APP)
            .body("My message body.")
            .title("My message title.")
            .build();

        MessageConfiguration messageConfiguration = MessageConfiguration.builder()
            .defaultMessage(defaultMessage)
            .build();

        WriteCampaignRequest request = WriteCampaignRequest.builder()
            .description("My description")
            .schedule(schedule)
            .name("MyCampaign")
            .segmentId(segmentID)
            .messageConfiguration(messageConfiguration)
            .build();
```

```
        CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
        .applicationId(appID)
        .writeCampaignRequest(request).build());

        System.out.println("Campaign ID: " + result.campaignResponse().id());
        return result.campaignResponse();

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return null;
}
}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Per l'esempio completo dell'SDK, consulta [CreateCampaign.java on GitHub](#)

## Creazione di campagne di test A/B

Un test A/B si comporta come una campagna standard, ma consente di definire diverse varianti per il messaggio o la pianificazione della campagna.

### Creazione di campagne di test A/B con AWS SDK for Java

L'esempio seguente mostra come creare una campagna di test A/B con l' AWS SDK for Java.

```
import com.amazonaws.services.pinpoint.AmazonPinpointClient;
import com.amazonaws.services.pinpoint.model.Action;
import com.amazonaws.services.pinpoint.model.CampaignResponse;
import com.amazonaws.services.pinpoint.model.CreateCampaignRequest;
import com.amazonaws.services.pinpoint.model.CreateCampaignResult;
import com.amazonaws.services.pinpoint.model.Message;
import com.amazonaws.services.pinpoint.model.MessageConfiguration;
import com.amazonaws.services.pinpoint.model.Schedule;
import com.amazonaws.services.pinpoint.model.WriteCampaignRequest;
import com.amazonaws.services.pinpoint.model.WriteTreatmentResource;
```

```
import java.util.ArrayList;
import java.util.List;

public class PinpointCampaignSample {

    public CampaignResponse createAbCampaign(AmazonPinpointClient client, String appId,
String segmentId) {
        Schedule schedule = new Schedule()
            .withStartTime("IMMEDIATE");

        // Default treatment.
        Message defaultMessage = new Message()
            .withAction(Action.OPEN_APP)
            .withBody("My message body.")
            .withTitle("My message title.");

        MessageConfiguration messageConfiguration = new MessageConfiguration()
            .withDefaultMessage(defaultMessage);

        // Additional treatments
        WriteTreatmentResource treatmentResource = new WriteTreatmentResource()
            .withMessageConfiguration(messageConfiguration)
            .withSchedule(schedule)
            .withSizePercent(40)
            .withTreatmentDescription("My treatment description.")
            .withTreatmentName("MyTreatment");

        List<WriteTreatmentResource> additionalTreatments = new
ArrayList<WriteTreatmentResource>();
        additionalTreatments.add(treatmentResource);

        WriteCampaignRequest request = new WriteCampaignRequest()
            .withDescription("My description.")
            .withSchedule(schedule)
            .withSegmentId(segmentId)
            .withName("MyCampaign")
            .withMessageConfiguration(messageConfiguration)
            .withAdditionalTreatments(additionalTreatments)
            .withHoldoutPercent(10); // Hold out of A/B test

        CreateCampaignRequest createCampaignRequest = new CreateCampaignRequest()
            .withApplicationId(appId).withWriteCampaignRequest(request);
    }
}
```

```
    CreateCampaignResult result = client.createCampaign(createCampaignRequest);  
  
    System.out.println("Campaign ID: " + result.getCampaignResponse().getId());  
  
    return result.getCampaignResponse();  
  }  
}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

# Utilizzo dell'API SMS e Voce versione 2 di Amazon Pinpoint

## Note

Amazon Pinpoint ha aggiornato la documentazione relativa alla guida per l'utente. Per ottenere le informazioni più recenti su come creare, configurare e gestire le risorse vocali e SMS di Amazon Pinpoint, consulta la nuova [Guida per l'utente per Amazon Pinpoint SMS](#). Il seguente argomento è stato spostato nella nuova [Amazon Pinpoint SMS User Guide](#).

- [Gestione dei numeri di telefono](#)
- [Gestione degli ID mittente](#)
- [Gestione dei pool](#)
- [Gestione degli elenchi di opt-out](#)
- [Gestione dei set di configurazione](#)
- [Gestione delle parole chiave](#)
- [Gestione delle destinazioni degli eventi](#)
- [Invio di messaggi](#)

Amazon Pinpoint include un'API (denominata API SMS e Voce versione 2) progettata per l'invio di SMS e messaggi vocali. Mentre l'API Amazon Pinpoint si concentra sull'invio di messaggi tramite campagne e percorsi pianificati e basati su eventi, l'API SMS e Voce offre nuove funzioni e funzionalità per l'invio diretto di SMS e messaggi vocali a singoli destinatari. Puoi utilizzare l'API SMS e Voce indipendentemente dalle funzionalità relative a campagne e percorsi di Amazon Pinpoint, oppure puoi usarle entrambe contemporaneamente per soddisfare diversi casi d'uso. Se utilizzi già Amazon Pinpoint per inviare SMS o messaggi vocali, il tuo account è già configurato per utilizzare questa API.

Questa API è una buona soluzione per gli utenti che dispongono di un'architettura multi-tenant, come i fornitori di software indipendenti (ISV). Questa API semplifica la separazione dei dati relativi a eventi, numeri di telefono di origine ed elenchi di opt-out per i diversi tenant.

Quando utilizzi l'API SMS e Voce, ti consigliamo di configurare set di configurazione e destinazioni di evento. L'API SMS e Voce non genera automaticamente i dati sugli eventi per i messaggi inviati. La configurazione delle destinazioni di evento garantisce l'acquisizione di dati importanti sugli eventi, come il recapito dei messaggi e gli eventi di errore.

La versione 2 di questa API è stata preceduta dalla versione 1. Se attualmente utilizzi la versione 1 di questa API, questa versione continuerà a essere disponibile e potrai continuare a usarla. Tuttavia, se esegui la migrazione alla versione 2, potrai disporre di funzionalità aggiuntive, come la possibilità di creare pool di numeri di telefono, richiedere nuovi numeri di telefono a livello di programmazione e abilitare o disabilitare determinate funzionalità dei numeri di telefono.

#### Note

Attualmente ci sono alcune attività che possono essere completate solo utilizzando la console Amazon Pinpoint. Ad esempio, se desideri [verificare un numero di telefono da utilizzare quando il tuo account si trova nell'ambiente di sperimentazione \(sandbox\) per SMS](#) o se desideri [registrarti per utilizzare codici lunghi 10DLC](#), devi utilizzare la console Amazon Pinpoint.

In questa sezione sono descritte le informazioni su questa API e sono contenuti esempi di come utilizzarla. Puoi anche trovare ulteriore documentazione di riferimento nella [documentazione di riferimento dell'API SMS e Voce versione 2](#).

# Invio e convalida di password monouso (OTP) con Amazon Pinpoint

Amazon Pinpoint include una funzionalità di gestione dei codici OTP (One-Time Password). Puoi utilizzare questa funzionalità per generare nuovi codici e inviarli ai destinatari come messaggi SMS. Le applicazioni possono quindi chiamare l'API Amazon Pinpoint per verificare questi codici.

## Important

Per utilizzare questa funzionalità, il tuo account deve disporre dell'accesso alla produzione e di un'identità di origine attiva. Per ulteriori informazioni, consulta [Informazioni sulla sandbox Amazon Pinpoint SMS](#) e [Richiedi un numero di telefono](#) nella Guida per l'utente di Amazon Pinpoint SMS.

In alcuni paesi e regioni, è necessario ottenere un numero di telefono o un ID di origine dedicato prima di poter inviare messaggi SMS. Ad esempio, quando invii messaggi a destinatari negli Stati Uniti, devi disporre di un numero verde dedicato, un numero 10DLC o un codice breve. Quando invii messaggi a destinatari in India, devi disporre di un ID mittente registrato, che include un ID entità principale (PEID) e un ID modello. Questi requisiti continuano a essere valido quando si utilizza la funzionalità OTP.

Per utilizzare questa funzionalità, sono necessarie le autorizzazioni per inviare e verificare i messaggi OTP; consulta [Codici OTP \(One-Time Password\)](#). Per assistenza durante la definizione delle autorizzazioni, consulta [Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint](#).

## Invio di un messaggio OTP

Puoi utilizzare l'operazione `SendOtpMessages` nell'API Amazon Pinpoint per inviare un codice OTP (One-Time Password) a un utente della tua applicazione. Quando usi questa API, Amazon Pinpoint genera un codice casuale e lo invia all'utente come messaggio SMS. La tua richiesta può includere i seguenti parametri:

- `Channel`: canale di comunicazione attraverso il quale viene inviato il codice OTP. Attualmente sono supportati solo i messaggi SMS, quindi l'unico valore accettabile è SMS.
- `BrandName`: nome del marchio, dell'azienda o del prodotto associato al codice OTP. Il nome può contenere un massimo di 20 caratteri.

**Note**

Quando Amazon Pinpoint invia il messaggio OTP, il nome del marchio viene inserito automaticamente nel seguente modello di messaggio:

```
This is your One Time Password: {{otp}} from {{brand}}
```

Quindi, se specifichi ExampleCorp come nome del marchio e Amazon Pinpoint genera una password monouso di 123456, invia il seguente messaggio all'utente:

```
This is your One Time Password: 123456 from ExampleCorp
```

- **CodeLength**: numero di cifre contenute nel codice OTP inviato al destinatario. I codici OTP possono contenere da 5 a 8 cifre.
- **ValidityPeriod**: periodo di validità, espresso in minuti, del codice OTP. Il periodo di validità può essere compreso tra 5 e 60 minuti.
- **AllowedAttempts**: numero di volte in cui il destinatario può tentare senza successo di verificare il codice OTP. Se il numero di tentativi supera questo valore, il codice OTP diventa automaticamente non valido. Il numero massimo di tentativi consentiti è 5.
- **Language**: lingua, in formato IETF BCP-47, da utilizzare per l'invio del messaggio. I valori accettabili sono:
  - de-DE: tedesco
  - en-GB: inglese (Regno Unito)
  - en-US: inglese (Stati Uniti)
  - es-419: spagnolo (America Latina)
  - es-ES: spagnolo
  - fr-CA: francese (Canada)
  - fr-FR: francese
  - it-IT: italiano
  - ja-JP: giapponese
  - ko-KR: coreano
  - pt-BR: portoghese (Brasile)
  - zh-CN: cinese (semplificato)

- `zh-TW`: cinese (tradizionale)
- `OriginationIdentity`: identità di origine (ad esempio codice lungo, codice breve o ID mittente) utilizzata per inviare il codice OTP. Se utilizzi un codice lungo o un numero verde per inviare il codice OTP (One-Time Password), il numero di telefono deve essere in formato E.164.
- `DestinationIdentity`: numero di telefono, nel formato E.164, al quale è stato inviato il codice OTP.
- `ReferenceId`: ID di riferimento univoco per la richiesta. L'ID di riferimento corrisponde esattamente all'ID di riferimento fornito al momento della verifica del codice OTP. L'ID di riferimento può contenere da 1 a 48 caratteri.
- `EntityId`: ID dell'entità registrato presso un ente regolatore. Questo parametro viene attualmente utilizzato solo per l'invio di messaggi ai destinatari in India. Se non effettui invii a destinatari in India, puoi omettere questo parametro.
- `TemplateId`: ID del modello registrato presso un ente regolatore. Questo parametro viene attualmente utilizzato solo per l'invio di messaggi ai destinatari in India. Se non effettui invii a destinatari in India, puoi omettere questo parametro.

#### Note

Per ulteriori informazioni sui requisiti per l'invio di messaggi a destinatari in India, consulta [Requisiti speciali per l'invio di messaggi SMS a destinatari in India](#) nella Guida per l'utente di Amazon Pinpoint.

Per assicurarti che il tuo account Amazon Pinpoint sia configurato correttamente per inviare messaggi OTP, puoi utilizzare l'opzione AWS CLI per inviare un messaggio di prova. Per ulteriori informazioni su AWS CLI, consulta la Guida per l'[AWS Command Line Interface utente](#).

Per inviare un messaggio OTP di prova utilizzando il AWS CLI, esegui il [send-otp-message](#) comando nel terminale:

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-otp-message-request-parameters Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

Nel comando precedente, procedi come segue.

- Sostituisci *7353f53e6885409fa32d07cedexample* con l'ID dell'applicazione.

- *ExampleCorp* Sostituiscilo con il nome della tua azienda.
- Sostituisci *5* pollici CodeLength con il numero di cifre che saranno presenti nel codice OTP che verrà inviato al destinatario.
- Sostituisci *20* in ValidityPeriod con il periodo di validità, in minuti, del codice OTP.
- Sostituisci *5* in AllowedAttempts con il numero di volte in cui il destinatario può tentare senza successo di verificare l'OTP.
- Sostituisci *+1855550142* OriginationIdentity con l'identità di origine utilizzata per inviare il codice OTP.
- Sostituisci *+12065550007 DestinationIdentity con il numero di telefono a cui inviare* il codice OTP.
- *SampleReferenceId* Sostituiscilo ReferenceId con un ID di riferimento univoco per la richiesta.

## Risposta **SendOtpMessage**

Quando invii correttamente un messaggio OTP, viene visualizzata una risposta analoga all'esempio seguente:

```
{
  "MessageResponse": {
    "ApplicationId": "7353f53e6885409fa32d07cedexample",
    "RequestId": "255d15ea-75fe-4040-b919-096f2example",
    "Result": {
      "+12065550007": {
        "DeliveryStatus": "SUCCESSFUL",
        "MessageId": "nvrimgq9kq4en96qgp0tlqli3og1at6aexample",
        "StatusCode": 200,
        "StatusMessage": "MessageId: nvrimgq9kq4en96qgp0tlqli3og1at6aexample"
      }
    }
  }
}
```

## Convalida di un messaggio OTP

Per verificare un codice OTP, chiama l'API `VerifyOtpMessages`. La richiesta deve includere i seguenti parametri:

- **DestinationIdentity**: numero di telefono, nel formato E.164, al quale è stato inviato il codice OTP.
- **ReferenceId**: ID di riferimento usato quando hai inviato il codice OTP al destinatario. L'ID di riferimento deve corrispondere esattamente.
- **Otp**: codice OTP che stai convalidando.

È possibile utilizzare il AWS CLI per testare il processo di convalida. Per ulteriori informazioni sull'installazione e la configurazione di AWS CLI, consulta la Guida per l'[AWS Command Line Interface utente](#).

Per verificare una OTP utilizzando il AWS CLI, esegui il [verify-otp-message](#) comando nel terminale:

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
  DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

Nel comando precedente, procedi come segue.

- Sostituisci *7353f53e6885409fa32d07cedexample* con l'ID dell'applicazione.
- *Sostituisci +12065550007 con il numero di telefono a cui è stato inviato il codice OTP.* DestinationIdentity
- *SampleReferenceId* Sostituiscilo ReferenceId con un ID di riferimento univoco per la richiesta. Questo valore deve corrispondere a ReferenceID quello utilizzato per inviare la richiesta.
- Sostituisci *01234* Otp con un Otp inviato a DestinationIdentity

## Risposta VerifyOtpMessage

Quando invii una richiesta all'API VerifyOTPMMessage, viene restituito un oggetto VerificationResponse contenente un'unica proprietà (Valid). Se l'ID di riferimento, il numero di telefono e il codice OTP corrispondono tutti ai valori previsti da Amazon Pinpoint e se il codice OTP non è scaduto, il valore Valid è true. In caso contrario, è false. Di seguito è riportato un esempio di risposta per una verifica riuscita del codice OTP.

```
{
  "VerificationResponse": {
    "Valid": true
  }
}
```

```
}
```

## Esempi di codice

Questa sezione contiene esempi di codice che mostrano come utilizzare SDK per Python (Boto3) per inviare e verificare codici OTP (One-Time Password).

### Generazione di un ID di riferimento

La seguente funzione genera un ID di riferimento univoco per ogni destinatario, in base al numero di telefono del destinatario, al prodotto o al marchio per cui il destinatario riceve un codice OTP (One-Time Password) e all'origine della richiesta (che, ad esempio, può essere il nome di una pagina di un sito o di un'app). Quando verifichi il codice OTP, devi fornire un ID di riferimento identico affinché la convalida abbia esito positivo. Sia l'esempio di codice di invio che quello di convalida utilizzano questa funzione di utilità.

Questa funzione non è obbligatoria, ma è un valido strumento per adattare il processo di invio e verifica del codice OTP a una transazione specifica in un modo da semplificarne il nuovo invio durante la fase di verifica. Puoi utilizzare qualsiasi ID di riferimento: questo infatti è solo un esempio di base. Tuttavia, gli altri esempi di codice in questa sezione si basano su questa funzione.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

import hashlib

def generate_ref_id(destinationNumber, brandName, source):
    refId = brandName + source + destinationNumber
    return hashlib.md5(refId.encode()).hexdigest()
```

### Invio di codici OTP

L'esempio di codice seguente mostra come utilizzare SDK per Python (Boto3) per inviare un codice OTP (One-Time Password).

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

import boto3
from botocore.exceptions import ClientError
```

```
from generate_ref_id import generate_ref_id

### Some variables that are unlikely to change from request to request. ###

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The phone number or short code to send the message from.
originationNumber = "+18555550142"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# The number of times the user can unsuccessfully enter the OTP code before it becomes
  invalid.
allowedAttempts = 3

# Function that sends the OTP as an SMS message.
def send_otp(destinationNumber, codeLength, validityPeriod, brandName, source, language):
    client = boto3.client('pinpoint', region_name=region)
    try:
        response = client.send_otp_message(
            ApplicationId=appId,
            SendOTPMessageRequestParameters={
                'Channel': 'SMS',
                'BrandName': brandName,
                'CodeLength': codeLength,
                'ValidityPeriod': validityPeriod,
                'AllowedAttempts': allowedAttempts,
                'Language': language,
                'OriginationIdentity': originationNumber,
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber, brandName, source)
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Send a message to +14255550142 that contains a 6-digit OTP that is valid for 15
  minutes. The
```

```
# message will include the brand name "ExampleCorp", and the request originated from a
# part of your
# site or application called "CreateAccount". The US English message template should be
# used to
# send the message.
send_otp("+14255550142",6,15,"ExampleCorp","CreateAccount","en-US")
```

## Convalida dei codici OTP

L'esempio di codice seguente mostra come utilizzare SDK per Python (Boto3) per verificare un codice OTP (One-Time Password) già inviato. Affinché la fase di convalida abbia esito positivo, la richiesta deve includere un ID di riferimento che corrisponda esattamente all'ID di riferimento utilizzato per inviare il messaggio.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# Function that verifies the OTP code.
def verify_otp(destinationNumber,otp,brandName,source):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.verify_otp_message(
            ApplicationId=appId,
            VerifyOTPMessageRequestParameters={
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source),
                'Otp': otp
            }
        )

    except ClientError as e:
        print(e.response)
    else:
```

```
print(response)

# Verify the OTP 012345, which was sent to +14255550142. The brand name ("ExampleCorp")
# and the
# source name ("CreateAccount") are used to generate the correct reference ID.
verify_otp("+14255550142","012345","ExampleCorp","CreateAccount")
```

# Invio e recupero di messaggi in-app in Amazon Pinpoint

Puoi utilizzare i messaggi in-app per inviare messaggi mirati agli utenti delle tue applicazioni. I messaggi in-app sono altamente personalizzabili. Possono includere pulsanti che aprono siti Web o indirizzano gli utenti a parti specifiche dell'app. Puoi configurare i colori dello sfondo e del testo, posizionare il testo e aggiungere pulsanti e immagini alla notifica. Puoi inviare un unico messaggio o creare un carosello contenente un massimo di cinque messaggi univoci. Per una panoramica dei messaggi in-app, comprese le istruzioni per la creazione di modelli di messaggi in-app, consulta [Creazione di modelli in-app](#) nella Guida per l'utente di Amazon Pinpoint.

Puoi utilizzarlo AWS Amplify per integrare senza problemi le funzionalità di messaggistica in-app di Amazon Pinpoint nelle app. Amplify può gestire automaticamente i processi di recupero e rendering dei messaggi, nonché il processo di invio di dati di analisi ad Amazon Pinpoint. Questa integrazione è attualmente supportata per le applicazioni React Native. Per ulteriori informazioni, consulta l'argomento relativo alla [messaggistica in-app](#) nella documentazione del Framework Amplify.

Questa sezione fornisce informazioni sulla richiesta di messaggi in-app per un endpoint nell'app e sull'interpretazione del risultato di tale richiesta.

## Recupero dei messaggi in-app per un endpoint

Le applicazioni possono chiamare l'API [GetInAppMessages](#) per recuperare tutti i messaggi in-app associati a un determinato endpoint. Quando esegui la chiamata all'API `GetInAppMessages`, è necessario fornire i parametri riportati di seguito:

- `ApplicationId`: ID univoco del progetto Amazon Pinpoint a cui è associata la campagna di messaggi in-app.
- `EndpointId`: ID univoco dell'endpoint per cui stai recuperando i messaggi.

Quando viene chiamata con questi valori, l'API restituisce un elenco di messaggi. Per ulteriori informazioni sulle risposte generate da questa operazione, consulta [Informazioni sulle risposte delle API `GetInAppMessages`](#).

Puoi chiamare l'operazione AWS anche tramite `GetInAppMessages` SDK. Gli esempi di codice seguenti includono funzioni che recuperano i messaggi in-app.

## JavaScript

Crea il client in un modulo separato e lo esporta.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

Recupera i messaggi in-app di un endpoint:

```
// Import required AWS SDK clients and commands for Node.js
import { PinpointClient, GetInAppMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "../lib/pinClient.js";

("use strict");

//The Amazon Pinpoint application ID.
const projectId = "4c545b28d21a490cb51b0b364example";

//The ID of the endpoint to retrieve messages for.
const endpointId = "c5ac671ef67ee3ad164cf7706example";

const params = {
  ApplicationId: projectId,
  EndpointId: endpointId
};

const run = async () => {
  try {
    const data = await pinClient.send(new GetInAppMessagesCommand(params));
    console.log(JSON.stringify(data, null, 4));
    return data;
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

## Python

```
import logging
```

```
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def retrieve_inapp_messages(
    pinpoint_client, project_id, endpoint_id):
    """
    Retrieves the in-app messages that a given endpoint is entitled to.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project ID.
    :param endpoint_id: The ID of the endpoint to retrieve messages for.
    :return: A JSON object that contains information about the in-app message.
    """

    try:
        response = pinpoint_client.get_in_app_messages(
            ApplicationId=project_id,
            EndpointId=endpoint_id)
    except ClientError:
        logger.exception("Couldn't retrieve messages.")
        raise
    else:
        return response

def main():
    project_id = "4c545b28d21a490cb51b0b364example"
    endpoint_id = "c5ac671ef67ee3ad164cf7706example"
    inapp_response = retrieve_inapp_messages(
        boto3.client('pinpoint'), project_id, endpoint_id)
    print(inapp_response)

if __name__ == '__main__':
    main()
```

## Informazioni sulle risposte delle API **GetInAppMessages**

Quando esegui una chiamata all'operazione API [GetInAppMessages](#), essa restituisce l'elenco dei messaggi associati all'endpoint specificato. L'app può quindi eseguire il rendering del messaggio in base ai valori della risposta.

Di seguito è riportato un esempio dell'oggetto JSON restituito quando si esegue una chiamata all'API `GetInAppMessages`:

```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
              "BodyConfig":{
                "Alignment":"CENTER",
                "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
                "TextColor":"#d0021b"
              },
              "HeaderConfig":{
                "Alignment":"CENTER",
                "Header":"Sample In-App Message",
                "TextColor":"#d0021b"
              },
              "ImageUrl":"https://example.com/images/thumbnail.png",
              "PrimaryBtn":{
                "DefaultConfig":{
                  "BackgroundColor":"#d0021b",
                  "BorderRadius":50,
                  "ButtonAction":"CLOSE",
                  "Text":"Dismiss",
                  "TextColor":"#f8e71c"
                }
              }
            }
          ],
          "Layout":"MIDDLE_BANNER"
        },
        "Priority":3,
        "Schedule":{
          "EndDate":"2021-11-06T00:08:05Z",
          "EventFilter":{
            "Dimensions":{
              "Attributes":{
```

```

    },
    "EventType":{
      "DimensionType":"INCLUSIVE",
      "Values":[
        "_session.start"
      ]
    },
    "Metrics":{
      }
    }
  },
  "SessionCap":0,
  "TotalCap":0,
  "TreatmentId":"0"
}
]
}
}

```

Nelle sezioni seguenti vengono fornite ulteriori informazioni sui componenti di questa risposta.

## Oggetto **InAppMessageCampaigns**

L'oggetto `InAppMessageCampaigns` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
<code>CampaignId</code>	Stringa contenente il nome e l'ID univoco della campagna Amazon Pinpoint da cui è stato inviato il messaggio . Il nome precede l'ID della campagna. I due valori sono separati da un trattino (-).	Creato automaticamente da Amazon Pinpoint al momento della creazione della campagna.
<code>TreatmentId</code>	Numero intero che rappresenta l'ID del trattamento della campagna per questo	

Attributo	Descrizione	Dove viene impostato
Priority	<p>messaggio. Se la campagna ha un solo trattamento, il valore è 0.</p> <p>La priorità del messaggio in-app, espressa come numero intero compreso tra 1 e 5, incluso, dove 1 indica la priorità più alta e 5 indica la priorità più bassa.</p>	<p><a href="#">Passaggio 1</a> del processo di creazione della campagna.</p>
InAppMessage	<p><a href="#">Oggetto InAppMessage</a> contenente informazioni su come viene visualizzato il messaggio.</p>	<p>In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.</p>
Schedule	<p>Oggetto Schedule contenente informazioni su quando il messaggio è stato inviato.</p>	<p><a href="#">Passaggio 4</a> del processo di creazione della campagna (se la campagna è stata creata nella console) o dell'oggetto Schedule (se la campagna è stata creata utilizzando l'API o un SDK).</p>
DailyCap	<p>Numero di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato all'utente nell'arco di 24 ore.</p>	<p>Ereditato dalle impostazioni <a href="#">a livello di progetto</a>. Se la campagna include impostazioni che sovrascrivono le impostazioni del progetto, vengono utilizzate le impostazioni della campagna.</p>
SessionCap	<p>Numero di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato all'utente durante una sessione dell'applicazione.</p>	

Attributo	Descrizione	Dove viene impostato
TotalCap	Numero totale di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato a un endpoint per campagna.	

## Oggetto **InAppMessage**

L'oggetto InAppMessage contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
Content	Array contenente un oggetto <a href="#">InAppMessageContent</a> , che descrive il contenuto del messaggio.	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
Layout	Stringa che descrive come verrà visualizzato il messaggio in-app sul dispositivo del destinatario. I valori possibili sono: <ul style="list-style-type: none"><li>• <b>BOTTOM_BANNER</b> : un messaggio visualizzato come banner nella parte inferiore della pagina.</li><li>• <b>TOP_BANNER</b> : un messaggio visualizzato come banner nella parte superiore della pagina.</li><li>• <b>OVERLAYS</b>: un messaggio che copre l'intero schermo.</li></ul>	

Attributo	Descrizione	Dove viene impostato
	<ul style="list-style-type: none"> <li>• <b>MOBILE_FEED</b> : un messaggio visualizzato in una finestra davanti alla pagina.</li> <li>• <b>MIDDLE_BANNER</b> : un messaggio visualizzato come banner nella parte centrale della pagina.</li> <li>• <b>CAROUSEL</b>: un layout scorrevole contenente fino a cinque messaggi univoci.</li> </ul>	

## Oggetto **HeaderConfig**

L'oggetto `HeaderConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
<code>Alignment</code>	Stringa che specifica l'allineamento del testo dell'intestazione. I valori possibili sono <code>LEFT</code> , <code>CENTER</code> e <code>RIGHT</code> .	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
<code>Header</code>	Testo dell'intestazione del messaggio.	
<code>TextColor</code>	Colore del testo dell'intestazione, espresso come codice colore esadecimale (come <code>"#000000"</code> per il nero).	

## Oggetto **BodyConfig**

L'oggetto `BodyConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
Alignment	Stringa che specifica l'allineamento del corpo del messaggio. I valori possibili sono LEFT, CENTER e RIGHT.	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
Body	Il corpo del testo principale del messaggio.	
TextColor	Colore del corpo del testo, espresso sotto forma di stringa contenente un codice colore esadecimale (come "#000000" per il nero).	

## Oggetto **InAppMessageContent**

L'oggetto InAppMessageContent contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
BackgroundColor	Colore di sfondo del messaggio in-app, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
BodyConfig	Oggetto <a href="#">BodyConfig</a> contenente informazioni relative al contenuto del corpo principale del messaggio.	
HeaderConfig	Oggetto <a href="#">HeaderConfig</a> contenente informazioni	

Attributo	Descrizione	Dove viene impostato
	relative all'intestazione o al titolo del messaggio.	
ImageUrl	URL dell'immagine visualizzata nel messaggio.	
PrimaryBtn	Oggetto <a href="#">InAppMessageButton</a> contenente informazioni sul pulsante principale nel messaggio.	
SecondaryBtn	Oggetto <a href="#">InAppMessageButton</a> contenente informazioni sul pulsante secondario nel messaggio. Non presente se il modello di messaggio in-app non specifica un pulsante secondario.	

## Oggetto Schedule

L'oggetto Schedule contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
EndDate	Ora pianificata, in formato ISO 8601, per il termine della campagna.	<a href="#">Passaggio 4</a> del processo di creazione della campagna (se la campagna è stata creata nella console) o dell'oggetto Schedule (se la campagna è stata creata utilizzando l'API o un SDK).
EventFilter	Informazioni sull'evento che causa la visualizzazione del messaggio in-app. Quando generi un evento che corrisponde a una campagna	

Attributo	Descrizione	Dove viene impostato
	in-app Amazon Pinpoint, viene visualizzato il messaggio.	

## Oggetto **InAppMessageButton**

Un oggetto `InAppMessageButton` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
<code>DefaultConfig</code>	Oggetto <a href="#">DefaultButtonConfig</a> contenente le informazioni sulle impostazioni predefinite di un pulsante in un messaggio in-app.	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
Android	Oggetto <a href="#">OverrideButtonConfig</a> che specifica il comportamento del pulsante sui dispositivi Android. Ciò sovrascrive la configurazione predefinita dei pulsanti specificata nell'oggetto <code>DefaultConfig</code> .	
iOS	Oggetto <a href="#">OverrideButtonConfig</a> che specifica il comportamento del pulsante sui dispositivi iOS. Ciò sovrascrive la configurazione predefinita dei pulsanti specificata nell'oggetto <code>DefaultConfig</code> .	
Web	Oggetto <a href="#">OverrideButtonConfig</a> che specifica il comportamento del pulsante nelle app Web. Ciò sovrascrive la	

Attributo	Descrizione	Dove viene impostato
	configurazione predefinita dei pulsanti specificata nell'oggetto <code>DefaultConfig</code> .	

## Oggetto `DefaultButtonConfig`

Un oggetto `DefaultButtonConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
<code>BackgroundColor</code>	Colore di sfondo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.
<code>BorderRadius</code>	Raggio del bordo del pulsante in pixel, espresso come numero intero. Un numero più alto genera angoli più arrotondati.	
<code>ButtonAction</code>	Stringa che descrive l'azione che si verifica quando un destinatario sceglie un pulsante in un messaggio in-app. I valori possibili sono: <ul style="list-style-type: none"> <li>• <code>LINK</code>: un collegamento a una destinazione Web.</li> <li>• <code>DEEP_LINK</code> : un collegamento a una pagina specifica di un'applicazione.</li> <li>• <code>CLOSE</code>: elimina il messaggio .</li> </ul>	

Attributo	Descrizione	Dove viene impostato
Link	URL di destinazione di un pulsante. Non presente per i pulsanti in cui l'attributo ButtonAction è impostato su CLOSE.	
Text	Il testo visualizzato sul pulsante.	
TextColor	Colore del testo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	

## Oggetto **OverrideButtonConfig**

L'oggetto `OverrideButtonConfig` è presente solo se il modello di messaggio in-app utilizza pulsanti di override. Un pulsante di override è un pulsante con una configurazione specifica per un particolare tipo di dispositivo, ad esempio un dispositivo iOS, un dispositivo Android o un browser Web.

Un oggetto `OverrideButtonConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
ButtonAction	Azione che si verifica quando un destinatario sceglie un pulsante in un messaggio in-app. I valori possibili sono: <ul style="list-style-type: none"> <li>LINK: un collegamento a una destinazione Web.</li> </ul>	In base al contenuto del <a href="#">modello di messaggio in-app</a> specificato per la campagna.

Attributo	Descrizione	Dove viene impostato
	<ul style="list-style-type: none"><li>• DEEP_LINK : un collegamento a una pagina specifica di un'applicazione.</li><li>• CLOSE: elimina il messaggio .</li></ul>	
Link	URL di destinazione di un pulsante. Non presente per i pulsanti in cui l'attributo <code>ButtonAction</code> è impostato su CLOSE.	
Text	Il testo visualizzato sul pulsante.	
TextColor	Colore del testo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	

# Convalida dei numeri di telefono in Amazon Pinpoint

Amazon Pinpoint include un servizio di convalida dei numeri di telefono che è possibile utilizzare per stabilire se un numero di telefono è valido e per ottenere ulteriori informazioni su tale numero. Ad esempio, quando si utilizza il servizio di convalida dei numeri di telefono, restituisce le informazioni riportate di seguito.

- Il numero di telefono nel formato E.164.
- Il tipo di numero di telefono (ad esempio mobile, rete fissa o VoIP).
- La città e il paese collegati al numero di telefono.
- Il fornitore di servizi associato al numero di telefono.

Non vi sono costi aggiuntivi per l'utilizzo del servizio di convalida dei numeri di telefono. Per ulteriori informazioni, consulta [Prezzi di Amazon Pinpoint](#).

## Important

Per i numeri di telefono la cui origine è negli Stati Uniti e in Canada, l'API di convalida dei numeri di telefono non restituirà più dati per City, County, Timezone e ZipCode.

## Casi d'uso della funzione di convalida dei numeri di telefono

È possibile utilizzare il servizio di convalida dei numeri di telefono per diversi casi d'utilizzo, tra cui i seguenti:

- Verifica di numeri di telefono forniti su un modulo Web: se utilizzi moduli basati sul Web per raccogliere informazioni di contatto dai clienti, devo convalidare il numero di telefono fornito dai clienti prima di inviare il modulo. Utilizza il back-end del sito Web per convalidare il numero utilizzando l'API di Amazon Pinpoint. La risposta dell'API indica se il numero non è valido, ad esempio se il numero di telefono è formattato in modo errato. Se verifichi che il numero di telefono che il cliente ha fornito non è valido, il modulo di richiesta del cliente Web può chiedere al cliente di fornire un numero diverso.
- Pulizia del database di contatti esistente: se disponi di un database dei numeri di telefono dei clienti, puoi convalidare ogni numero di telefono e aggiornare il database in base ai risultati. Ad

esempio, se trovi endpoint con i numeri di telefono che non sono in grado di ricevere messaggi SMS, puoi modificare la proprietà `ChannelType` per l'endpoint da SMS a VOICE. Puoi prima convalidare il numero di telefono e poi aggiornare la proprietà `ChannelType` per gli endpoint nuovi o esistenti seguendo le istruzioni in [Aggiunta di endpoint ad Amazon Pinpoint](#) per un singolo endpoint o in [Aggiunta di un batch di endpoint ad Amazon Pinpoint](#) per più endpoint.

- Scelta del canale corretto prima di inviare un messaggio: se intendi inviare un messaggio SMS, ma ti accorgi che il numero di destinazione non è valido, puoi inviare un messaggio al destinatario tramite un canale diverso. Ad esempio, se l'endpoint non è in grado di ricevere messaggi SMS, puoi inviare un messaggio vocale.

## Utilizzo del servizio di convalida dei numeri di telefono

L'esempio seguente mostra come convalidare un numero di telefono utilizzando il AWS CLI. Per ulteriori informazioni, vedere [phone-number-validate](#) nel AWS CLI Command Reference. Ad esempio, le risposte di convalida, vedere [Risposte di convalida dei numeri di telefono](#). Per ulteriori informazioni sulla configurazione di AWS CLI, vedere [Configure the AWS CLI nella Guida](#) per l'[AWS Command Line Interface utente](#).

Per utilizzare il servizio di convalida del numero di telefono utilizzando AWS CLI

- Nella riga di comando, inserisci il comando seguente:

```
aws pinpoint phone-number-validate --number-validate-request  
PhoneNumber=+442079460881,IsoCountryCode=GB
```

Nel comando precedente, sostituisci **+442079460881** con il numero di telefono che desideri convalidare e **GB** con il codice ISO a due cifre del paese o dell'area.

### Note

Quando si fornisce un numero di telefono al servizio di convalida del numero di telefono, includere sempre il prefisso internazionale. Se non si include il prefisso internazionale, il servizio può restituire informazioni per un numero di telefono in un paese diverso.

*È possibile inserire dei trattini nel numero di telefono, ad esempio **+44-207-946-0881**.*

## Risposte di convalida dei numeri di telefono

Le informazioni che il servizio di convalida dei numeri di telefono fornisce variano leggermente in base ai dati disponibili per il numero di telefono fornito dall'utente. Questa sezione contiene esempi di risposte che il servizio di convalida dei numeri di telefono restituisce.

### Note

I dati forniti dal servizio di convalida dei numeri di telefono si basano su informazioni fornite dai provider di servizi di telecomunicazione e da altre entità in tutto il mondo. I provider in alcuni paesi potrebbero aggiornare queste informazioni meno frequentemente rispetto a provider di altri paesi. Ad esempio, se invii una richiesta di convalida di un numero di telefono cellulare e il numero fornito è stato trasferito da un operatore mobile a un altro, la risposta dal servizio di convalida dei numeri di telefono potrebbe includere il nome dell'operatore originale, rispetto a quello corrente.

### Numeri di cellulare validi

Quando invii una richiesta al servizio di convalida dei numeri di telefono e il numero di telefono è un numero di telefono cellulare valido, restituisce informazioni simili al seguente esempio:

```
{
  "NumberValidateResponse": {
    "Carrier": "ExampleCorp Mobile",
    "City": "Seattle",
    "CleansedPhoneNumberE164": "+12065550142",
    "CleansedPhoneNumberNational": "2065550142",
    "Country": "United States",
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "OriginalPhoneNumber": "+12065550142",
    "PhoneType": "MOBILE",
    "PhoneTypeCode": 0,
    "Timezone": "America/Los_Angeles",
    "ZipCode": "98101"
  }
}
```

### Numeri di telefono fisso validi

Se la richiesta contiene un numero di telefono fisso valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
  "CountryCodeIso2": "US",
  "CountryCodeNumeric": "1",
  "Country": "United States",
  "City": "Santa Clara",
  "ZipCode": "95037",
  "Timezone": "America/Los_Angeles",
  "CleansedPhoneNumberNational": "4085550101",
  "CleansedPhoneNumberE164": "14085550101",
  "Carrier": "AnyCompany",
  "PhoneTypeCode": 1,
  "PhoneType": "LANDLINE",
  "OriginalPhoneNumber": "+14085550101"
}
```

### Numeri di telefono VoIP validi

Se la richiesta contiene un numero di telefono VoIP valido (Voice over Internet Protocol), il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
  "NumberValidateResponse": {
    "Carrier": "ExampleCorp",
    "City": "Countrywide",
    "CleansedPhoneNumberE164": "+441514960001",
    "CleansedPhoneNumberNational": "1514960001",
    "Country": "United Kingdom",
    "CountryCodeIso2": "GB",
    "CountryCodeNumeric": "44",
    "OriginalPhoneNumber": "+441514960001",
    "PhoneType": "VOIP",
    "PhoneTypeCode": 2
  }
}
```

### Numeri di telefono non validi

Se la richiesta contiene un numero di telefono non valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
  "NumberValidateResponse": {
    "CleansedPhoneNumberE164": "+44163296076",
    "CleansedPhoneNumberNational": "163296076",
    "Country": "United Kingdom",
    "CountryCodeIso2": "GB",
    "CountryCodeNumeric": "44",
    "OriginalPhoneNumber": "+440163296076",
    "PhoneType": "INVALID",
    "PhoneTypeCode": 3
  }
}
```

Notare che la proprietà `PhoneType` in questo tipo di risposta indica che questo numero di telefono è `INVALID` e che non include informazioni sull'operatore o su un percorso associato al numero di telefono. Si consiglia di evitare l'invio di SMS o messaggi vocali ai numeri di telefono in cui `PhoneType` è `INVALID`, perché questi numeri non appartengono a destinatari effettivi.

### Altri numeri di telefono

Occasionalmente, la risposta dal servizio di convalida dei numeri di telefono include un valore `PhoneType` di `OTHER`. Il servizio può restituire questo tipo di risposta nelle seguenti situazioni:

- Il numero di telefono è un numero gratuito.
- Il numero di telefono è riservato all'utilizzo in programmi televisivi e filmati, ad esempio numeri di telefono del Nord America che iniziano con 555.
- Il numero di telefono include un prefisso che non è attualmente in uso, ad esempio il prefisso 999 in America del Nord.
- Il numero di telefono è riservato a un altro scopo.

L'esempio seguente mostra la risposta che il servizio di convalida dei numeri di telefono fornisce quando la richiesta include un numero di telefono dell'America del Nord fittizio:

```
{
  "NumberValidateResponse": {
    "Carrier": "Multiple OCN Listing",
    "CleansedPhoneNumberE164": "+14255550199",
    "CleansedPhoneNumberNational": "4255550199",
    "Country": "United States",
  }
}
```

```
"CountryCodeIso2": "US",
"CountryCodeNumeric": "1",
"OriginalPhoneNumber": "+14255550199",
"PhoneType": "OTHER",
"PhoneTypeCode": 4,
"Timezone": "America/Los_Angeles"
}
}
```

## Numeri di telefono prepagati

Se la richiesta contiene un numero di telefono prepagato valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
  "NumberValidateResponse": {
    "Carrier": "ExampleCorp",
    "City": "Countrywide",
    "CleansedPhoneNumberE164": "+14255550199",
    "CleansedPhoneNumberNational": "4255550199",
    "Country": "United States",
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "OriginalPhoneNumber": "+14255550199",
    "PhoneType": "PREPAID",
    "PhoneTypeCode": 5
  }
}
```

Per ulteriori informazioni sulle informazioni contenute in queste risposte, consulta l'articolo relativo alla [convalida dei numeri di telefono](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

# Invio di messaggi delle transazioni dalle tue app

È possibile usare l'API Amazon Pinpoint e AWS SDK per l'invio di messaggi transazionali direttamente dall'app. I messaggi transazionali sono messaggi inviati a destinatari specifici, a differenza dei messaggi inviati a segmenti. Sono diversi i motivi per cui potresti voler inviare messaggi transazionali anziché messaggi basati su campagne. Ad esempio, è possibile inviare una conferma dell'ordine via e-mail quando un cliente effettua un ordine. È inoltre possibile inviare una password monouso tramite SMS o voce utilizzabile dal cliente per completare il processo di creazione di un account per il tuo servizio.

Questa sezione include il codice di esempio in diversi linguaggi di programmazione che puoi utilizzare per iniziare a inviare messaggi vocali, messaggi SMS ed e-mail transazionali.

Argomenti in questa sezione:

- [Invio di messaggi e-mail transazionali](#)
- [Invio di messaggi SMS](#)
- [Invio di messaggi vocali](#)
- [Invio di notifiche push](#)

## Invio di messaggi e-mail transazionali

Questa sezione fornisce esempi di codice completi utilizzabili per inviare messaggi e-mail transazionali tramite Amazon Pinpoint.

- [Utilizzando l' SendMessages operazione nell'API Amazon Pinpoint: puoi utilizzare l'SendMessagesoperazione nell'API](#) Amazon Pinpoint per inviare messaggi in tutti i canali supportati da Amazon Pinpoint, inclusi i canali di notifica push, SMS, voce ed e-mail.

Il vantaggio di utilizzare questa operazione è che la sintassi richiesta per l'invio di messaggi è molto simile in tutti i canali. In questo modo è più semplice riutilizzare il codice esistente. L'operazione SendMessages consente anche di sostituire i contenuti nei messaggi e-mail e di inviare e-mail agli ID degli endpoint Amazon Pinpoint anziché a specifici indirizzi e-mail.

Questa sezione include il codice di esempio in diversi linguaggi di programmazione che puoi utilizzare per iniziare a inviare e-mail transazionali.

Argomenti in questa sezione:

- [Scelta di un metodo per l'invio di e-mail](#)
- [Scegliere tra Amazon Pinpoint e Amazon Simple Email Service \(SES\)](#)
- [Invio di e-mail tramite l'API Amazon Pinpoint](#)
- [Invio di un'e-mail con intestazioni di annullamento dell'iscrizione](#)

## Scelta di un metodo per l'invio di e-mail

Il miglior metodo da utilizzare per l'invio di e-mail transazionali varia a seconda del caso d'uso specifico. Ad esempio, se devi inviare e-mail utilizzando un'applicazione di terze parti o se non è disponibile un AWS SDK per il tuo linguaggio di programmazione, potresti dover utilizzare l'interfaccia SMTP. Se desideri inviare messaggi in altri canali supportati da Amazon Pinpoint e desideri utilizzare il codice consistente per effettuare tali richieste, devi utilizzare l'operazione `SendMessage` nell'API Amazon Pinpoint.

## Scegliere tra Amazon Pinpoint e Amazon Simple Email Service (SES)

Se invii un gran numero di e-mail transazionali, come le conferme di acquisto o i messaggi di reimpostazione della password, prendi in considerazione l'utilizzo di Amazon SES. Amazon SES ha un'API e un'interfaccia SMTP, entrambe adatte per l'invio di e-mail da applicazioni o servizi. Offre inoltre funzionalità di posta elettronica aggiuntive, tra cui funzionalità di ricezione e-mail, set di configurazione e funzionalità di autorizzazione per l'invio.

Amazon SES include anche un'interfaccia SMTP che puoi integrare con le applicazioni di terze parti esistenti, inclusi i servizi di gestione delle relazioni con i clienti (CRM) come Salesforce. Per maggiori informazioni sull'invio di e-mail tramite Amazon SES, consulta [Amazon Simple Email Service Developer Guide](#).

## Invio di e-mail tramite l'API Amazon Pinpoint

Questa sezione contiene esempi di codice completi che puoi utilizzare per inviare e-mail tramite l'API Amazon Pinpoint utilizzando un AWS SDK.

C#

Utilizza questo esempio per l'invio di e-mail tramite l'[AWS SDK for .NET](#). Questo esempio si basa sul presupposto che l' AWS SDK for .NET sia già stato installato e configurato. Per ulteriori

informazioni, consulta l'argomento relativo alle [nozioni di base su AWS SDK for .NET](#) nella Guida per gli sviluppatori di AWS SDK for .NET .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione delle credenziali AWS](#) nella Guida per gli sviluppatori di AWS SDK for .NET .

Questo esempio di codice è stato testato utilizzando la AWS SDK for .NET versione 3.3.29.13 e la versione di runtime 2.1.2 di .NET Core.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;
```

```

    // The Amazon Pinpoint project/application ID to use when you send this
message.
    // Make sure that the SMS channel is enabled for the project or application
// that you choose.
    string appId = configuration["AppId"]!;

    try
    {
        await SendEmailMessage(region, appId, toAddress, senderAddress);
    }
    catch (Exception ex)
    {
        Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
    }
}

public static async Task<MessageResponse> SendEmailMessage(
    string region, string appId, string toAddress, string senderAddress)
{
    var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

    // The subject line of the email.
    string subject = "Amazon Pinpoint Email test";

    // The body of the email for recipients whose email clients don't
// support HTML content.
    string textBody = @"Amazon Pinpoint Email Test (.NET)"
        + "\n-----"
        + "\nThis email was sent using the Amazon Pinpoint API
using the AWS SDK for .NET.";

    // The body of the email for recipients whose email clients support
// HTML content.
    string htmlBody = @"<html>"
        + "\n<head></head>"
        + "\n<body>"
        + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
        + "\n  <p>This email was sent using the "
        + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
Pinpoint</a> API "

```

```
        + "\n    using the <a href='https://aws.amazon.com/sdk-  
for-net/'>AWS SDK for .NET</a>"  
        + "\n </p>"  
        + "\n</body>"  
        + "\n</html>";  
  
// The character encoding the you want to use for the subject line and  
// message body of the email.  
string charset = "UTF-8";  
  
var sendRequest = new SendMessagesRequest  
{  
    ApplicationId = appId,  
    MessageRequest = new MessageRequest  
    {  
        Addresses = new Dictionary<string, AddressConfiguration>  
        {  
            {  
                toAddress,  
                new AddressConfiguration  
                {  
                    ChannelType = ChannelType.EMAIL  
                }  
            }  
        },  
        MessageConfiguration = new DirectMessageConfiguration  
        {  
            EmailMessage = new EmailMessage  
            {  
                FromAddress = senderAddress,  
                SimpleEmail = new SimpleEmail  
                {  
                    HtmlPart = new SimpleEmailPart  
                    {  
                        Charset = charset,  
                        Data = htmlBody  
                    },  
                    TextPart = new SimpleEmailPart  
                    {  
                        Charset = charset,  
                        Data = textBody  
                    },  
                    Subject = new SimpleEmailPart  
                    {
```

```
        Charset = charset,
        Data = subject
    }
}
}
};
Console.WriteLine("Sending message...");
SendMessageResponse response = await client.SendMessageAsync(sendRequest);
Console.WriteLine("Message sent!");
return response.MessageResponse;
}
}
```

## Java

Utilizza questo esempio per l'invio di e-mail tramite l'[AWS SDK for Java](#). Questo esempio si basa sul presupposto che l' AWS SDK for Java 2.x sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base](#) nella Guida per gli sviluppatori di AWS SDK for Java 2.x .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione della regione e delle credenziali predefinite](#) nella Guida per gli sviluppatori di AWS SDK for Java .

Questo esempio di codice è stato testato utilizzando la AWS SDK for Java versione 2.3.1 e la versione OpenJDK 11.0.1.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessageRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessage {
```

```

// The character encoding the you want to use for the subject line and
// message body of the email.
public static String charset = "UTF-8";

// The body of the email for recipients whose email clients support HTML
content.
static final String body = ""
    Amazon Pinpoint test (AWS SDK for Java 2.x)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
for Java 2.x

    """;

public static void main(String[] args) {
    final String usage = ""

        Usage:    <subject> <appId> <senderAddress>
<toAddress>

        Where:
            subject - The email subject to use.
            senderAddress - The from address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
            toAddress - The to address. This address has to be verified in Amazon
Pinpoint in the region you're using to send email\s
    """;

    if (args.length != 3) {
        System.out.println(usage);
        System.exit(1);
    }

    String subject = args[0];
    String senderAddress = args[1];
    String toAddress = args[2];
    System.out.println("Sending a message");
    PinpointEmailClient pinpoint = PinpointEmailClient.builder()
        .region(Region.US_EAST_1)
        .build();

    sendEmail(pinpoint, subject, senderAddress, toAddress);
    System.out.println("Email was sent");
    pinpoint.close();

```

```
}

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
    try {
        Content content = Content.builder()
            .data(body)
            .build();

        Body messageBody = Body.builder()
            .text(content)
            .build();

        Message message = Message.builder()
            .body(messageBody)
            .subject(Content.builder().data(subject).build())
            .build();

        Destination destination = Destination.builder()
            .toAddresses(toAddress)
            .build();

        EmailContent emailContent = EmailContent.builder()
            .simple(message)
            .build();

        SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
            .fromEmailAddress(senderAddress)
            .destination(destination)
            .content(emailContent)
            .build();

        pinpointEmailClient.sendEmail(sendEmailRequest);
        System.out.println("Message Sent");

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

[Per l'esempio completo dell'SDK, consulta `.java on. SendEmailMessage` GitHub](#)

## JavaScript (Node.js)

Utilizzate questo esempio per inviare e-mail utilizzando l'[AWS SDK for JavaScript](#) in Node.js. L'esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js. Per ulteriori informazioni, consulta [Guida introduttiva](#) all'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta [Impostazione delle credenziali](#) nell'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio di codice è stato testato utilizzando l'SDK per JavaScript Node.js versione 2.388.0 e Node.js versione 11.7.0.

```
"use strict";

const AWS = require("aws-sdk");

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)`
```

```
-----  
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in  
Node.js.  
For more information, see https://aws.amazon.com/sdk-for-node-js/;  
  
// The body of the email for recipients whose email clients support HTML content.  
var body_html = `  
<head></head>  
<body>  
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>  
  <p>This email was sent with  
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the  
    <a href='https://aws.amazon.com/sdk-for-node-js/'>  
      AWS SDK for JavaScript in Node.js</a>.</p>  
</body>  
</html>`;  
  
// The character encoding the you want to use for the subject line and  
// message body of the email.  
var charset = "UTF-8";  
  
// Specify that you're using a shared credentials file.  
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });  
AWS.config.credentials = credentials;  
  
// Specify the region.  
AWS.config.update({ region: aws_region });  
  
//Create a new Pinpoint object.  
var pinpoint = new AWS.Pinpoint();  
  
// Specify the parameters to pass to the API.  
var params = {  
  ApplicationId: appId,  
  MessageRequest: {  
    Addresses: {  
      [toAddress]: {  
        ChannelType: "EMAIL",  
      },  
    },  
    MessageConfiguration: {  
      EmailMessage: {  
        FromAddress: senderAddress,  
        SimpleEmail: {
```

```

        Subject: {
            Charset: charset,
            Data: subject,
        },
        HtmlPart: {
            Charset: charset,
            Data: body_html,
        },
        TextPart: {
            Charset: charset,
            Data: body_text,
        },
    },
},
},
},
};

//Try to send the email.
pinpoint.sendMessage(params, function (err, data) {
    // If something goes wrong, print an error message.
    if (err) {
        console.log(err.message);
    } else {
        console.log(
            "Email sent! Message ID: ",
            data["MessageResponse"]["Result"][toAddress]["MessageId"]
        );
    }
});

```

## Python

Utilizza questo esempio per l'invio di e-mail tramite l'[AWS SDK for Python \(Boto3\)](#). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato. Per ulteriori informazioni, consulta [Quickstart](#) nella documentazione di riferimento delle API SDK AWS per Python (Boto3).

```

import logging
import boto3
from botocore.exceptions import ClientError

```

```
logger = logging.getLogger(__name__)

def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
    account
                           is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
    can
                           display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
    to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
```

```

        "FromAddress": sender,
        "SimpleEmail": {
            "Subject": {"Charset": char_set, "Data": subject},
            "HtmlPart": {"Charset": char_set, "Data": html_message},
            "TextPart": {"Charset": char_set, "Data": text_message},
        },
    },
)
except ClientError:
    logger.exception("Couldn't send email.")
    raise
else:
    return {
        to_address: message["MessageId"]
        for to_address, message in response["MessageResponse"]["Result"].items()
    }

def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -----
    This email was sent with Amazon Pinpoint using the AWS SDK for Python (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
    """
    html_message = """<html>
    <head></head>
    <body>
    <h1>Amazon Pinpoint Test (SDK for Python (Boto3))</h1>
    <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
    <a href='https://aws.amazon.com/sdk-for-python/'>
    AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
    """

    print("Sending email.")

```

```
message_ids = send_email_message(
    boto3.client("pinpoint"),
    app_id,
    sender,
    [to_address],
    char_set,
    subject,
    html_message,
    text_message,
)
print(f"Message sent! Message IDs: {message_ids}")

if __name__ == "__main__":
    main()
```

È inoltre possibile utilizzare modelli di messaggi per inviare messaggi e-mail, come illustrato nell'esempio seguente:

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
    template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
    message.
    :param sender: The "From" address. This address must be verified in
    Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
    account is in the sandbox, these addresses must be
    verified.
    :param template_name: The name of the email template to use when sending the
    message.
```

```
:param template_version: The version number of the message template.

:return: A dict of to_addresses and their message IDs.
"""
try:
    response = pinpoint_client.send_messages(
        ApplicationId=project_id,
        MessageRequest={
            "Addresses": {
                to_address: {"ChannelType": "EMAIL"} for to_address in
to_addresses
            },
            "MessageConfiguration": {"EmailMessage": {"FromAddress": sender}},
            "TemplateConfiguration": {
                "EmailTemplate": {
                    "Name": template_name,
                    "Version": template_version,
                }
            },
        },
    )
except ClientError:
    logger.exception("Couldn't send email.")
    raise
else:
    return {
        to_address: message["MessageId"]
        for to_address, message in response["MessageResponse"]["Result"].items()
    }

def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
```

```
        template_name,  
        template_version,  
    )  
    print(f"Message sent! Message IDs: {message_ids}")  
  
if __name__ == "__main__":  
    main()
```

Questi esempi presuppongono che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta [Credenziali](#) nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

## Invio di un'e-mail con intestazioni di annullamento dell'iscrizione

### Note

Prima di poter utilizzare le intestazioni delle e-mail, è necessario impostare un ruolo di invio dell'orchestrazione delle e-mail se si inviano e-mail da una campagna o da un percorso. Per l'invio diretto di e-mail, devi disporre delle autorizzazioni per `ses:SendEmail` e `ses:SendRawEmail`. Per ulteriori informazioni, consulta [Creazione di un ruolo di invio per l'orchestrazione delle e-mail nella Guida](#) per l'utente di [Amazon Pinpoint](#).

L'inclusione di un link di annullamento dell'iscrizione nell'e-mail rappresenta una best practice e in alcuni paesi è un requisito legale. Per aggiungere un link di annullamento dell'iscrizione con un clic, aggiungi le seguenti intestazioni:

1. Imposta il nome dell'intestazione `List-Unsubscribe` e imposta il valore sul link di annullamento dell'iscrizione. Il link deve supportare le richieste HTTP POST per elaborare la richiesta di annullamento dell'iscrizione del destinatario.
2. Imposta il nome dell'intestazione su **List-Unsubscribe-Post** e imposta il valore su `List-Unsubscribe=One-Click`

Puoi aggiungere fino a 15 intestazioni a un messaggio di posta elettronica. Per un elenco delle intestazioni supportate, consulta i [campi di intestazione di Amazon SES](#) nella [Amazon Simple Email Service Developer Guide](#).

L'esempio seguente mostra come inviare un messaggio e-mail con intestazioni di annullamento dell'iscrizione utilizzando il [AWS Command Line Interface](#). [Per ulteriori informazioni sulla configurazione di AWS CLI, vedere Configure the AWS CLI nella Guida per l'utente.](#) [AWS Command Line Interface](#)

Nel comando seguente, effettuate le seguenti operazioni:

- Sostituisci *AppId* con l'ID dell'applicazione.
- Sostituisci *richard\_roe@example.com* con l'indirizzo e-mail del destinatario.
- Sostituisci *https://example.com/unsub* con il tuo link di annullamento dell'iscrizione.
- Sostituisci *example123456* con un identificatore univoco per il destinatario.

```
aws pinpoint send-messages --application-id AppId --message-request '{
  "Addresses": {
    "richard_roe@example.com": {
      "ChannelType": "EMAIL"
    }
  },
  "MessageConfiguration": {
    "EmailMessage": {
      "Substitutions": {
        "url": [
          "https://example.com/unsub"
        ],
        "id1": [
          "/example123456"
        ]
      },
      "SimpleEmail": {
        "TextPart": {
          "Data": "Sample email message with an subscribe header",
          "Charset": "UTF-8"
        },
        "Subject": {
          "Data": "Hello",
          "Charset": "UTF-8"
        },
        "Headers": [
          {
            "Name": "List-Unsubscribe",
            "Value": "{{url}}{{id1}}"
          }
        ]
      }
    }
  }
}
```

```
    },
    {
      "Name": "List-Unsubscribe-Post",
      "Value": "List-Unsubscribe=One-Click"
    }
  ]
}
}
}
}'
```

## Invio di messaggi SMS

È possibile utilizzare l'API Amazon Pinpoint per inviare messaggi SMS (messaggi di testo) a ID degli endpoint e numeri di telefono specifici. Questa sezione contiene esempi di codice completi che puoi utilizzare per inviare messaggi SMS tramite l'API Amazon Pinpoint utilizzando un AWS SDK.

### C#

Utilizza questo esempio per inviare un messaggio SMS tramite l'[AWS SDK for .NET](#). Questo esempio si basa sul presupposto che l' AWS SDK for .NET sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base](#) nella Guida per gli sviluppatori di AWS SDK for .NET .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione delle credenziali AWS](#) nella Guida per gli sviluppatori di AWS SDK for .NET .

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
```

```
var configuration = new ConfigurationBuilder()
    .SetBasePath(Directory.GetCurrentDirectory())
    .AddJsonFile("settings.json") // Load test settings from .json file.
    .AddJsonFile("settings.local.json",
        true) // Optionally load local settings.
    .Build();

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
string region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon
Pinpoint
// account. For best results, specify long codes in E.164 format.
string originationNumber = configuration["OriginationNumber"]!;

// The recipient's phone number. For best results, you should specify the
// phone number in E.164 format.
string destinationNumber = configuration["DestinationNumber"]!;

// The Pinpoint project/ application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
string appId = configuration["AppId"]!;

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
MessageType messageType = MessageType.TRANSACTIONAL;

// The registered keyword associated with the originating short code.
string? registeredKeyword = configuration["RegisteredKeyword"];

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
string? senderId = configuration["SenderId"];

try
{
    var response = await SendSmsMessage(region, appId, destinationNumber,
```

```

        originationNumber, registeredKeyword, senderId, messageType);
        Console.WriteLine($"Message sent to
{response.MessageResponse.Result.Count} recipient(s).");
        foreach (var messageResultValue in
            response.MessageResponse.Result.Select(r => r.Value))
        {
            Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
    }
}

public static async Task<SendMessagesResponse> SendSmsMessage(
    string region, string appId, string destinationNumber, string
originationNumber,
    string? keyword, string? senderId, MessageType messageType)
{
    // The content of the SMS message.
    string message = "This message was sent through Amazon Pinpoint using" +
        " the AWS SDK for .NET. Reply STOP to opt out.";

    var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

    SendMessagesRequest sendRequest = new SendMessagesRequest
    {
        ApplicationId = appId,
        MessageRequest = new MessageRequest
        {
            Addresses =
                new Dictionary<string, AddressConfiguration>
                {
                    {
                        destinationNumber,
                        new AddressConfiguration { ChannelType =
ChannelType.SMS }
                    }
                }
        }
    }
}

```

```
        },
        MessageConfiguration = new DirectMessageConfiguration
        {
            SMSMessage = new SMSMessage
            {
                Body = message,
                MessageType = MessageType.TRANSACTIONAL,
                OriginationNumber = originationNumber,
                SenderId = senderId,
                Keyword = keyword
            }
        }
    };
    SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
    return response;
}
}
```

## Java

Utilizza questo esempio per inviare un messaggio SMS tramite l'[AWS SDK for Java](#). In questo esempio si presume che SDK per Java sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base](#) nella Guida per gli sviluppatori di AWS SDK for Java .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione della regione e delle credenziali predefinite](#) nella Guida per gli sviluppatori di AWS SDK for Java .

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
```

```
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
    public static String senderId = "MySenderId";

    public static void main(String[] args) {
        final String usage = ""

```

```
Usage: <message> <appId> <originationNumber>
<destinationNumber>\s
```

Where:

message - The body of the message to send.

appId - The Amazon Pinpoint project/application ID to use when you send this message.

originationNumber - The phone number or short code that you specify has to be associated with your Amazon Pinpoint account. For best results, specify long codes in E.164 format (for example, +1-555-555-5654).

destinationNumber - The recipient's phone number.

For best results, you should specify the phone number in E.164 format (for example, +1-555-555-5654).\s

```
""";
```

```
    if (args.length != 4) {
        System.out.println(usage);
        System.exit(1);
    }

    String message = args[0];
    String appId = args[1];
    String originationNumber = args[2];
    String destinationNumber = args[3];
    System.out.println("Sending a message");
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
    pinpoint.close();
}

public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
    String originationNumber,
    String destinationNumber) {
    try {
        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
        AddressConfiguration addConfig =
AddressConfiguration.builder()
            .channelType(ChannelType.SMS)
```

```
        .build();

        addressMap.put(destinationNumber, addConfig);
        SMSMessage smsMessage = SMSMessage.builder()
            .body(message)
            .messageType(messageType)
            .originationNumber(originationNumber)
            .senderId(senderId)
            .keyword(registeredKeyword)
            .build();

        // Create a DirectMessageConfiguration object.
        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
            .smsMessage(smsMessage)
            .build();

        MessageRequest msgReq = MessageRequest.builder()
            .addresses(addressMap)
            .messageConfiguration(direct)
            .build();

        // create a SendMessagesRequest object
        SendMessagesRequest request = SendMessagesRequest.builder()
            .applicationId(appId)
            .messageRequest(msgReq)
            .build();

        SendMessagesResponse response =
pinpoint.sendMessage(request);
        MessageResponse msg1 = response.messageResponse();
        Map map1 = msg1.result();

        // Write out the result of sendMessage.
        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

[Per l'esempio completo dell'SDK, consulta SendMessage .java on. GitHub](#)

## JavaScript (Node.js)

Utilizzate questo esempio per inviare un messaggio SMS utilizzando l'[AWS SDK for JavaScript](#) in Node.js. L'esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js. Per ulteriori informazioni, consulta [Guida introduttiva](#) all'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta [Impostazione delle credenziali](#) nell'AWS SDK for JavaScript in Node.js Developer Guide.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number. For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";
```

```
// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
var senderId = "MySenderId";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
  },
  MessageConfiguration: {
    SMSMessage: {
      Body: message,
      Keyword: registeredKeyword,
      MessageType: messageType,
      OriginationNumber: originationNumber,
      SenderId: senderId,
    },
  },
},
```

```
};

//Try to send the message.
pinpoint.sendMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
      data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

## Python

Utilizza questo esempio per inviare un messaggio SMS tramite l'[AWS SDK for Python \(Boto3\)](#). Questo esempio si basa sul presupposto che SDK per Python sia già stato installato e configurato. Per ulteriori informazioni, consulta [Quickstart](#) in [AWS SDK for Python \(Boto3\) Getting Started](#).

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.
```

```

:param pinpoint_client: A Boto3 Pinpoint client.
:param app_id: The Amazon Pinpoint project/application ID to use when you send
               this message. The SMS channel must be enabled for the project or
               application.
:param destination_number: The recipient's phone number in E.164 format.
:param origination_number: The phone number to send the message from. This phone
                           number must be associated with your Amazon Pinpoint
                           account and be in E.164 format.
:param message: The content of the SMS message.
:param message_type: The type of SMS message that you want to send. If you send
                    time-sensitive content, specify TRANSACTIONAL. If you send
                    marketing-related content, specify PROMOTIONAL.
:return: The ID of the message.
"""
try:
    response = pinpoint_client.send_messages(
        ApplicationId=app_id,
        MessageRequest={
            "Addresses": {destination_number: {"ChannelType": "SMS"}},
            "MessageConfiguration": {
                "SMSMessage": {
                    "Body": message,
                    "MessageType": message_type,
                    "OriginationNumber": origination_number,
                }
            },
        },
    )
except ClientError:
    logger.exception("Couldn't send message.")
    raise
else:
    return response["MessageResponse"]["Result"][destination_number]
["MessageId"]

def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
"
        "Python (Boto 3)."
```

```
)
message_type = "TRANSACTIONAL"

print("Sending SMS message.")
message_id = send_sms_message(
    boto3.client("pinpoint"),
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
)
print(f"Message sent! Message ID: {message_id}.")

if __name__ == "__main__":
    main()
```

È inoltre possibile utilizzare modelli di messaggi per inviare messaggi SMS, come illustrato nell'esempio seguente:

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
```

```

:param message_type: The type of SMS message (promotional or transactional).
:param origination_number: The phone number that the message is sent from.
:param template_name: The name of the SMS template to use when sending the
message.
:param template_version: The version number of the message template.

:return The ID of the message.
"""
try:
    response = pinpoint_client.send_messages(
        ApplicationId=project_id,
        MessageRequest={
            "Addresses": {destination_number: {"ChannelType": "SMS"}},
            "MessageConfiguration": {
                "SMSMessage": {
                    "MessageType": message_type,
                    "OriginationNumber": origination_number,
                }
            },
            "TemplateConfiguration": {
                "SMSTemplate": {"Name": template_name, "Version":
template_version}
            },
        },
    )

except ClientError:
    logger.exception("Couldn't send message.")
    raise
else:
    return response["MessageResponse"]["Result"][destination_number]
["MessageId"]

def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),

```

```
        project_id,\n        destination_number,\n        message_type,\n        origination_number,\n        template_name,\n        template_version,\n    )\n    print(f"Message sent! Message ID: {message_id}.")\n\nif __name__ == "__main__":\n    main()
```

Questi esempi presuppongono che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per informazioni dettagliate, consulta [Credentials](#) nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

## Invio di messaggi vocali

Puoi utilizzare l'API Amazon Pinpoint per inviare messaggi vocali a numeri di telefono specifici. Questa sezione contiene esempi di codice completo che puoi usare per inviare messaggi vocali mediante l'API SMS e Voce Amazon Pinpoint utilizzando AWS SDK.

### Java

Utilizza questo esempio per l'invio di un messaggio vocale tramite l'[AWS SDK for Java](#). In questo esempio si presume che SDK per Java sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle [nozioni di base](#) nella Guida per gli sviluppatori di AWS SDK for Java.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta [Configurazione delle credenziali e della regione AWS per lo sviluppo](#) nella Guida per gli sviluppatori di AWS SDK for Java.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;\nimport software.amazon.awssdk.regions.Region;\nimport software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;\nimport software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
```

```
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
    list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";
```

```

    // The content of the message. This example uses SSML to customize and
control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
        + "<emphasis>Amazon Pinpoint</emphasis> "
        + "using the <break strength='weak'/>AWS "
        + "SDK for Java. "
        + "<amazon:effect phonation='soft'>Thank "
        + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = ""

            Usage:  <originationNumber> <destinationNumber>\s

            Where:
                originationNumber - The phone number or short code
that you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                destinationNumber - The recipient's phone number.
For best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s

            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String originationNumber = args[0];
        String destinationNumber = args[1];
        System.out.println("Sending a voice message");

        // Set the content type to application/json.
        List<String> listVal = new ArrayList<>();
        listVal.add("application/json");
        Map<String, List<String>> values = new HashMap<>();
        values.put("Content-Type", listVal);

        ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()

```

```
        .headers(values)
        .build();

    PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
        .overrideConfiguration(config2)
        .region(Region.US_EAST_1)
        .build();

    sendVoiceMsg(client, originationNumber, destinationNumber);
    client.close();
}

public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
    String destinationNumber) {
    try {
        SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
            .languageCode(languageCode)
            .text(ssmlMessage)
            .voiceId(voiceName)
            .build();

        VoiceMessageContent content = VoiceMessageContent.builder()
            .ssmlMessage(ssmlMessageType)
            .build();

        SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
            .destinationPhoneNumber(destinationNumber)
            .originationPhoneNumber(originationNumber)
            .content(content)
            .build();

        client.sendVoiceMessage(voiceMessageRequest);
        System.out.println("The message was sent successfully.");

    } catch (PinpointSmsVoiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

Per l'esempio completo dell'SDK, consulta [SendVoiceMessage.java](#) su [GitHub](#).

## JavaScript (Node.js)

Utilizza questo esempio per l'invio di un messaggio vocale tramite AWS SDK per JavaScript in Node.js. In questo esempio si presume che SDK per JavaScript in Node.js sia già stato installato e configurato.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni consulta [Impostazione delle credenziali](#) nella Guida per gli sviluppatori di AWS SDK per JavaScript in Node.js.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best results,
// you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
```

```
// The message can't contain any line breaks.
var ssmLMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointSMSVoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmLMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointSMSVoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
```

```
    if (err) {
      console.log(err.message);
      // Otherwise, show the unique ID for the message.
    } else {
      console.log("Message sent! Message ID: " + data["MessageId"]);
    }
  });
```

## Python

Utilizza questo esempio per l'invio di un messaggio vocale tramite l'AWS SDK for Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta [Credentials](#) nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
```

```

        Pinpoint account and be in E.164 format.
:param caller_id: The phone number that you want to appear on the recipient's
                  device. The phone number must be associated with your Amazon
                  Pinpoint account and be in E.164 format.
:param destination_number: The recipient's phone number. Specify the phone
                           number in E.164 format.
:param language_code: The language to use when sending the message.
:param voice_id: The Amazon Polly voice that you want to use to send the
message.
:param ssml_message: The content of the message. This example uses SSML to
control
                    certain aspects of the message, such as the volume and the
                    speech rate. The message must not contain line breaks.
:return: The ID of the message.
"""
try:
    response = sms_voice_client.send_voice_message(
        DestinationPhoneNumber=destination_number,
        OriginationPhoneNumber=origination_number,
        CallerId=caller_id,
        Content={
            "SSMLMessage": {
                "LanguageCode": language_code,
                "VoiceId": voice_id,
                "Text": ssml_message,
            }
        },
    )
except ClientError:
    logger.exception(
        "Couldn't send message from %s to %s.",
        origination_number,
        destination_number,
    )
    raise
else:
    return response["MessageId"]

def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"

```

```
voice_id = "Matthew"
ssml_message = (
    "<speak>"
    "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
    "using the <break strength='weak'>AWS SDK for Python (Boto3). "
    "<amazon:effect phonation='soft'>Thank you for listening."
    "</amazon:effect>"
    "</speak>"
)
print(f"Sending voice message from {origination_number} to
{destination_number}.")
message_id = send_voice_message(
    boto3.client("pinpoint-sms-voice"),
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
)
print(f"Message sent!\nMessage ID: {message_id}")

if __name__ == "__main__":
    main()
```

## Invio di notifiche push

L'API Amazon Pinpoint può inviare notifiche push transazionali a identificatori di dispositivo specifici. Questa sezione contiene esempi di codice completo che puoi usare per inviare notifiche push tramite l'API Amazon Pinpoint utilizzando AWS SDK.

Puoi utilizzare questi esempi per inviare notifiche push tramite qualsiasi servizio di notifica push supportato da Amazon Pinpoint. Al momento, Amazon Pinpoint supporta i seguenti canali: Firebase Cloud Messaging (FCM), il servizio di notifiche push Apple (APN), Baidu Cloud Push e Amazon Device Messaging (ADM).

### Note

Quando invii notifiche push tramite il servizio Firebase Cloud Messaging (FCM), utilizza il nome del servizio GCM nella chiamata all'API Amazon Pinpoint. Il servizio Google Cloud

Messaging (GCM) è stato interrotto da Google il 10 aprile 2018. Tuttavia, l'API Amazon Pinpoint utilizza il nome del servizio GCM per i messaggi che invia tramite il servizio FCM per mantenere la compatibilità con il codice API scritto prima dell'interruzione del servizio GCM.

## JavaScript (Node.js)

Utilizza questo esempio per inviare notifiche push utilizzando AWS SDK per JavaScript in Node.js. In questo esempio si presume che SDK per JavaScript in Node.js sia già stato installato e configurato.

Questo esempio presuppone anche che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni consulta [Impostazione delle credenziali](#) nella Guida per gli sviluppatori di AWS SDK per JavaScript in Node.js.

```
'use strict';

const AWS = require('aws-sdk');

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const region = 'us-east-1';

// The title that appears at the top of the push notification.
var title = 'Test message sent from Amazon Pinpoint.';

// The content of the push notification.
var message = 'This is a sample message sent from Amazon Pinpoint by using the '
    + 'AWS SDK for JavaScript in Node.js';

// The Amazon Pinpoint project ID that you want to use when you send this
// message. Make sure that the push channel is enabled for the project that
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

// An object that contains the unique token of the device that you want to send
// the message to, and the push service that you want to use to send the message.
var recipient = {
    'token': 'a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0',
    'service': 'GCM'
```

```
};

// The action that should occur when the recipient taps the message. Possible
// values are OPEN_APP (opens the app or brings it to the foreground),
// DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
// specific URL in the device's web browser.)
var action = 'URL';

// This value is only required if you use the URL action. This variable contains
// the URL that opens in the recipient's web browser.
var url = 'https://www.example.com';

// The priority of the push notification. If the value is 'normal', then the
// delivery of the message is optimized for battery usage on the recipient's
// device, and could be delayed. If the value is 'high', then the notification is
// sent immediately, and might wake a sleeping device.
var priority = 'normal';

// The amount of time, in seconds, that the push notification service provider
// (such as FCM or APNS) should attempt to deliver the message before dropping
// it. Not all providers allow you specify a TTL value.
var ttl = 30;

// Boolean that specifies whether the notification is sent as a silent
// notification (a notification that doesn't display on the recipient's device).
var silent = false;

function CreateMessageRequest() {
  var token = recipient['token'];
  var service = recipient['service'];
  if (service == 'GCM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'GCM'
        }
      },
      'MessageConfiguration': {
        'GCMMessage': {
          'Action': action,
          'Body': message,
          'Priority': priority,
          'SilentPush': silent,
          'Title': title,

```

```
        'TimeToLive': ttl,
        'Url': url
    }
}
};
} else if (service == 'APNS') {
var messageRequest = {
    'Addresses': {
        [token]: {
            'ChannelType' : 'APNS'
        }
    },
    'MessageConfiguration': {
        'APNSMessage': {
            'Action': action,
            'Body': message,
            'Priority': priority,
            'SilentPush': silent,
            'Title': title,
            'TimeToLive': ttl,
            'Url': url
        }
    }
};
} else if (service == 'BAIDU') {
var messageRequest = {
    'Addresses': {
        [token]: {
            'ChannelType' : 'BAIDU'
        }
    },
    'MessageConfiguration': {
        'BaiduMessage': {
            'Action': action,
            'Body': message,
            'SilentPush': silent,
            'Title': title,
            'TimeToLive': ttl,
            'Url': url
        }
    }
};
} else if (service == 'ADM') {
var messageRequest = {
```

```
'Addresses': {
  [token]: {
    'ChannelType' : 'ADM'
  }
},
'MessageConfiguration': {
  'ADMMessage': {
    'Action': action,
    'Body': message,
    'SilentPush': silent,
    'Title': title,
    'Url': url
  }
}
};
}

return messageRequest
}

function ShowOutput(data){
  if (data["MessageResponse"]["Result"][recipient["token"]]["DeliveryStatus"]
    == "SUCCESSFUL") {
    var status = "Message sent! Response information: ";
  } else {
    var status = "The message wasn't sent. Response information: ";
  }
  console.log(status);
  console.dir(data, { depth: null });
}

function SendMessage() {
  var token = recipient['token'];
  var service = recipient['service'];
  var messageRequest = CreateMessageRequest();

  // Specify that you're using a shared credentials file, and specify the
  // IAM profile to use.
  var credentials = new AWS.SharedIniFileCredentials({ profile: 'default' });
  AWS.config.credentials = credentials;

  // Specify the AWS Region to use.
  AWS.config.update({ region: region });
}
```

```
//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();
var params = {
    "ApplicationId": applicationId,
    "MessageRequest": messageRequest
};

// Try to send the message.
pinpoint.sendMessage(params, function(err, data) {
    if (err) console.log(err);
    else     ShowOutput(data);
});
}

SendMessage()
```

## Python

Utilizza questo esempio per inviare notifiche push utilizzando AWS SDK for Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato.

Questo esempio presuppone anche che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta [Credentials](#) nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

```
import json
import boto3
from botocore.exceptions import ClientError

# The AWS Region that you want to use to send the message. For a list of
# AWS Regions where the Amazon Pinpoint API is available, see
# https://docs.aws.amazon.com/pinpoint/latest/apireference/
region = "us-east-1"

# The title that appears at the top of the push notification.
title = "Test message sent from Amazon Pinpoint."

# The content of the push notification.
message = ("This is a sample message sent from Amazon Pinpoint by using the "
           "AWS SDK for Python (Boto3).")
```

```
# The Amazon Pinpoint project/application ID to use when you send this message.
# Make sure that the push channel is enabled for the project or application
# that you choose.
application_id = "ce796be37f32f178af652b26eexample"

# A dictionary that contains the unique token of the device that you want to send
# the
# message to, and the push service that you want to use to send the message.
recipient = {
    "token": "a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0",
    "service": "GCM"
}

# The action that should occur when the recipient taps the message. Possible
# values are OPEN_APP (opens the app or brings it to the foreground),
# DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
# specific URL in the device's web browser.)
action = "URL"

# This value is only required if you use the URL action. This variable contains
# the URL that opens in the recipient's web browser.
url = "https://www.example.com"

# The priority of the push notification. If the value is 'normal', then the
# delivery of the message is optimized for battery usage on the recipient's
# device, and could be delayed. If the value is 'high', then the notification is
# sent immediately, and might wake a sleeping device.
priority = "normal"

# The amount of time, in seconds, that the push notification service provider
# (such as FCM or APNS) should attempt to deliver the message before dropping
# it. Not all providers allow you specify a TTL value.
ttl = 30

# Boolean that specifies whether the notification is sent as a silent
# notification (a notification that doesn't display on the recipient's device).
silent = False

# Set the MessageType based on the values in the recipient variable.
def create_message_request():

    token = recipient["token"]
    service = recipient["service"]
```

```
if service == "GCM":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'GCM'
            }
        },
        'MessageConfiguration': {
            'GCMMessage': {
                'Action': action,
                'Body': message,
                'Priority' : priority,
                'SilentPush': silent,
                'Title': title,
                'TimeToLive': ttl,
                'Url': url
            }
        }
    }
elif service == "APNS":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'APNS'
            }
        },
        'MessageConfiguration': {
            'APNSMessage': {
                'Action': action,
                'Body': message,
                'Priority' : priority,
                'SilentPush': silent,
                'Title': title,
                'TimeToLive': ttl,
                'Url': url
            }
        }
    }
elif service == "BAIDU":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'BAIDU'
            }
        }
    }
```

```

    },
    'MessageConfiguration': {
        'BaiduMessage': {
            'Action': action,
            'Body': message,
            'SilentPush': silent,
            'Title': title,
            'TimeToLive': ttl,
            'Url': url
        }
    }
}
elif service == "ADM":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'ADM'
            }
        },
        'MessageConfiguration': {
            'ADMMMessage': {
                'Action': action,
                'Body': message,
                'SilentPush': silent,
                'Title': title,
                'Url': url
            }
        }
    }
else:
    message_request = None

return message_request

# Show a success or failure message, and provide the response from the API.
def show_output(response):
    if response['MessageResponse']['Result']['recipient["token"]']['DeliveryStatus']
    == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))

# Send the message through the appropriate channel.

```

```
def send_message():

    token = recipient["token"]
    service = recipient["service"]
    message_request = create_message_request()

    client = boto3.client('pinpoint', region_name=region)

    try:
        response = client.send_messages(
            ApplicationId=application_id,
            MessageRequest=message_request
        )
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        show_output(response)

send_message()
```

# Creazione di canali personalizzati in Amazon Pinpoint

Amazon Pinpoint include il supporto integrato per l'invio di messaggi tramite notifiche push, e-mail, SMS e canali vocali. È inoltre possibile configurare Amazon Pinpoint per l'invio di messaggi tramite altri canali creando canali personalizzati. I canali personalizzati in Amazon Pinpoint consentono di inviare messaggi tramite qualsiasi servizio dotato di un'API, inclusi i servizi di terze parti. È possibile interagire con le API utilizzando un webhook o richiamando una funzione AWS Lambda.

I segmenti a cui si inviano campagne del canale personalizzato possono contenere endpoint di tutti i tipi, ovvero endpoint in cui il valore dell'attributo `ChannelType` è `EMAIL`, `VOICE`, `SMS`, `CUSTOM` o uno dei vari tipi di endpoint delle notifiche push.

## Creazione di una campagna che invia messaggi attraverso un canale personalizzato

Per assegnare una funzione Lambda o un webhook a una campagna specifica, usa l'API Amazon Pinpoint per creare o aggiornare un oggetto [Campaign](#).

L'oggetto `MessageConfiguration` della campagna deve contenere anche un oggetto `CustomMessage`. Questo oggetto ha un membro, `Data`. Il valore di `Data` è una stringa JSON che contiene il payload del messaggio che si desidera inviare al canale personalizzato.

La campagna deve contenere un oggetto `CustomDeliveryConfiguration`. All'interno dell'oggetto `CustomDeliveryConfiguration` specificare quanto segue:

- `EndpointTypes`: array contenente tutti i tipi di endpoint a cui deve essere inviata la campagna personalizzata mediante il canale personalizzato. Può contenere uno o tutti i seguenti tipi di canale:
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`
  - `CUSTOM`
  - `EMAIL`

- GCM
- SMS
- VOICE
- `DeliveryUri`: destinazione a cui vengono inviati gli endpoint. Puoi specificare uno dei seguenti:
  - Il nome della risorsa Amazon (ARN) di una funzione Lambda che desideri eseguire durante l'esecuzione della campagna.
  - L'URL del webhook a cui inviare i dati dell'endpoint durante l'esecuzione della campagna.

### Note

L'oggetto Campaign può anche contenere un oggetto Hook. Questo oggetto viene utilizzato solo per creare segmenti personalizzati da una funzione Lambda durante l'esecuzione di una campagna. Per ulteriori informazioni, consulta [Personalizzazione dei segmenti con AWS Lambda](#).

## Informazioni sui dati degli eventi inviati da Amazon Pinpoint ai canali personalizzati

Prima di creare una funzione Lambda per l'invio di messaggi su un canale personalizzato, è necessario acquisire familiarità con i dati trasmessi da Amazon Pinpoint. Quando una campagna Amazon Pinpoint invia messaggi su un canale personalizzato, invia un payload alla funzione Lambda di destinazione come nell'esempio seguente:

```
{
  "Message": {},
  "Data": "The payload that's provided in the CustomMessage object in
MessageConfiguration",
  "ApplicationId": "3a9b1f4e6c764ba7b031e7183example",
  "CampaignId": "13978104ce5d6017c72552257example",
  "TreatmentId": "0",
  "ActivityId": "575cb1929d5ba43e87e2478eeexample",
  "ScheduledTime": "2020-04-08T19:00:16.843Z",
  "Endpoints": {
    "1dbcd396df28ac6cf8c1c2b7fexample": {
      "ChannelType": "EMAIL",
      "Address": "mary.major@example.com",
```

```
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "Location":{
      "City":"Seattle",
      "Country":"USA"
    },
    "Demographic":{
      "Make":"OnePlus",
      "Platform":"android"
    },
    "EffectiveDate":"2020-04-01T01:05:17.267Z",
    "Attributes":{
      "CohortId":[
        "42"
      ]
    },
    "CreationDate":"2020-04-01T01:05:17.267Z"
  }
}
```

I dati sugli eventi forniscono gli attributi seguenti:

- **ApplicationId**: ID del progetto Amazon Pinpoint a cui appartiene la campagna.
- **CampaignId**: ID della campagna Amazon Pinpoint che ha richiamato la funzione Lambda.
- **TreatmentId**: ID della variante della campagna. Se hai creato una campagna standard, questo valore è sempre 0. Se è stata creata una campagna di test A/B, questo valore è un numero intero compreso tra 0 e 4.
- **ActivityId**: ID dell'attività eseguita dalla campagna.
- **ScheduledTime**: ora in cui Amazon Pinpoint ha eseguito la campagna, espressa in formato ISO 8601.
- **Endpoints**: elenco degli endpoint interessati dalla campagna. Ogni payload può contenere fino a 50 endpoint. Se il segmento a cui è stata inviata la campagna contiene più di 50 endpoint, Amazon Pinpoint richiama più volte la funzione, con un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti gli endpoint.

È possibile utilizzare questi dati di esempio durante la creazione e il test della funzione Lambda del canale personalizzato.

## Configurazione dei webhook

Se si utilizza un webhook per inviare messaggi di canale personalizzato, l'URL del webhook deve iniziare con "https://". L'URL del webhook può contenere solo caratteri alfanumerici, oltre ai seguenti simboli: trattino (-), punto (.), carattere di sottolineatura (\_), tilde (~), punto interrogativo (?), barra o segno di frazione (/), cancelletto o hash (#) e punto e virgola (:). L'URL deve essere conforme a [RFC3986](#).

Quando si crea una campagna che specifica un URL webhook, Amazon Pinpoint invia un metodo HTTP HEAD a tale URL. La risposta alla richiesta HEAD deve contenere un'intestazione chiamata X-Amz-Pinpoint-AccountId. Il valore di questa intestazione deve essere uguale all'ID dell'account AWS.

## Configurazione delle funzioni Lambda

Questa sezione fornisce una panoramica delle fasi che è necessario eseguire quando si crea una funzione Lambda che invia messaggi su un canale personalizzato. Innanzitutto, creare la funzione. Successivamente, aggiungere una policy di esecuzione alla funzione. Questa policy consente ad Amazon Pinpoint di eseguire la policy durante l'esecuzione di una campagna.

Per un'introduzione alla creazione di funzioni Lambda, consulta l'argomento relativo alla [creazione di funzioni Lambda](#) nella Guida per gli sviluppatori di AWS Lambda.

## Funzione Lambda di esempio

L'esempio di codice seguente elabora il payload e registra il numero di endpoint di ciascun tipo in CloudWatch.

```
import boto3
import random
import pprint
import json
import time

cloudwatch = boto3.client('cloudwatch')

def lambda_handler(event, context):
    customEndpoints = 0
    smsEndpoints = 0
    pushEndpoints = 0
```

```
emailEndpoints = 0
voiceEndpoints = 0
numEndpoints = len(event['Endpoints'])

print("Payload:\n", event)
print("Endpoints in payload: " + str(numEndpoints))

for key in event['Endpoints'].keys():
    if event['Endpoints'][key]['ChannelType'] == "CUSTOM":
        customEndpoints += 1
    elif event['Endpoints'][key]['ChannelType'] == "SMS":
        smsEndpoints += 1
    elif event['Endpoints'][key]['ChannelType'] == "EMAIL":
        emailEndpoints += 1
    elif event['Endpoints'][key]['ChannelType'] == "VOICE":
        voiceEndpoints += 1
    else:
        pushEndpoints += 1

response = cloudwatch.put_metric_data(
    MetricData = [
        {
            'MetricName': 'EndpointCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': len(event['Endpoints'])
        },
        {
            'MetricName': 'CustomCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
            ],
        }
    ]
)
```

```
        'Name': 'ApplicationId',
        'Value': event['ApplicationId']
    }
],
'Unit': 'None',
'Value': customEndpoints
},
{
    'MetricName': 'SMSCount',
    'Dimensions': [
        {
            'Name': 'CampaignId',
            'Value': event['CampaignId']
        },
        {
            'Name': 'ApplicationId',
            'Value': event['ApplicationId']
        }
    ],
    'Unit': 'None',
    'Value': smsEndpoints
},
{
    'MetricName': 'EmailCount',
    'Dimensions': [
        {
            'Name': 'CampaignId',
            'Value': event['CampaignId']
        },
        {
            'Name': 'ApplicationId',
            'Value': event['ApplicationId']
        }
    ],
    'Unit': 'None',
    'Value': emailEndpoints
},
{
    'MetricName': 'VoiceCount',
    'Dimensions': [
        {
            'Name': 'CampaignId',
            'Value': event['CampaignId']
        },
```

```
        {
            'Name': 'ApplicationId',
            'Value': event['ApplicationId']
        }
    ],
    'Unit': 'None',
    'Value': voiceEndpoints
},
{
    'MetricName': 'PushCount',
    'Dimensions': [
        {
            'Name': 'CampaignId',
            'Value': event['CampaignId']
        },
        {
            'Name': 'ApplicationId',
            'Value': event['ApplicationId']
        }
    ],
    'Unit': 'None',
    'Value': pushEndpoints
},
{
    'MetricName': 'EndpointCount',
    'Dimensions': [
    ],
    'Unit': 'None',
    'Value': len(event['Endpoints'])
},
{
    'MetricName': 'CustomCount',
    'Dimensions': [
    ],
    'Unit': 'None',
    'Value': customEndpoints
},
{
    'MetricName': 'SMSCount',
    'Dimensions': [
    ],
    'Unit': 'None',
    'Value': smsEndpoints
},
```

```
    {
      'MetricName': 'EmailCount',
      'Dimensions': [
      ],
      'Unit': 'None',
      'Value': emailEndpoints
    },
    {
      'MetricName': 'VoiceCount',
      'Dimensions': [
      ],
      'Unit': 'None',
      'Value': voiceEndpoints
    },
    {
      'MetricName': 'PushCount',
      'Dimensions': [
      ],
      'Unit': 'None',
      'Value': pushEndpoints
    }
  ],
  Namespace = 'PinpointCustomChannelExecution'
)
print("cloudwatchResponse:\n", response)
```

Quando una campagna Amazon Pinpoint esegue questa funzione Lambda, Amazon Pinpoint invia alla funzione un elenco di membri del segmento. La funzione conta il numero di endpoint di ogni ChannelType. Invia quindi i dati ad Amazon CloudWatch. È possibile visualizzare queste metriche nella sezione Parametri della console CloudWatch. I parametri sono disponibili nello spazio dei nomi PinpointCustomChannelExecution.

È possibile modificare questo esempio di codice in modo che si connetta anche all'API di un servizio esterno per inviare messaggi tramite tale servizio.

## Formato di risposta della funzione Lambda per Amazon Pinpoint

Se desideri utilizzare la multivariata del percorso o la suddivisione sì/no per determinare il percorso dell'endpoint dopo un'attività del canale personalizzato, devi strutturare la risposta della funzione Lambda in un formato comprensibile da Amazon Pinpoint e quindi inviare gli endpoint lungo il percorso corretto.

La struttura della risposta deve avere il formato seguente:

```
{
  <Endpoint ID 1>:{
    EventAttributes: {
      <Key1>: <Value1>,
      <Key2>: <Value2>,
      ...
    }
  },
  <Endpoint ID 2>:{
    EventAttributes: {
      <Key1>: <Value1>,
      <Key2>: <Value2>,
      ...
    }
  },
  ...
}
```

Ciò ti consentirà quindi di selezionare una chiave e un valore per determinare il percorso degli endpoint.

Yes/no split [Info](#) ✕

Select a condition type  
Event

Choose a journey message activity and event  
lambdaFunction (Custom)  
Call to function or webhook response

Response [Info](#)

Attribute	Value
<input type="text"/>	<input type="text"/>

[+ Add condition](#)

Condition evaluation  
The amount of time that Amazon Pinpoint waits before it evaluates the conditions.

Evaluate after

1 hours

Description - optional  
*Add description*

[Save](#)

## Concessione dell'autorizzazione Amazon Pinpoint a richiamare la funzione Lambda

È possibile usare l'AWS Command Line Interface (AWS CLI) per aggiungere autorizzazioni alla policy di funzione Lambda assegnata alla funzione Lambda stessa. Per consentire ad Amazon Pinpoint di richiamare una funzione, usa il comando [add-permission](#), come mostrato nell'esempio seguente:

```
aws lambda add-permission \
--function-name myFunction \
--statement-id sid0 \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:111122223333:apps/* \
--source-account 111122223333
```

Nel comando precedente, procedi come segue.

- Sostituisci *myFunction* con il nome della funzione Lambda.
- Sostituisci *us-east-1* con la regione AWS in cui viene utilizzato Amazon Pinpoint.
- Sostituisci *111122223333* con l'ID del tuo account AWS.

Quando esegui il comando `add-permission`, la funzione Lambda restituisce l'output seguente:

```
{
  "Statement": "{ \"Sid\": \"sid\",
    \"Effect\": \"Allow\",
    \"Principal\": { \"Service\": \"pinpoint.us-east-1.amazonaws.com\" },
    \"Action\": \"lambda:InvokeFunction\",
    \"Resource\": \"arn:aws:lambda:us-east-1:111122223333:function:myFunction\",
    \"Condition\":
      { \"ArnLike\":
        { \"AWS:SourceArn\":
          \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/*\" } },
      { \"StringEquals\":
        { \"AWS:SourceAccount\":
          \"111122223333\" } } } }
}
```

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy di funzione Lambda.

## Ulteriore limitazione della policy di esecuzione

È possibile modificare la policy di esecuzione limitandola a un progetto Amazon Pinpoint specifico. A tale scopo, sostituire `*` nell'esempio precedente con l'ID univoco del progetto. È possibile limitare ulteriormente la policy limitandola a una campagna specifica. Ad esempio, per limitare la policy in modo che il consenta solo una campagna con l'ID `95fee4cd1d7f5cd67987c1436example` in un progetto con l'ID `dbaf6ec2226f0a9a8615e3ea5example`, utilizzare il seguente valore per l'attributo `source-arn`:

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/campaigns/95fee4cd1d7f5cd67987c1436example
```

 Note

Se si limita l'esecuzione della funzione Lambda a una campagna specifica, è necessario innanzitutto creare la funzione con una policy meno restrittiva. Successivamente, è necessario creare la campagna in Amazon Pinpoint e scegliere la funzione. Infine, è necessario aggiornare la policy di esecuzione per fare riferimento alla campagna specificata.

# Streaming di eventi Amazon Pinpoint su Kinesis

In Amazon Pinpoint, un evento è un'azione che si verifica quando un utente interagisce con una delle tue applicazioni, quando invii un messaggio da una campagna o un percorso oppure quando invii un messaggio transazionale SMS o e-mail. Ad esempio, se invii un messaggio e-mail, si verificano diversi eventi:

- Quando invii il messaggio, si verifica un evento di invio .
- Quando il messaggio raggiunge la casella di posta in arrivo del destinatario, si verifica un evento recapitato .
- Quando il destinatario apre il messaggio, si verifica un evento di apertura .

È possibile configurare Amazon Pinpoint in modo che invii informazioni sugli eventi ad Amazon Kinesis. La piattaforma Kinesis offre servizi che puoi utilizzare per raccogliere, elaborare e analizzare i dati dei AWS servizi in tempo reale. Amazon Pinpoint può inviare dati sugli eventi a Firehose, che li trasmette ad archivi AWS dati come Amazon S3 o Amazon Redshift. Amazon Pinpoint può anche trasmettere dati al flusso di dati Kinesis, che acquisisce e archivia più flussi di dati per l'elaborazione da parte delle applicazioni di analisi.

Il flusso di eventi Amazon Pinpoint include informazioni sulle interazioni degli utenti con le applicazioni (app) connesse ad Amazon Pinpoint. Include anche informazioni su tutti i messaggi inviati dalle campagne, attraverso qualsiasi canale e dai viaggi. Ciò può includere anche tutti gli eventi personalizzati precedentemente definiti. Infine, include informazioni sulle tutte le e-mail transazionali e su tutti i messaggi SMS inviati.

## Note

Amazon Pinpoint non esegue lo streaming di informazioni sulle notifiche push transazionali o sui messaggi vocali.

In questo capitolo vengono fornite informazioni sulla configurazione di Amazon Pinpoint per lo streaming dei dati degli eventi in Kinesis. Contiene anche esempi di dati di eventi trasmessi in streaming da Amazon Pinpoint.

## Argomenti

- [Configurazione dello streaming degli eventi](#)
- [Eventi app](#)
- [Eventi di campagna](#)
- [Eventi di viaggio](#)
- [Eventi e-mail](#)
- [Eventi SMS](#)

## Configurazione dello streaming degli eventi

Puoi configurare Amazon Pinpoint per inviare i dati degli eventi a uno stream Amazon Kinesis o a un flusso di distribuzione Amazon Data Firehose. Amazon Pinpoint può inviare dati sugli eventi per campagne, percorsi e messaggi e-mail e SMS transazionali.

Questa sezione include informazioni sulla configurazione dello streaming di eventi a livello di programmazione. Puoi anche utilizzare la console Amazon Pinpoint per configurare lo streaming di eventi. Per informazioni sulla configurazione dello streaming di eventi utilizzando la console Amazon Pinpoint, consulta [Impostazioni del flusso di eventi](#) nella Guida per l'utente di Amazon Pinpoint.

### Prerequisiti

Gli esempi in questa sezione richiedono il seguente input:

- L'ID di un'applicazione integrata con Amazon Pinpoint e che segnala gli eventi. Per informazioni sull'integrazione, consulta [Integrazione di Amazon Pinpoint con le applicazioni](#).
- L'Amazon Resource Name (ARN) di uno stream Kinesis o di un flusso di distribuzione Firehose nel tuo account. AWS Per informazioni sulla creazione di queste risorse, consulta [Creating and Managing Streams](#) nella Amazon Kinesis Data Streams Developer Guide o [Creazione di un flusso di distribuzione Amazon Data Firehose nella Amazon Data Firehose Developer Guide](#).
- L'ARN di un ruolo AWS Identity and Access Management (IAM) che autorizza Amazon Pinpoint a inviare dati allo stream. Per informazioni su come creare un ruolo, consulta [Ruolo IAM per lo streaming di eventi su Kinesis](#).

### AWS CLI

L' AWS CLI esempio seguente utilizza il comando. [put-event-stream](#) Questo comando configura Amazon Pinpoint per l'invio di eventi a un flusso Kinesis:

```
aws pinpoint put-event-stream \  
--application-id projectId \  
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

## AWS SDK for Java

L'esempio Java seguente configura Amazon Pinpoint per l'invio di eventi a un flusso Kinesis:

```
public PutEventStreamResult createEventStream(AmazonPinpoint pinClient,  
    String appId, String streamArn, String roleArn) {  
  
    WriteEventStream stream = new WriteEventStream()  
        .withDestinationStreamArn(streamArn)  
        .withRoleArn(roleArn);  
  
    PutEventStreamRequest request = new PutEventStreamRequest()  
        .withApplicationId(appId)  
        .withWriteEventStream(stream);  
  
    return pinClient.putEventStream(request);  
}
```

Questo esempio crea un oggetto [WriteEventStream](#) che archivia i nomi delle risorse Amazon (ARN) del flusso Kinesis e del ruolo IAM. L'oggetto `WriteEventStream` viene passato a un oggetto [PutEventStreamRequest](#) per configurare Amazon Pinpoint per lo streaming degli eventi per un'applicazione specifica. L'oggetto `PutEventStreamRequest` viene passato al metodo [putEventStream](#) del client Amazon Pinpoint.

Puoi assegnare un flusso Kinesis a più applicazioni. In questo caso, Amazon Pinpoint invia i dati degli eventi con codifica Base64 da ogni applicazione al flusso e ciò consente di analizzare i dati come una raccolta. Il seguente metodo di esempio accetta un elenco di ID app (applicazione) e utilizza il metodo dell'esempio precedente, `createEventStream`, per assegnare un flusso a ciascuna applicazione:

```
public List<PutEventStreamResult> createEventStreamFromAppList(  
    AmazonPinpoint pinClient, List<String> appIDs,  
    String streamArn, String roleArn) {  
  
    return appIDs.stream()  
        .map(appId -> createEventStream(pinClient, appId, streamArn,  
            roleArn))
```

```
        .collect(Collectors.toList());  
    }
```

Sebbene sia possibile assegnare un flusso a più applicazioni, non è possibile assegnare più flussi a una sola applicazione.

## Disabilitazione dello streaming di eventi

Se assegni un flusso Kinesis a un'applicazione, puoi disabilitare lo streaming di eventi per quell'applicazione. Amazon Pinpoint interrompe lo streaming degli eventi su Kinesis, ma puoi visualizzare l'analisi degli eventi utilizzando la console Amazon Pinpoint.

### AWS CLI

Utilizza il comando [delete-event-stream](#):

```
aws pinpoint delete-event-stream --application-id application-id
```

### AWS SDK for Java

Utilizza il [deleteEventStream](#) metodo del client Amazon Pinpoint:

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

## Eventi app

Dopo aver integrato l'applicazione (app) con Amazon Pinpoint, Amazon Pinpoint può eseguire lo streaming dei dati degli eventi relativi all'attività degli utenti e ai recapiti dei messaggi per l'app.

### Esempio

L'oggetto JSON per un evento app contiene i dati mostrati nell'esempio seguente.

```
{  
  "event_type": "_session.stop",  
  "event_timestamp": 1487973802507,  
  "arrival_timestamp": 1487973803515,  
  "event_version": "3.0",  
  "application": {  
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",  
  }  
}
```

```

"cognito_identity_pool_id": "us-east-1:a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6",
"package_name": "main.page",
"sdk": {
  "name": "aws-sdk-mobile-analytics-js",
  "version": "0.9.1:2.4.8"
},
"title": "title",
"version_name": "1.0",
"version_code": "1"
},
"client": {
  "client_id": "m3n4o5p6-a1b2-c3d4-e5f6-g7h8i9j0k1l2",
  "cognito_id": "us-east-1:i9j0k1l2-m3n4-o5p6-a1b2-c3d4e5f6g7h8"
},
"device": {
  "locale": {
    "code": "en_US",
    "country": "US",
    "language": "en"
  },
  "make": "generic web browser",
  "model": "Unknown",
  "platform": {
    "name": "android",
    "version": "10.10"
  }
},
"session": {
  "session_id": "f549dea9-1090-945d-c3d1-e4967example",
  "start_timestamp": 1487973202531,
  "stop_timestamp": 1487973802507
},
"attributes": {},
"metrics": {}
}

```

## Attributi di eventi dell'app

Questa sezione definisce gli attributi inclusi nel flusso di eventi dell'app.

Attributo	Descrizione
event_type	Tipo di evento. I valori possibili sono:

Attributo	Descrizione
	<ul style="list-style-type: none"><li>• <code>_session.start</code>: l'endpoint ha iniziato una nuova sessione.</li><li>• <code>_session.stop</code>: l'endpoint ha terminato una sessione.</li><li>• <code>_userauth.sign_in</code>: l'endpoint ha stabilito la connessione all'app.</li><li>• <code>_userauth.sign_up</code>: un nuovo endpoint ha completato la procedura di registrazione nell'app.</li><li>• <code>_userauth.auth_fail</code>: l'endpoint ha tentato di accedere all'app, ma non è stato in grado di completare il processo.</li><li>• <code>_monetization.purchase</code>: l'endpoint ha effettuato un acquisto nell'app.</li><li>• <code>_session.pause</code>: la sessione dell'endpoint è stata sospesa. Le sessioni in pausa possono essere riavviate in modo da poter continuar e a raccogliere i parametri senza avviare una sessione completamente nuova.</li><li>• <code>_session.resume</code>: la sessione dell'endpoint è stata ripresa.</li></ul>
<code>event_timestamp</code>	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
<code>arrival_timestamp</code>	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.

Attributo	Descrizione
<code>event_version</code>	<p>La versione dello schema JSON dell'evento.</p> <div><p> <b>Tip</b></p><p>Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.</p></div>
<code>application</code>	<p>Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Applicazione</a>.</p>
<code>client</code>	<p>Informazioni sull'endpoint che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Client</a>.</p>
<code>device</code>	<p>Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Dispositivo</a>.</p>
<code>session</code>	<p>Informazioni sulla sessione che ha generato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Sessione</a>.</p>
<code>attributes</code>	<p>Attributi associati all'evento. Per gli eventi segnalati dalle app, questo oggetto include gli attributi personalizzati definiti dall'utente.</p>
<code>metrics</code>	<p>Parametri correlati all'evento. Facoltativamente, puoi configurare le app per l'invio di metriche personalizzate ad Amazon Pinpoint.</p>

## Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
app_id	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
cognito_identity_pool_id	ID del pool di identità Amazon Cognito a cui è associato l'endpoint.
package_name	Il nome del pacchetto dell'app, ad esempio <code>com.example.my_app</code> .
sdk	Informazioni sull'SDK utilizzato per segnalare l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">SDK</a> .
title	Il nome dell'app.
version_name	Il nome della versione dell'app, ad esempio <code>V2.5</code> .
version_code	Il numero di versione dell'app, ad esempio <code>3</code> .

## SDK

Include informazioni sull'SDK utilizzato per segnalare l'evento.

Attributo	Descrizione
name	Il nome dell'SDK utilizzato per segnalare l'evento.
version	La versione dell'SDK.

## Client

Include informazioni sull'endpoint che ha generato l'evento.

Attributo	Descrizione
<code>client_id</code>	ID dell'endpoint.
<code>cognito_id</code>	Token ID Amazon Cognito associato all'endpoint.

## Dispositivo

Include informazioni sul dispositivo dell'endpoint che ha generato l'evento.

Attributo	Descrizione
<code>locale</code>	Informazioni sulle impostazioni di lingua e regione per il dispositivo dell'endpoint. Per ulteriori informazioni, consulta la tabella <a href="#">Locale</a> .
<code>make</code>	Il produttore del dispositivo dell'endpoint.
<code>model</code>	L'identificatore del modello del dispositivo dell'endpoint.
<code>platform</code>	Informazioni sul sistema operativo del dispositivo dell'endpoint. Per ulteriori informazioni, consulta la tabella <a href="#">Piattaforma</a> .

## Locale

Include informazioni sulle impostazioni di lingua e regione per il dispositivo dell'endpoint.

Attributo	Descrizione
<code>code</code>	L'identificatore locale associato al dispositivo.

Attributo	Descrizione
<code>country</code>	Il paese o la regione associata alle impostazioni locali del dispositivo.
<code>language</code>	La lingua associata alle impostazioni locali del dispositivo.

## Piattaforma

Include informazioni sul sistema operativo del dispositivo dell'endpoint.

Attributo	Descrizione
<code>name</code>	Il nome del sistema operativo sul dispositivo.
<code>version</code>	La versione del sistema operativo sul dispositivo.

## Sessione

Include informazioni sulla sessione che ha generato l'evento.

Attributo	Descrizione
<code>session_id</code>	Un ID univoco che identifica la sessione.
<code>start_timestamp</code>	La data e l'ora in cui è iniziata la sessione, espresse in formato orario Unix in millisecondi.
<code>stop_timestamp</code>	La data e l'ora in cui è terminata la sessione, espresse in formato orario Unix in millisecondi.

## Eventi di campagna

Se utilizzi Amazon Pinpoint per inviare campagne attraverso qualsiasi canale, Amazon Pinpoint può eseguire lo streaming dei dati di evento su tali campagne. Questo include i dati di evento relativi ai

messaggi e-mail o SMS che invii da una campagna. Per informazioni dettagliate sui dati inviati in streaming da Amazon Pinpoint per questi tipi di messaggi, consulta [the section called “Eventi e-mail”](#) e [the section called “Eventi SMS”](#).

## Evento di esempio

L'oggetto JSON per un evento di campagna contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l12m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
        #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
        #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

## Attributi degli eventi di campagna

Questa sezione definisce gli attributi inclusi nel flusso di eventi della campagna.

Attributo	Descrizione
event_type	<p>Tipo di evento. I valori possibili sono:</p> <ul style="list-style-type: none"><li>• <code>_campaign.send</code>: Amazon Pinpoint ha eseguito la campagna.</li><li>• <code>_campaign.opened_notification</code>: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha toccato la notifica per aprirla.</li><li>• <code>_campaign.received_foreground</code>: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha ricevuto il messaggio come notifica in primo piano.</li><li>• <code>_campaign.received_background</code>: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha ricevuto il messaggio come notifica in background.</li></ul> <div data-bbox="862 1333 1510 1801" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p><code>_campaign.opened_notification</code>, <code>_campaign.received_foreground</code> e <code>_campaign.received_background</code> vengono restituiti solo se utilizzi AWS Amplify. Per ulteriori informazioni sull'integrazione dell'app con AWS Amplify: Per informazioni, consultare <a href="#">Connessione dell'applicazione front-</a></p></div>

Attributo	Descrizione
	<p><a href="#">end ad Amazon Pinpoint con AWS Amplify.</a></p>
event_timestamp	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
arrival_timestamp	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
event_version	<p>La versione dello schema JSON dell'evento.</p> <div data-bbox="829 751 1507 1115" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Tip</b></p> <p>Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.</p> </div>
application	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Applicazione</a> .
client	Informazioni sull'endpoint a cui è associato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Client</a> .
device	Informazioni sul dispositivo che ha segnalato l'evento. Per i messaggi di campagna e transazionali, questo oggetto è vuoto.
session	Informazioni sulla sessione che ha generato l'evento. Per le campagne, questo oggetto è vuoto.

Attributo	Descrizione
<code>attributes</code>	<p>Attributi associati all'evento. Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi che vengono creati quando si invia una campagna, questo oggetto contiene gli attributi associati alla campagna. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso.</p> <p>Per ulteriori informazioni, consulta la tabella <a href="#">Attributi</a>.</p>
<code>client_context</code>	<p>Contiene un oggetto custom, che contiene una proprietà <code>endpoint</code>. La proprietà <code>endpoint</code> contiene i contenuti del record di endpoint per l'endpoint a cui è stata inviata la campagna.</p>
<code>awsAccountId</code>	<p>L'ID dell'account AWS utilizzato per inviare il messaggio.</p>

## Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
<code>app_id</code>	<p>ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.</p>
<code>sdk</code>	<p>L'SDK utilizzato per segnalare l'evento.</p>

## Attributi

Include informazioni sulla campagna che ha prodotto l'evento.

Attributo	Descrizione
<code>treatment_id</code>	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresenta il numero della descrizione ( <code>treatment</code> ) del messaggio. Per le campagne standard, questo valore è 0.
<code>campaign_activity_id</code>	ID univoco generato da Amazon Pinpoint quando si verifica l'evento.
<code>delivery_type</code>	<p>Metodo di recapito della campagna. Non confondere questo attributo con il campo <code>ChannelType</code> specificato sotto la proprietà <code>endpoint</code> di <code>client_context</code>. Il campo <code>ChannelType</code> è in genere basato sull'endpoint a cui viene inviato il messaggio.</p> <p>Per i canali che supportano un solo tipo di endpoint, i campi <code>delivery_type</code> e <code>ChannelType</code> hanno lo stesso valore. Ad esempio, per il canale e-mail, i campi <code>delivery_type</code> e <code>ChannelType</code> hanno lo stesso valore di EMAIL.</p> <p>Tuttavia, questa condizione non è sempre vera per i canali che supportano diversi tipi di endpoint, come i canali personalizzati. Puoi utilizzare un canale personalizzato per diversi endpoint, come EMAIL, SMS, CUSTOM e così via. In questo caso, <code>delivery_type</code> identifica un evento di recapito personalizzato (CUSTOM) e <code>ChannelType</code> specifica il tipo di endpoint a cui è stata inviata la campagna,</p>

Attributo	Descrizione
	<p>ad esempio EMAIL, SMS, CUSTOM e così via. Per ulteriori informazioni sulla creazione di canali personalizzati, consulta <a href="#">Creazione di canali personalizzati</a>.</p> <p>I valori possibili sono:</p> <ul style="list-style-type: none"><li>• EMAIL</li><li>• SMS</li><li>• ADM</li><li>• APNS</li><li>• APNS_SANDBOX</li><li>• APNS_VOIP</li><li>• APNS_VOIP_SANDBOX</li><li>• VOICE</li><li>• GCM</li><li>• BAIDU</li><li>• PUSH</li><li>• CUSTOM</li></ul>
campaign_id	L'ID univoco della campagna da cui è stato inviato il messaggio.

Attributo	Descrizione
<code>campaign_send_status</code>	<p>Indica lo stato della campagna per l'endpoint di destinazione. I valori possibili includono:</p> <ul style="list-style-type: none"><li>• <b>SUCCESS</b>: invio della campagna all'endpoint riuscito.</li><li>• <b>FAILURE</b>: invio della campagna all'endpoint non riuscito.</li><li>• <b>DAILY_CAP</b>: la campagna non è stata inviata all'endpoint perché all'endpoint è già stato inviato il numero massimo di messaggi giornalieri.</li><li>• <b>EXPIRED</b>: la campagna non è stata inviata all'endpoint perché l'invio ha superato la durata massima o la velocità di invio impostata per la campagna.</li><li>• <b>QUIET_TIME</b>: la campagna non è stata inviata all'endpoint a causa delle limitazioni relative al periodo di silenzio.</li><li>• <b>HOLDOUT</b>: la campagna non è stata inviata all'endpoint perché l'endpoint è un membro del gruppo di esclusione.</li><li>• <b>DUPLICATE_ADDRESS</b>: nel segmento sono presenti indirizzi di endpoint duplicati. La campagna è stata inviata una volta all'indirizzo dell'endpoint.</li><li>• <b>QUIET_TIME</b>: la campagna non è stata inviata all'endpoint a causa delle limitazioni relative al periodo di silenzio.</li><li>• <b>CAMPAIGN_CAP</b>: la campagna non è stata inviata all'endpoint perché all'endpoint è già stato inviato il numero massimo di messaggi dalla campagna in corso.</li></ul>

Attributo	Descrizione
	<ul style="list-style-type: none"><li>• <b>FAILURE_PERMANENT</b>: si è verificato un errore permanente durante l'invio all'endpo int.</li><li>• <b>TRANSIENT_FAILURE</b>: si è verificato un errore temporaneo durante l'invio all'endpo int.</li><li>• <b>THROTTLED</b>: l'invio è stato limitato a livello di larghezza di banda della rete.</li><li>• <b>UNKNOWN</b>: errore sconosciuto.</li><li>• <b>HOOK_FAILURE</b>: hook della campagna non riuscito.</li><li>• <b>CUSTOM_DELIVERY_FAILURE</b>: recapito personalizzato non riuscito.</li><li>• <b>RECOMMENDATION_FAILURE</b>: raccomandazione non riuscita.</li><li>• <b>UNSUPPORTED_CHANNEL</b>: canale non supportato.</li></ul>

## Client

Include informazioni sull'endpoint a cui era destinata la campagna.

Attributo	Descrizione
<code>client_id</code>	L'ID dell'endpoint a cui è stata inviata la campagna.

## Eventi di viaggio

Se si pubblica un percorso, Amazon Pinpoint può eseguire lo streaming dei dati degli eventi relativi al percorso. Ciò include i dati degli eventi relativi a tutti i messaggi e-mail, SMS, push o personalizzati inviati dal percorso.

Per informazioni sui dati trasmessi da Amazon Pinpoint, consulta quanto segue:

- Per i messaggi e-mail, consulta [the section called “Eventi e-mail”](#).
- Per i messaggi SMS, consulta [Eventi SMS](#).

## Evento di esempio

L'oggetto JSON per un evento di viaggio contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_journey.send",
  "event_timestamp": 1572989078843,
  "arrival_timestamp": 1572989078843,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {

    }
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {

    }
  },
  "session": {

  },
  "attributes": {
    "journey_run_id": "edc9a0b577164d1daf72ebd15example",
    "journey_send_status": "SUCCESS",
    "journey_id": "546401670c5547b08811ac6a9example",
    "journey_activity_id": "0yKexample",
    "journey_activity_type": "EMAIL",
    "journey_send_status_message": "200",
    "journey_send_status_code": "200"
  },
  "client_context": {
    "custom": {

    }
  }
}
```

```

    "endpoint": {"ChannelType": "EMAIL", "EndpointStatus": "ACTIVE", "OptOut": "NONE", "Demographic": {"Timezone": "America/Los_Angeles"}}
  }
},
"awsAccountId": "123456789012"
}

```

## Attributi di eventi di viaggio

Questa sezione definisce gli attributi inclusi nei dati del flusso di eventi generati da Amazon Pinpoint per un percorso.

Attributo	Descrizione
event_type	Tipo di evento. Per gli eventi dei percorsi, il valore per questo attributo è sempre <code>_journey.send</code> , che indica che Amazon Pinpoint ha eseguito il percorso.
event_timestamp	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
arrival_timestamp	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
event_version	La versione dello schema JSON dell'evento. <div data-bbox="829 1383 1507 1747" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Tip</b></p> <p>Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.</p> </div>

Attributo	Descrizione
<code>application</code>	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Applicazione</a> .
<code>client</code>	Informazioni sull'endpoint associato all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Client</a> .
<code>device</code>	Informazioni sul dispositivo che ha segnalato l'evento. Per i viaggi, questo oggetto è vuoto.
<code>session</code>	Informazioni sulla sessione che ha generato l'evento. Per i viaggi, questo oggetto è vuoto.
<code>attributes</code>	Attributi associati al viaggio e all'attività di viaggio che ha generato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Attributi</a> .
<code>client_context</code>	Contiene un oggetto custom, che contiene una proprietà <code>endpoint</code> . La proprietà <code>endpoint</code> contiene i contenuti del record di endpoint per l'endpoint associato all'evento.
<code>awsAccountId</code>	L'ID dell' AWS account utilizzato per eseguire il viaggio.

## Applicazione

Include le informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
<code>app_id</code>	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
<code>sdk</code>	L'SDK utilizzato per segnalare l'evento.

## Client

Include informazioni sull'endpoint associato all'evento.

Attributo	Descrizione
<code>client_id</code>	ID dell'endpoint.

## Attributes

Include informazioni sul viaggio che ha generato l'evento.

Attributo	Descrizione
<code>journey_run_id</code>	ID univoco del percorso che ha generato l'evento. Amazon Pinpoint genera e assegna automaticamente questo ID a ogni nuova esecuzione di un percorso.
<code>journey_send_status</code>	Indica lo stato di consegna del messaggio associato all'evento. I valori possibili includono: <ul style="list-style-type: none"><li>• <b>SUCCESS</b>: la campagna è stata inviata correttamente all'endpoint.</li><li>• <b>FAILURE</b>: il messaggio non è stato inviato all'endpoint a causa di un errore.</li><li>• <b>CUSTOM_DELIVERY_FAILURE</b>: recapito personalizzato non riuscito.</li><li>• <b>FAILURE_PERMANENT</b>: si è verificato un errore permanente durante l'invio all'endpoint.</li></ul> <div data-bbox="862 1654 1511 1885"><p> <b>Tip</b></p><p>È possibile filtrare gli eventi con stato <code>FAILURE_PERMANENT</code> e <code>journey_send_status_code</code></p></div>

Attributo	Descrizione
	<p>impostarli su 403 per determinare se vi è una violazione della politica di accesso e del ruolo. Per le campagne vocali in uscita, queste eccezioni sono tipiche dei casi in cui il ruolo di esecuzione della campagna Connect che collega i viaggi di Amazon Pinpoint alle campagne Amazon Connect viene inavvertitamente eliminato per le esecuzioni di viaggio in corso.</p> <ul style="list-style-type: none"> <li>• THROTTLED: l'invio è stato limitato a livello di larghezza di banda della rete.</li> <li>• UNSUPPORTED_CHANNEL: canale non supportato.</li> <li>• DAILY_CAP: il messaggio non è stato inviato all'endpoint in quanto l'invio del messaggio supera il numero massimo di messaggi che il percorso o il progetto può inviare a un singolo endpoint in un intervallo di 24 ore.</li> <li>• QUIET_TIME: il messaggio non è stato inviato a causa delle limitazioni per l'intervallo di silenzio per il percorso o progetto.</li> <li>• QUIET_TIME_MISSING_TIMEZONE: il messaggio non è stato inviato perché la stima del fuso orario non è in grado di stimare un fuso orario per l'endpoint e il periodo di silenzio è abilitato.</li> </ul>
journey_id	L'ID univoco del viaggio che ha generato l'evento.

Attributo	Descrizione
<code>journey_activity_id</code>	L'ID univoco dell'attività di viaggio che ha generato l'evento.
<code>journey_activity_type</code>	Il tipo di attività di percorso dell'evento. Può essere EMAIL, SMS, PUSH, CONTACT_C ENTER o CUSTOM. <div data-bbox="829 527 1508 747"><p> <b>Note</b> VOICE non è un tipo di attività di percorso supportato.</p></div>
<code>journey_send_status_message</code>	La descrizione dello stato dell'evento di invio.
<code>journey_send_status_code</code>	Il codice di stato HTTP della richiesta.

## Eventi e-mail

Quando invii messaggi e-mail, Amazon Pinpoint può eseguire lo streaming dei dati che forniscono informazioni aggiuntive sui seguenti tipi di eventi relativi ai messaggi:

- Invii
- Consegne
- Mancati recapiti
- Reclami
- Aperture
- Clic
- Rifiuti
- Annullamenti di sottoscrizioni
- Errori di rendering

I tipi di evento nell'elenco precedente sono illustrati in dettaglio in [Attributi di eventi e-mail](#).

A seconda dell'API e delle impostazioni utilizzate per inviare messaggi e-mail, è possibile che vengano visualizzati altri tipi di eventi o dati diversi. Ad esempio, se invii messaggi utilizzando set di configurazione che pubblicano i dati degli eventi in Amazon Kinesis, ad esempio quelli forniti da Amazon Simple Email Service (Amazon SES), i dati possono includere anche eventi per errori di rendering dei modelli. Per informazioni su questi dati, consulta [Monitoraggio tramite la pubblicazione degli eventi Amazon SES](#) nella Guida per gli sviluppatori di Amazon Simple Email Service.

## Eventi di esempio

### Invio di e-mail

L'oggetto JSON per un evento di invio e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
```

```
    "name": "From",
    "value": "sender@example.com"
  }, {
    "name": "To",
    "value": "recipient@example.com"
  }, {
    "name": "Subject",
    "value": "Amazon Pinpoint Test"
  }, {
    "name": "MIME-Version",
    "value": "1.0"
  }, {
    "name": "Content-Type",
    "value": "multipart/alternative; boundary=\"-----=_Part_314159_271828\""
  }
],
"common_headers": {
  "from": "sender@example.com",
  "to": ["recipient@example.com"],
  "subject": "Amazon Pinpoint Test"
}
},
"send": {}
}
}
}
```

## Consegna di e-mail

L'oggetto JSON per un evento di consegna e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
```

```
  "platform": {}
},
"session": {},
"attributes": {
  "feedback": "delivered"
},
"awsAccountId": "123456789012",
"facets": {
  "email_channel": {
    "mail_event": {
      "mail": {
        "message_id": "0200000073rnbnmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
        "message_send_timestamp": 1564618621380,
        "from_address": "sender@example.com",
        "destination": ["recipient@example.com"],
        "headers_truncated": false,
        "headers": [{
          "name": "From",
          "value": "sender@example.com"
        }, {
          "name": "To",
          "value": "recipient@example.com"
        }, {
          "name": "Subject",
          "value": "Amazon Pinpoint Test"
        }, {
          "name": "MIME-Version",
          "value": "1.0"
        }, {
          "name": "Content-Type",
          "value": "multipart/alternative; boundary=\"-----=_Part_314159_271828\""
        }
      ],
      "common_headers": {
        "from": "sender@example.com",
        "to": ["recipient@example.com"],
        "subject": "Amazon Pinpoint Test"
      }
    }
  },
  "delivery": {
    "smtp_response": "250 ok: Message 82080542 accepted",
    "reporting_mta": "a8-53.smtp-out.amazonses.com",
    "recipients": ["recipient@example.com"],
    "processing_time_millis": 1310
  }
}
```

```
    }  
  }  
}  
}
```

## Clic su e-mail

L'oggetto JSON per un evento di clic su e-mail contiene i dati mostrati nell'esempio seguente.

```
{  
  "event_type": "_email.click",  
  "event_timestamp": 1564618621380,  
  "arrival_timestamp": 1564618713751,  
  "event_version": "3.1",  
  "application": {  
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",  
    "sdk": {}  
  },  
  "client": {  
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"  
  },  
  "device": {  
    "platform": {}  
  },  
  "session": {},  
  "attributes": {  
    "feedback": "https://aws.amazon.com/pinpoint/"  
  },  
  "awsAccountId": "123456789012",  
  "facets": {  
    "email_channel": {  
      "mail_event": {  
        "mail": {  
          "message_id": "0200000073rn bmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",  
          "message_send_timestamp": 1564618621380,  
          "from_address": "sender@example.com",  
          "destination": ["recipient@example.com"],  
          "headers_truncated": false,  
          "headers": [{  
            "name": "From",  
            "value": "sender@example.com"  
          }, {  
            "name": "To",  
            "value": "recipient@example.com"  
          }  
        ]  
      }  
    }  
  }  
}
```

```

        }, {
          "name": "Subject",
          "value": "Amazon Pinpoint Test"
        }, {
          "name": "MIME-Version",
          "value": "1.0"
        }, {
          "name": "Content-Type",
          "value": "multipart/alternative; boundary=\"-----=_Part_314159_271828\""
        }, {
          "name": "Message-ID",
          "value": "null"
        }
      ]],
      "common_headers": {
        "from": "sender@example.com",
        "to": ["recipient@example.com"],
        "subject": "Amazon Pinpoint Test"
      }
    },
    "click": {
      "ip_address": "72.21.198.67",
      "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
      "link": "https://aws.amazon.com/pinpoint/"
    }
  }
}
}
}
}
}
}

```

## Apertura di e-mail

L'oggetto JSON per un evento di apertura e-mail contiene i dati mostrati nell'esempio seguente.

```

{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {

```

```
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rn bmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative; boundary=\"-----_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }
        ],
        "common_headers": {
          "from": "sender@example.com",
          "to": ["recipient@example.com"],
          "subject": "Amazon Pinpoint Test"
        }
      }
    }
  },
},
```

```
    "open": {
      "ip_address": "72.21.198.67",
      "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko)"
    }
  }
}
```

## Attributi di eventi e-mail

Questa sezione definisce gli attributi inclusi nei dati del flusso di eventi generati da Amazon Pinpoint quando invii messaggi e-mail.

Attributo	Descrizione
event_type	<p>Tipo di evento. I valori possibili sono:</p> <ul style="list-style-type: none"><li>• <code>_email.send</code>: Amazon Pinpoint ha accettato il messaggio e ha tentato di recapitarlo al destinatario.</li><li>• <code>_email.delivered</code>: il messaggio è stato recapitato al destinatario.</li><li>• <code>_email.rejected</code>: Amazon Pinpoint ha stabilito che il messaggio contiene malware e non ha tentato di inviarlo.</li><li>• <code>_email.hardbounce</code>: un problema permanente ha impedito ad Amazon Pinpoint di recapitare il messaggio. Amazon Pinpoint non tenterà più di recapitare il messaggio.</li><li>• <code>_email.softbounce</code>: un problema temporaneo ha impedito ad Amazon Pinpoint di recapitare il messaggio. Amazon Pinpoint tenterà di recapitare nuovamente il messaggio per un certo periodo di tempo. Se il messaggio continua a non essere recapitato, non verrà</li></ul>

Attributo	Descrizione
	<p>effettuato alcun altro tentativo. Lo stato finale dell'e-mail sarà quindi SOFTBOUNCE.</p> <ul style="list-style-type: none"><li>• <code>_email.complaint</code>: il destinatario ha ricevuto il messaggio e lo ha segnalato al proprio provider e-mail come spam (ad esempio, utilizzando la funzionalità "Segnala come spam" del client e-mail).</li><li>• <code>_email.open</code>: il destinatario ha ricevuto il messaggio e lo ha aperto.</li><li>• <code>_email.click</code>: il destinatario ha ricevuto il messaggio e ha fatto clic su un collegamento al suo interno.</li><li>• <code>_email.unsubscribe</code>: il destinatario ha ricevuto il messaggio e ha fatto clic su un collegamento per l'annullamento della sottoscrizione al suo interno.</li><li>• <code>_email.rendering_failure</code>: l'e-mail non è stata inviata a causa di un errore di rendering. Ciò può verificarsi se i dati del modello mancano o se non vi è corrispondenza tra i parametri e i dati del modello.</li></ul>
<code>event_timestamp</code>	L'ora in cui il messaggio è stato inviato, espressa nel formato orario Unix in millisecondi. Questo valore è in genere lo stesso per tutti gli eventi generati per un messaggio.
<code>arrival_timestamp</code>	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.

Attributo	Descrizione
event_version	<p data-bbox="831 226 1455 260">La versione dello schema JSON dell'evento.</p> <div data-bbox="831 302 1507 667"><p data-bbox="857 344 954 378"> Tip</p><p data-bbox="906 403 1471 621">Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.</p></div>
application	<p data-bbox="831 705 1477 835">Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.</p>
client	<p data-bbox="831 884 1487 1014">Informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Client.</p>
device	<p data-bbox="831 1062 1471 1192">Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Dispositivo.</p> <p data-bbox="831 1234 1461 1268">Per gli eventi e-mail, questo oggetto è vuoto.</p>
session	<p data-bbox="831 1316 1461 1350">Per gli eventi e-mail, questo oggetto è vuoto.</p>

Attributo	Descrizione
<code>attributes</code>	<p>Attributi associati all'evento. Per ulteriori informazioni, consulta la tabella Attributi.</p> <p>Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi creati quando si invia un messaggio da una campagna o da un viaggio, questo oggetto contiene gli attributi associati alla campagna o al viaggio. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso.</p>
<code>client_context</code>	<p>Per gli eventi e-mail, questo oggetto contiene un oggetto denominato <code>custom</code>, che contiene un attributo <code>legacy_identifier</code>. Il valore dell'attributo <code>legacy_identifier</code> è l'ID del progetto da cui è stato inviato il messaggio.</p>
<code>facets</code>	<p>Ulteriori informazioni sul messaggio, ad esempio le intestazioni e-mail. Per ulteriori informazioni, consulta la tabella Facet.</p>
<code>awsAccountId</code>	<p>L'ID dell'account AWS utilizzato per inviare il messaggio.</p>

## Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
<code>app_id</code>	<p>ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.</p>

Attributo	Descrizione
sdk	L'SDK utilizzato per segnalare l'evento. Se invii un messaggio e-mail transazionale chiamando direttamente l'API Amazon Pinpoint o utilizzando la console Amazon Pinpoint, questo oggetto è vuoto.

## Attributi

Include informazioni sulla campagna o viaggio che ha prodotto l'evento.

### Campagna

Include informazioni sulla campagna che ha prodotto l'evento.

Attributo	Descrizione
feedback	Per gli eventi <code>_email.click</code> , il valore di questo attributo è l'URL del collegamento su cui il destinatario ha fatto clic per generare l'evento. Per gli altri eventi, questo valore rappresenta il tipo di evento, ad esempio <code>received</code> , <code>opened</code> o <code>clicked</code> .
treatment_id	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresenta il numero della descrizione (treatment) del messaggio. Per le campagne standard e i messaggi e-mail transazionali, questo valore è <code>0</code> .
campaign_activity_id	ID univoco generato da Amazon Pinpoint quando si verifica l'evento.
campaign_id	L'ID univoco della campagna che ha inviato il messaggio.

## Viaggio

Include informazioni sul viaggio che ha prodotto l'evento.

Attributo	Descrizione
<code>journey_run_id</code>	ID univoco del percorso che ha inviato il messaggio. Amazon Pinpoint genera e assegna automaticamente questo ID a ogni nuova esecuzione di un percorso.
<code>feedback</code>	Per gli eventi <code>_email.click</code> , il valore di questo attributo è l'URL del collegamento su cui il destinatario ha fatto clic per generare l'evento. Per gli altri eventi, questo valore rappresenta il tipo di evento, ad esempio <code>received</code> , <code>delivered</code> o <code>opened</code> .
<code>journey_id</code>	L'ID univoco del viaggio che ha inviato il messaggio.
<code>journey_activity_id</code>	L'ID univoco dell'attività di viaggio che ha inviato il messaggio.

## Client

Identificatore univoco del client che è stato scelto come target dalla campagna o dal percorso.

Attributo	Descrizione
<code>client_id</code>	ID del client. Il valore è l'ID dell'endpoint per campagne e percorsi e, per l'invio transazionale, è un UUID.

## Facet

Include informazioni sul messaggio e sul tipo di evento.

Attributo	Descrizione
<code>email_channel</code>	Contiene un oggetto <code>mail_event</code> , che contiene due oggetti: <code>mail</code> e un oggetto corrispondente al tipo di evento.

## Mail

Include informazioni sul contenuto del messaggio e-mail, nonché i metadati relativi al messaggio.

Attributo	Descrizione
<code>message_id</code>	ID univoco del messaggio. Amazon Pinpoint genera automaticamente questo ID quando accetta il messaggio.
<code>message_send_timestamp</code>	La data e l'ora in cui il messaggio è stato inviato, nel formato specificato in <a href="#">RFC 822</a> .
<code>from_address</code>	L'indirizzo e-mail da cui è stato inviato il messaggio.
<code>destination</code>	Un array che contiene gli indirizzi e-mail a cui è stato inviato il messaggio.
<code>headers_truncated</code>	Un valore booleano che indica se le intestazioni e-mail sono state troncate.
<code>headers</code>	Un oggetto che contiene diverse coppie nome-valore che corrispondono alle intestazioni nel messaggio. Questo oggetto in genere contiene informazioni sulle seguenti intestazioni: <ul style="list-style-type: none"><li>• <code>From</code>: indirizzo e-mail del mittente.</li><li>• <code>To</code>: indirizzo e-mail del destinatario.</li><li>• <code>Subject</code>: riga dell'oggetto dell'e-mail.</li></ul>

Attributo	Descrizione
common_headers	<div data-bbox="860 210 1510 472" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> <b>Tip</b></p> <p>L'intestazione dell'oggetto non è inclusa per gli eventi <code>_email.send</code> della campagna.</p> </div> <ul style="list-style-type: none"> <li>• <code>MIME-Version</code> : indica che il messaggio è in formato MIME. Se questa intestazione è presente, il valore è sempre <code>1.0</code>.</li> <li>• <code>Content-Type</code> : tipo di supporto MIME del contenuto del messaggio.</li> </ul> <p>Contiene informazioni su varie intestazioni comuni per i messaggi e-mail. Le informazioni possono includere la data di invio del messaggio e le righe del destinatario, del mittente e dell'oggetto del messaggio.</p>

## Eventi SMS

Se in un progetto è abilitato il canale SMS, Amazon Pinpoint può eseguire lo streaming dei dati di eventi relativi ai recapiti dei messaggi SMS per il progetto. La ricezione degli eventi SMS generati dagli operatori può richiedere fino a 72 ore. Tali eventi non devono essere utilizzati per determinare se esiste un ritardo nel recapito dei messaggi in uscita. Dopo 72 ore, se Amazon Pinpoint non ha ricevuto un evento finale da un operatore, il servizio restituirà automaticamente un valore `record_status UNKNOWN` poiché non è noto cosa sia successo a tale messaggio.

## Esempio

L'oggetto JSON per un evento SMS contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
```

```

"event_version": "3.1",
"application": {
  "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
  "sdk": {}
},
"client": {
  "client_id": "123456789012"
},
"device": {
  "platform": {}
},
"session": {},
"attributes": {
  "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
  "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
  "origination_phone_number": "+12065550142",
  "destination_phone_number": "+14255550199",
  "record_status": "DELIVERED",
  "iso_country_code": "US",
  "treatment_id": "0",
  "number_of_message_parts": "1",
  "message_id": "1111-2222-3333",
  "message_type": "Transactional",
  "campaign_id": "52dc44b35c4742c98c5935269example"
},
"metrics": {
  "price_in_millicents_usd": 645.0
},
"awsAccountId": "123456789012"
}

```

## Attributi di eventi SMS

Questa sezione definisce gli attributi inclusi nei dati del flusso di eventi generati da Amazon Pinpoint quando invii messaggi SMS.

### Evento

Attributo	Descrizione
event_type	Tipo di evento. I valori possibili sono:

Attributo	Descrizione
	<ul style="list-style-type: none"><li>• <code>_SMS.BUFFERED</code>: il messaggio non è ancora stato recapitato al destinatario.</li><li>• <code>_SMS.SUCCESS</code>: il messaggio è stato correttamente accettato dall'operatore/recapitato al destinatario.</li><li>• <code>_SMS.FAILURE</code>: Amazon Pinpoint non è stato in grado di recapitare il messaggio al destinatario. Per ulteriori informazioni sull'errore che ha impedito il recapito del messaggio, consulta <code>attribute.s.record_status</code>.</li><li>• <code>_SMS.OPTOUT</code>: il cliente ha ricevuto il messaggio e ha risposto inviando la parola chiave di opt-out (in genere "STOP").</li></ul>
<code>event_timestamp</code>	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
<code>arrival_timestamp</code>	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
<code>event_version</code>	La versione dello schema JSON dell'evento. <div data-bbox="829 1325 1507 1688" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> <b>Tip</b></p><p>Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.</p></div>
<code>application</code>	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Applicazione</a> .

Attributo	Descrizione
<code>client</code>	Informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Client</a> .
<code>device</code>	Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Dispositivo</a> .  Per gli eventi SMS, questo oggetto è vuoto.
<code>session</code>	Per gli eventi SMS, questo oggetto è vuoto.
<code>attributes</code>	Attributi associati all'evento. Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi che vengono creati quando si invia una campagna, questo oggetto contiene gli attributi associati alla campagna. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso.  Per ulteriori informazioni, consulta la tabella <a href="#">Attributi</a> .
<code>metrics</code>	Parametri aggiuntivi associati all'evento. Per ulteriori informazioni, consulta la tabella <a href="#">Parametri</a> .
<code>awsAccountId</code>	L'ID dell'account AWS utilizzato per inviare il messaggio.

## Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento e, se applicabile, sull'SDK utilizzato per segnalare l'evento.

Attributo	Descrizione
app_id	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
sdk	L'SDK utilizzato per segnalare l'evento. Se invii un messaggio SMS transazionale chiamando direttamente l'API Amazon Pinpoint o utilizzando la console Amazon Pinpoint, questo oggetto è vuoto.

## Attributi

Include informazioni sugli attributi associati all'evento.

Attributo	Descrizione
sender_request_id	Un ID univoco associato alla richiesta di invio del messaggio SMS.
campaign_activity_id	L'ID univoco dell'attività all'interno della campagna.
origination_phone_number	Il numero di telefono da cui è stato inviato il messaggio.
destination_phone_number	Il numero di telefono a cui hai tentato di inviare il messaggio.
record_status	Ulteriori informazioni sullo stato del messaggio. I valori possibili includono: <ul style="list-style-type: none"><li>SUCCESSFUL/DELIVERED: il recapito del messaggio è riuscito.</li></ul>

Attributo	Descrizione
	<ul style="list-style-type: none"><li>• <b>PENDING</b>: il messaggio non è stato ancora recapitato al dispositivo del destinatario.</li><li>• <b>INVALID</b>: il numero di telefono di destinazione non è valido.</li><li>• <b>UNREACHABLE</b>: il dispositivo del destinatario è attualmente irraggiungibile o non disponibile. Ad esempio, il dispositivo potrebbe essere spento o disconnesso dalla rete. Puoi provare a inviare nuovamente il messaggio in un secondo momento.</li><li>• <b>UNKNOWN</b>: si è verificato un errore che ha impedito il recapito del messaggio. Questo errore è in genere transitorio e puoi tentare di inviare nuovamente il messaggio in un secondo momento.</li><li>• <b>BLOCKED</b>: il dispositivo del destinatario sta bloccando i messaggi SMS dal numero di origine.</li><li>• <b>CARRIER_UNREACHABLE</b>: un problema a livello di rete mobile del destinatario ha impedito il recapito del messaggio. Questo errore è in genere transitorio e puoi tentare di inviare nuovamente il messaggio in un secondo momento.</li><li>• <b>SPAM</b>: l'operatore di telefonia mobile del destinatario ha identificato il contenuto del messaggio come spam e ha bloccato il recapito del messaggio.</li><li>• <b>INVALID_MESSAGE</b>: il corpo del messaggio SMS non è valido e non può essere recapitato.</li><li>• <b>CARRIER_BLOCKED</b>: l'operatore del destinatario ha bloccato il recapito di questo</li></ul>

Attributo	Descrizione
	<p>messaggio. Ciò si verifica spesso quando il carrier identifica il contenuto del messaggio come non richiesto o dannoso.</p> <ul style="list-style-type: none"><li>• <b>TTL_EXPIRED</b>: non è stato possibile recapitare il messaggio SMS entro un determinato intervallo di tempo. Questo errore è in genere transitorio e puoi tentare di inviare nuovamente il messaggio in un secondo momento.</li><li>• <b>MAX_PRICE_EXCEEDED</b>: l'invio del messaggio comporta un addebito che supera la quota mensile di spesa per SMS per il tuo account. Puoi richiedere un aumento di questa quota completando la procedura descritta in <a href="#">Richiesta di aumento della quota di spesa mensile per l'invio di SMS per Amazon SNS</a> nella Guida per l'utente di Amazon Pinpoint.</li><li>• <b>OPTED_OUT</b> : il messaggio SMS non è stato inviato perché il destinatario ha scelto mediante opt-out di non ricevere messaggi da te.</li><li>• <b>NO_QUOTA_LEFT_ON_ACCOUNT</b>: sul tuo account non è disponibile una quota di spesa sufficiente per inviare il messaggio . Puoi richiedere un aumento di questa quota completando la procedura descritta in <a href="#">Richiesta di aumento della quota di spesa mensile per l'invio di SMS per Amazon SNS</a> nella Guida per l'utente di Amazon Pinpoint.</li><li>• <b>NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND</b>: il tuo account non contiene un numero di telefono utilizzabile per inviare il messaggio alla destinazione.</li></ul>

Attributo	Descrizione
	<ul style="list-style-type: none"><li>• <b>DESTINATION_COUNTRY_NOT_SUPPORTED</b>: il paese di destinazione è bloccato. Per tutti i paesi supportati, consulta <a href="#">Regioni e paesi supportati (canale SMS)</a>.</li><li>• <b>ACCOUNT_IN_SANDBOX</b>: il tuo account si trova in un ambiente di sperimentazione (sandbox) e può inviare messaggi solo a numeri di destinazione verificati. Puoi verificare il numero di destinazione nella console Amazon Pinpoint o avviare il processo per spostare l'account dall'ambiente di sperimentazione (sandbox). Consulta <a href="#">Passaggio dall'ambiente di sperimentazione (sandbox) per SMS di Amazon Pinpoint alla produzione</a>.</li><li>• <b>RATE_EXCEEDED</b>: hai tentato di inviare un messaggio troppo velocemente e si è verificata una limitazione (della larghezza di banda della rete). Devi rallentare la frequenza delle chiamate. Per informazioni dettagliate sui limiti validi, consulta <a href="#">Limiti relativi al numero di parti di messaggio al secondo (MPS)</a>.</li><li>• <b>INVALID_ORIGINATION_IDENTITY</b>: l'identità di origine specificata non è valida.</li><li>• <b>ORIGINATION_IDENTITY_DOES_NOT_EXIST</b>: l'identità di origine specificata non esiste.</li><li>• <b>INVALID_DLT_PARAMETERS</b>: sono stati specificati parametri DLT non validi (obbligatori per le destinazioni in India).</li><li>• <b>INVALID_PARAMETERS</b>: sono stati specificati parametri non validi.</li></ul>

Attributo	Descrizione
	<ul style="list-style-type: none"> <li>• <b>ACCESS_DENIED</b>: al tuo account è impedito l'invio di messaggi. Contatta il Supporto clienti per scoprire la causa e risolvere il problema.</li> <li>• <b>INVALID_KEYWORD</b>: la parola chiave specificata non è valida. È possibile che la parola chiave abbia un formato errato o non sia stata impostata nel tuo account.</li> <li>• <b>INVALID_SENDER_ID</b>: l'ID mittente specificato non è valido. È possibile che l'ID mittente abbia un formato o una lunghezza errati.</li> <li>• <b>INVALID_POOL_ID</b>: l'ID del pool specificato non è valido. È possibile che l'ID del pool sia essere in un formato errato o non appartenga al tuo account.</li> <li>• <b>SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION</b>: il paese di destinazione non supporta l'ID mittente. È necessario utilizzare un numero di telefono o un'altra identità di origine per l'invio.</li> <li>• <b>INVALID_PHONE_NUMBER</b>: il numero di telefono di origine specificato non è valido. È possibile che il numero di telefono abbia un formato o una lunghezza errati.</li> </ul>
iso_country_code	Il paese associati al numero di telefono del destinatario, mostrati nel formato ISO 3166-1 alpha-2.
treatment_id	L'ID della descrizione (treatment) del messaggio, se il messaggio è stato inviato in una campagna A/B.

Attributo	Descrizione
<code>treatment_id</code>	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresenta il numero della descrizione (treatment) del messaggio. Per i messaggi SMS transazionali, questo valore è 0.
<code>number_of_message_parts</code>	<p>Numero di parti di messaggio che Amazon Pinpoint ha creato per inviare il messaggio.</p> <p>In genere, i messaggi SMS possono contenere solo 160 caratteri GSM-7 o 67 caratteri non GSM, anche se questi limiti possono variare in base al paese . Se invii un messaggio che supera questi limiti, Amazon Pinpoint lo suddivide automaticamente in parti più piccole. I costi vengono fatturati in base al numero di parti del messaggio inviate.</p>
<code>message_id</code>	ID univoco che Amazon Pinpoint genera quando accetta il messaggio.
<code>message_type</code>	Il tipo di messaggio. I valori possibili sono Promotional (Promozionale) e Transactional (Transazionale). Puoi specificare questo valore quando crei una campagna o quando invii messaggi transazionali utilizzando l'operazione <a href="#">SendMessage</a> nell'API Amazon Pinpoint.
<code>campaign_id</code>	ID univoco della campagna Amazon Pinpoint che ha inviato il messaggio.

## Client

Include informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento.

Attributo	Descrizione
<code>client_id</code>	<p>Per gli eventi generati dalle app, questo valore è l'ID univoco del client dell'app installato sul dispositivo. Questo ID viene generato automaticamente dall'AWS Mobile SDK for iOS e dall'AWS Mobile SDK for Android.</p> <p>Per gli eventi generati quando invii campagne e messaggi transazionali, questo valore è uguale all'ID dell'endpoint a cui hai inviato il messaggio.</p>
<code>cognito_id</code>	ID univoco assegnato al client dell'app nel pool di identità Amazon Cognito utilizzato dall'app.

## Dispositivo

Include informazioni sul dispositivo che ha segnalato l'evento.

Attributo	Descrizione
<code>locale</code>	Le impostazioni locali del dispositivo.
<code>make</code>	La marca del dispositivo, ad esempio Apple o Samsung.
<code>model</code>	Il modello del dispositivo, ad esempio iPhone.
<code>platform</code>	La piattaforma del dispositivo, ad esempio ios o android.

## Parametri

Include informazioni sui parametri associati all'evento.

Attributo	Descrizione
price_in_millicents_usd	<p>L'importo addebitato per l'invio del messaggio . Questo prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000=0,645¢=\$0,00645).</p> <div data-bbox="829 562 1507 829"><p> <b>Note</b></p><p>Questa proprietà non viene visualizzata per i messaggi con un event_type di _SMS.BUFFERED.</p></div>

# Esecuzione di query sui dati di analisi di Amazon Pinpoint

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, puoi utilizzare l'API di analisi Amazon Pinpoint per eseguire query sui dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni dettagliate sulle tendenze relative al coinvolgimento dell'utente, alla sensibilizzazione della campagna e a molto altro. Questi parametri, detti anche indicatori di prestazioni chiave (KPI), sono i valori misurabili che possono aiutare a monitorare e valutare le prestazioni dei progetti, delle campagne e dei viaggi.

Se utilizzi le API per eseguire query sui dati di analisi, puoi analizzare i dati utilizzando lo strumento di reporting di tua scelta, senza dover accedere alla console Amazon Pinpoint o analizzare i dati degli eventi in formato RAW da origini quali i flussi Amazon Kinesis. Ad esempio, puoi creare un dashboard personalizzato che visualizzi i risultati settimanali delle campagne o che fornisca analisi approfondite sulle percentuali di consegna per le tue campagne.

Puoi interrogare i dati utilizzando l'API REST di Amazon Pinpoint, AWS Command Line Interface (AWS CLI) o un AWS SDK. Per eseguire una query sui dati, invia una richiesta all'API Amazon Pinpoint e utilizza i parametri supportati per specificare i dati desiderati e i filtri da applicare. Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. È quindi possibile trasferire i risultati a un altro servizio o a un'altra applicazione per un'analisi più approfondita, lo storage o la generazione di report.

## Parametri supportati

Amazon Pinpoint fornisce l'accesso programmatico ai dati di analisi per diversi tipi di metriche standard:

- **Metriche delle applicazioni:** queste metriche forniscono informazioni sulle tendenze relative a tutte le campagne e ai messaggi transazionali associati a un progetto. Ad esempio, è possibile utilizzare un parametro di applicazione per ottenere i dettagli del numero di messaggi aperti dai destinatari per ogni campagna associata a un progetto. Per accedere ai dati per una metrica dell'applicazione, utilizza la risorsa [Metriche delle applicazioni](#) dell'API Amazon Pinpoint.
- **Metriche della campagna:** queste metriche forniscono informazioni dettagliate sulle prestazioni delle campagne individuali. Ad esempio, è possibile utilizzare un parametro di campagna per stabilire il numero di endpoint a cui è stato inviato un messaggio di campagna. Per accedere ai dati per una metrica della campagna, utilizza la risorsa [Parametri della campagna](#) dell'API Amazon Pinpoint.

- **Metriche di coinvolgimento del percorso:** queste metriche forniscono informazioni dettagliate sulle prestazioni dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di coinvolgimento del viaggio per ottenere una ripartizione del numero di messaggi aperti dai partecipanti a ciascuna attività di un viaggio. Per accedere ai dati per una metrica di coinvolgimento del percorso, utilizza la risorsa [Metriche di coinvolgimento del percorso](#) dell'API Amazon Pinpoint.
- **Metriche di esecuzione del percorso:** queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di esecuzione del viaggio per determinare quanti partecipanti stanno procedendo nelle attività del viaggio. Per accedere ai dati per una metrica di esecuzione del percorso, utilizza la risorsa [Metriche di esecuzione delle attività del percorso](#) dell'API Amazon Pinpoint.
- **Metriche di esecuzione delle attività del percorso:** queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione delle singole attività in un percorso. Ad esempio, è possibile utilizzare un parametro di esecuzione dell'attività di viaggio per determinare quanti partecipanti hanno completato un'attività. Per accedere ai dati per una metrica di esecuzione dell'attività del percorso, utilizza la risorsa [Metriche di esecuzione delle attività del percorso](#) dell'API Amazon Pinpoint.

Per un elenco completo dei parametri standard su cui è possibile eseguire query a livello di programmazione, consulta [Parametri standard](#).

Amazon Pinpoint raccoglie e aggrega automaticamente i dati per tutti i tipi di metriche supportati e per tutti i progetti, le campagne e i percorsi. Inoltre, i dati vengono aggiornati in modo continuo, generando un intervallo di tempo di latenza dei dati che è limitato a circa due ore. Si noti, tuttavia, che potrebbe esserci una latenza dei dati aggiuntiva per determinati parametri. Questo perché i dati per alcuni parametri si basano sulle informazioni che riceviamo dai provider di posta elettronica dei destinatari. Alcuni fornitori ci inviano immediatamente queste informazioni, mentre altri le inviano meno frequentemente.

Amazon Pinpoint archivia i dati per 90 giorni. Per archiviare i dati per più di 90 giorni o per accedere ai dati di analisi non elaborati in tempo reale, puoi configurare un progetto Amazon Pinpoint per lo streaming dei dati degli eventi su Amazon Kinesis Data Streams o Amazon Data Firehose. Per informazioni sulla configurazione dei flussi di eventi, consulta [Streaming di eventi Amazon Pinpoint su Kinesis](#).

## Nozioni di base delle query

Per eseguire una query sui dati per un parametro, inviare una richiesta get alla risorsa dei parametri appropriata dell'API di Amazon Pinpoint. Nella richiesta, è possibile definire la query utilizzando parametri supportati per i seguenti componenti di query:

- **Progetto:** specifica un progetto fornendo il relativo ID come valore per il parametro `application-id`. Questo parametro è obbligatorio per tutti i parametri.
- **Campagna:** specifica una campagna fornendo il relativo ID come valore per il parametro `campaign-id`. Questo parametro è obbligatorio solo per i parametri della campagna.
- **Percorso:** specificare un percorso fornendo il relativo ID come valore per il parametro `journey-id`. Questo parametro è obbligatorio solo per i parametri di coinvolgimento e di esecuzione del viaggio e per i parametri di esecuzione dell'attività di viaggio.
- **Attività del percorso:** specifica un'attività del percorso fornendo il relativo ID come valore per il parametro `journey-activity-id`. Questo parametro è obbligatorio solo per i parametri di esecuzione dell'attività di viaggio.
- **Intervallo di date:** per filtrare facoltativamente i dati per intervallo di date, specifica la prima e l'ultima data e l'ora dell'intervallo utilizzando i parametri di ora di inizio e fine supportati. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019.

Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. I parametri dell'intervallo di date sono supportati da tutti i parametri, ad eccezione dei parametri di esecuzione del viaggio e dei parametri di esecuzione dell'attività di viaggio.

- **Parametro:** specifica una metrica fornendo il relativo nome come valore per il parametro `kpi-name`. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Alcuni esempi sono `email-open-rate` e `successful-delivery-rate`. Questo parametro è obbligatorio per tutti i parametri, ad eccezione dei parametri di esecuzione del viaggio e dei parametri di esecuzione dell'attività di viaggio. Per un elenco completo dei parametri supportati e il valore `kpi-name` da utilizzare per ciascuno di essi, consulta [Parametri standard](#).

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. Nella risposta, la struttura dei risultati varia a seconda del parametro richiesto.

Alcune metriche forniscono un solo valore, ad esempio il numero di messaggi recapitati da una campagna. Altre metriche forniscono più valori e in genere raggruppano tali valori in base a un campo pertinente, ad esempio il numero di messaggi recapitati da ciascuna campagna, raggruppati per esecuzione della campagna. Se un parametro fornisce e raggruppa più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati. Per ulteriori informazioni sulla struttura dei risultati delle query, consulta [Utilizzo dei risultati delle query](#).

## Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint

Utilizzando l'API Amazon Pinpoint, puoi eseguire query sui dati di analisi per un sottoinsieme di metriche standard, dette anche indicatori di prestazioni chiave (KPI), applicabili a progetti, campagne e percorsi Amazon Pinpoint. Questi parametri consentono di monitorare e valutare le prestazioni dei progetti, delle campagne e dei viaggi.

Per gestire l'accesso a questi dati, puoi creare policy AWS Identity and Access Management (IAM) che definiscano le autorizzazioni per i ruoli o gli utenti IAM autorizzati ad accedere ai dati. Per supportare il controllo granulare dell'accesso a questi dati, Amazon Pinpoint fornisce diverse azioni distinte che è possibile specificare nelle policy IAM. Per visualizzare i dati di analisi sulla console Amazon Pinpoint (`mobiletargeting:GetReports`), è disponibile un'azione distinta, mentre per accedere ai dati di analisi a livello di programmazione tramite l'API Amazon Pinpoint sono disponibili altre azioni.

Per creare policy IAM che gestiscano l'accesso ai dati di analisi, puoi utilizzare la AWS Management Console, l'AWS CLI o l'API IAM. Tieni presente che la scheda Editor visivo sulla AWS Management Console al momento non include azioni per la visualizzazione o l'esecuzione di query sui dati di analisi di Amazon Pinpoint. Tuttavia, puoi aggiungere manualmente le operazioni necessarie alle policy IAM utilizzando la scheda JSON sulla console.

Ad esempio, la policy seguente consente l'accesso programmatico a tutti i dati di analisi per tutti i progetti, le campagne e i viaggi in tutte le regioni AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "QueryAllAnalytics",
    "Effect": "Allow",
    "Action": [
        "mobiletargeting:GetApplicationDateRangeKpi",
        "mobiletargeting:GetCampaignDateRangeKpi",
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
    ],
    "Resource": [
        "arn:aws:mobiletargeting:*:accountId:apps/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/campaigns/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/execution-
metrics",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/activities/*/
execution-metrics"
    ]
}
]
}

```

Dove *accountId* è il tuo ID dell'account AWS.

Tuttavia, è consigliabile definire policy in grado di seguire il principio del privilegio minimo. In altre parole, è necessario creare policy che includano solo le autorizzazioni necessarie per eseguire un'attività specifica. Per supportare questa pratica e implementare un controllo più granulare, puoi limitare l'accesso programmatico ai dati di analisi solo per un particolare progetto in una regione AWS specifica, ad esempio:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryProjectAnalytics",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:GetApplicationDateRangeKpi",
        "mobiletargeting:GetCampaignDateRangeKpi",
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
      ],
    }
  ],
}

```

```

    "Resource": [
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/*",
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/*/
kpis/*",
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
kpis/*",
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
execution-metrics",
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
activities/*/execution-metrics"
    ]
  }
}

```

Dove:

- *region* è il nome della regione AWS che ospita il progetto.
- *accountId* è l'ID dell'account AWS.
- *projectId* è l'identificatore del progetto a cui desideri consentire l'accesso.

Analogamente, la seguente policy di esempio consente l'accesso programmatico ai dati di analisi solo per una particolare campagna:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryCampaignAnalytics",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetCampaignDateRangeKpi",
      "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/
campaigns/campaignId/kpis/*"
    }
  ]
}

```

Dove:

- *region* è il nome della regione AWS che ospita il progetto.
- *accountId* è il tuo ID Account AWS.

- *projectId* è l'identificatore del progetto associato alla campagna.
- *campaignId* è l'identificatore della campagna a cui desideri consentire l'accesso.

E la seguente policy di esempio consente l'accesso programmatico a tutti i dati di analisi, sia di coinvolgimento sia di esecuzione, per un particolare percorso e le attività che lo compongono:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryJourneyAnalytics",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
        journeys/journeyId/kpis/*",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
        journeys/journeyId/execution-metrics",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
        journeys/journeyId/activities/*/execution-metrics"
      ]
    }
  ]
}
```

Dove:

- *region* è il nome della regione AWS che ospita il progetto.
- *accountId* è l'ID dell'account AWS.
- *projectId* è l'identificatore del progetto associato al viaggio.
- *projectId* è l'identificatore del viaggio a cui desideri consentire l'accesso.

Per l'elenco completo delle azioni API Amazon Pinpoint che è possibile utilizzare nelle policy IAM, consulta [Azioni di Amazon Pinpoint per le policy IAM](#). Per ulteriori informazioni sulla creazione e sulla gestione delle policy IAM, consulta la [Guida per l'utente di IAM](#).

## Metriche di analisi standard di Amazon Pinpoint

È possibile utilizzare le API di analisi di Amazon Pinpoint per eseguire query sui dati di analisi per un sottoinsieme di metriche standard applicabili a progetti, campagne e percorsi Amazon Pinpoint. Questi parametri, detti anche indicatori di prestazioni chiave (KPI), sono i valori misurabili che possono aiutare a monitorare e valutare le prestazioni dei progetti, delle campagne e dei viaggi.

Amazon Pinpoint fornisce l'accesso programmatico ai dati di analisi per diversi tipi di metriche standard:

- **Metriche delle applicazioni:** queste metriche forniscono informazioni dettagliate sulle tendenze per tutte le campagne associate a un progetto, che viene indicato anche come applicazione. Ad esempio, è possibile utilizzare un parametro di applicazione per ottenere i dettagli del numero di messaggi aperti dai destinatari per ogni campagna associata a un progetto.
- **Metriche della campagna:** queste metriche forniscono informazioni dettagliate sulle prestazioni delle campagne individuali. Ad esempio, è possibile utilizzare un parametro di campagna per stabilire il numero di endpoint ai quali è stato inviato un messaggio della campagna o quanti di questi messaggi sono stati distribuiti agli endpoint.
- **Metriche di coinvolgimento del percorso:** queste metriche forniscono informazioni dettagliate sulle prestazioni dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di coinvolgimento del viaggio per ottenere una ripartizione del numero di messaggi aperti dai partecipanti a ciascuna attività di un viaggio.
- **Metriche di esecuzione del percorso:** queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di esecuzione del viaggio per determinare quanti partecipanti hanno iniziato un viaggio.
- **Metriche di esecuzione delle attività del percorso:** queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione delle singole attività in un percorso. Ad esempio, è possibile utilizzare un parametro di esecuzione dell'attività di viaggio per determinare quanti partecipanti hanno iniziato un'attività e quanti partecipanti hanno completato ciascun percorso in un'attività.

Gli argomenti in questa sezione elencano e descrivono i singoli parametri su cui è possibile eseguire query per ogni tipo di parametro.

### Argomenti

- [Parametri delle applicazioni per le campagne](#)

- [Parametri dell'applicazione per messaggi e-mail transazionali](#)
- [Parametri dell'applicazione per messaggi SMS transazionali](#)
- [Parametri della campagna](#)
- [Parametri di coinvolgimento del viaggio](#)
- [Parametri di esecuzione del viaggio](#)
- [Parametri di esecuzione delle attività di viaggio](#)
- [Metriche relative all'esecuzione di percorsi e campagne](#)

## Parametri delle applicazioni per le campagne

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per valutare le prestazioni di tutte le campagne associate a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa [Metriche delle applicazioni](#) dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro kpi-name in una query.

Metrica	Kpi-name	Descrizione
Percentuale di recapiti	successful-delivery-rate	<p>Per tutte le campagne associate a un progetto, la percentuale di messaggi inviati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati da tutte le campagne per un progetto e recapitati ai destinatari, diviso il numero di messaggi che sono stati inviati da tutte le campagne.</p>
Percentuale di recapiti, raggruppati per data	successful-delivery-rate-grouped-by-date	Per tutte le campagne associate a un progetto, la percentuale di messaggi recapitati ai destinatari per

Metrica	Kpi-name	Descrizione
		<p>ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati da tutte le campagne per un progetto e recapitati ai destinatari, diviso il numero di messaggi che sono stati inviati da tutte le campagne, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Percentuale di e-mail aperte	email-open-rate	<p>Per tutte le campagne associate a un progetto, la percentuale di messaggi e-mail aperti dai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail inviati da tutte le campagne per un progetto e aperti dai destinatari, diviso il numero di messaggi e-mail inviati da tutte le campagne e recapitati ai destinatari.</p>

Metrica	Kpi-name	Descrizione
Percentuale di e-mail aperte, raggruppate per campagna	email-open-rate-grouped-by-campaign	<p>Per ogni campagna associata a un progetto, la percentuale di messaggi e-mail aperti dai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail inviati da una campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati dalla campagna e recapitati ai destinatari.</p> <p>I risultati della query per questo parametro sono raggruppati in base all'ID campagna (CampaignId ), che è una stringa che identifica in modo univoco una campagna.</p>
Consegne endpoint	unique-deliveries	Per tutte le campagne associate a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.

Metrica	Kpi-name	Descrizione
Consegne endpoint, raggruppate per campagna	unique-deliveries-grouped-by-campaign	<p>Per ogni campagna associata a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.</p> <p>I risultati della query per questo parametro sono raggruppati in base all'ID campagna (CampaignId ), che è una stringa che identifica in modo univoco una campagna.</p>
Consegne endpoint, raggruppate per data	unique-deliveries-grouped-by-date	<p>Per tutte le campagne associate a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Messaggi recapitati, raggruppati per campagna	successful-deliveries-grouped-by-campaign	<p>Per ogni campagna associata a un progetto, il numero di messaggi che sono stati recapitati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati da una campagna, meno il numero di messaggi che sono stati inviati dalla campagna e non sono stati recapitati ai destinatari a causa di un hard bounce.</p> <p>I risultati della query per questo parametro sono raggruppati in base all'ID campagna (CampaignId ), che è una stringa che identifica in modo univoco una campagna.</p>

Metrica	Kpi-name	Descrizione
Percentuale di notifiche push aperte	push-open-rate	<p>Per tutte le campagne associate a un progetto, la percentuale di notifiche push aperte dai destinatari.</p> <p>Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da tutte le campagne per un progetto e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate da tutte quelle campagne e consegnate ai destinatari.</p>

Metrica	Kpi-name	Descrizione
Percentuale di notifiche push aperte, raggruppate per campagna	push-open-rate-grouped-by-campaign	<p>Per ogni campagna associata a un progetto, la percentuale di notifiche push aperte dai destinatari.</p> <p>Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate tramite una campagna e aperte dai destinatari, diviso il numero di notifiche push inviate dalla campagna e consegnate ai destinatari.</p> <p>I risultati della query per questo parametro sono raggruppati in base all'ID campagna (CampaignId), che è una stringa che identifica in modo univoco una campagna.</p>

## Parametri dell'applicazione per messaggi e-mail transazionali

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per monitorare le tendenze relative a tutti i messaggi e-mail transazionali associati a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa [Metriche delle applicazioni](#) dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro kpi-name in una query.

Tieni presente che questi parametri non forniscono dati sui messaggi e-mail inviati dalle campagne. Forniscono dati solo sui messaggi e-mail transazionali. Per eseguire query sui dati per i messaggi inviati da una o più campagne, utilizza un [parametro della campagna](#) o un [parametro di applicazione per le campagne](#).

Metrica	Kpi-name	Descrizione
Clic	txn-emails-clicked	<p>Il numero di volte che i destinatari hanno fatto clic sui collegamenti nei messaggi. Se un singolo destinatario ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.</p>
Clic, raggruppati per data	txn-emails-clicked-grouped-by-date	<p>Il numero di volte in cui i destinatari hanno fatto clic sui collegamenti nei messaggi, per ogni giorno nell'intervallo di date specificato. Se un singolo destinatario ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Complaint rate (Percentuale di reclami)	txn-emails-complaint-rate	<p>La percentuale di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate.</p> <p>Questo parametro viene calcolato come il numero</p>

Metrica	Kpi-name	Descrizione
		<p>di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, diviso per il numero di messaggi inviati.</p>
<p>Percentuale di reclami, raggruppati per data</p>	<p>txn-emails-complaint-rate-grouped-by-date</p>	<p>La percentuale di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, per ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, diviso per il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
<p>Reclami</p>	<p>txn-emails-with-complaints</p>	<p>Il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate.</p>

Metrica	Kpi-name	Descrizione
Reclami, raggruppati per data	txn-emails-with-complaints-grouped-by-date	<p>Il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Consegne	txn-emails-delivered	<p>Il numero di messaggi consegnati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.</p>

Metrica	Kpi-name	Descrizione
Consegne, raggruppate per data	txn-emails-delivered-grouped-by-date	<p>Il numero di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati, per ogni giorno nell'intervallo di date specificato. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Percentuale di recapiti	txn-emails-delivery-rate	<p>La percentuale di messaggi consegnati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati.</p>

Metrica	Kpi-name	Descrizione
Percentuale di recapiti, raggruppati per data	txn-emails-delivery-rate-grouped-by-date	<p>La percentuale di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Mancati recapiti permanenti	txn-emails-hard-bounced	<p>Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del destinatario non esiste.</p>

Metrica	Kpi-name	Descrizione
Hard bounce, raggruppati per data	txn-emails-hard-bounced-grouped-by-date	<p>Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un hard bounce, per ogni giorno nell'intervallo di date specificato. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del destinatario non esiste.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Aperture	txn-emails-opened	Mostra il numero di messaggi che sono stati aperti dai destinatari:
Aperture, raggruppate per data	txn-emails-opened-grouped-by-date	<p>Il numero di messaggi che sono stati aperti dai destinatari, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Invii	txn-emails-sent	Il numero di messaggi che sono stati inviati.

Metrica	Kpi-name	Descrizione
Invii, raggruppati per data	txn-emails-sent-grouped-by-date	<p>Il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Soft bounce	txn-emails-soft-bounced	<p>Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo o impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del destinatario è piena o il server di ricezione è temporaneamente non disponibile.</p>

Metrica	Kpi-name	Descrizione
Soft bounce, raggruppati per data	txn-emails-soft-bounced-grouped-by-date	<p>Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un soft bounce, per ogni giorno nell'intervallo di date specificato. Si verifica un caso di e-mail non recapitata se un problema temporaneo impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del destinatario è piena o il server di ricezione è temporaneamente non disponibile.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Eventi clic utente univoci	txn-emails-unique-clicks	<p>Il numero di destinatari univoci (endpoint) che hanno fatto clic sui collegamenti nei messaggi.</p> <p>A differenza del parametro Clicks (Clic), questo parametro indica il numero di destinatari univoci che hanno fatto clic sui collegamenti, non il numero di eventi di clic che si sono verificati. Ad esempio, se un singolo destinatario ha fatto clic su più collegamenti nello stesso messaggio o ha fatto clic sullo stesso link più di una volta, questo parametro segnala un solo evento clic per quel destinatario.</p>

Metrica	Kpi-name	Descrizione
Eventi clic utente univoci, raggruppati per data	txn-emails-unique-clicks-grouped-by-date	<p>Il numero di destinatari univoci (endpoint) che hanno fatto clic sui collegamenti nei messaggi, per ogni giorno nell'intervallo di date specificato.</p> <p>A differenza del parametro Clicks, grouped by date (Clic, raggruppati per data) questo parametro indica il numero di destinatari univoci che hanno fatto clic sui collegamenti, non il numero di eventi clic che si sono verificati. Ad esempio, se un singolo destinatario ha fatto clic su più collegamenti nello stesso messaggio o ha fatto clic sullo stesso link più di una volta, questo parametro segnala un solo evento clic per quel destinatario.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Eventi univoci di apertura degli utenti	txn-emails-unique-opens	<p>Il numero di destinatari univoci (endpoint) che hanno aperto i messaggi.</p> <p>A differenza del parametro Opens (Aperture), questo parametro indica il numero di destinatari univoci che hanno aperto i messaggi, non il numero di eventi di apertura che si sono verificati. Ad esempio, se un singolo destinatario apre lo stesso messaggio più volte, questo parametro segnala un solo evento di apertura per quel destinatario.</p>

Metrica	Kpi-name	Descrizione
Eventi univoci di apertura degli utenti, raggruppati per data	txn-emails-unique-opens-grouped-by-date	<p>Il numero di destinatari univoci (endpoint) che hanno aperto messaggi, per ogni giorno nell'intervallo di date specificato.</p> <p>A differenza del parametro Opens, grouped by date (Aperture, raggruppate per data) questo parametro indica il numero di destinatari univoci che hanno aperto i messaggi, non il numero di eventi di apertura che si sono verificati. Ad esempio, se un singolo destinatario apre lo stesso messaggio più volte, questo parametro segnala un solo evento di apertura per quel destinatario.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

## Parametri dell'applicazione per messaggi SMS transazionali

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per ottenere informazioni sulle tendenze relative a tutti i messaggi SMS transazionali associati a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa [Metriche delle applicazioni](#) dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro kpi-name in una query.

Tieni presente che questi parametri non forniscono dati sui messaggi SMS inviati dalle campagne. Forniscono dati solo sui messaggi SMS transazionali. Per eseguire query sui dati per i messaggi inviati da una o più campagne, utilizza un [parametro della campagna](#) o un [parametro di applicazione per le campagne](#).

Metrica	Kpi-name	Descrizione
Prezzo medio per messaggio, raggruppato per paese	txn-sms-average-price-grouped-by-country	<p>Costo totale dell'invio dei messaggi, per ogni paese o regione a cui sono stati inviati i messaggi. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 = 0,645¢ = \$0,00645).</p> <p>Questo parametro viene calcolato come costo totale di tutti i messaggi inviati ai destinatari in ciascun paese o regione, diviso per il numero di messaggi inviati ai destinatari in ciascuno di tali paesi e regioni.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>
Prezzo medio per parte di messaggio, raggruppato per paese	txn-sms-average-price-by-parts-grouped-by-country	Costo totale dell'invio delle parti di messaggio, per ogni paese o regione a cui sono stati inviati i messaggi. Una

Metrica	Kpi-name	Descrizione
		<p>parte di messaggio è una sezione di un messaggio SMS. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 = 0,645¢ = \$0,00645).</p> <p>Questa metrica viene calcolata come costo totale di tutte le parti di messaggio inviate ai destinatari in ciascun paese o regione, diviso per il numero di parti di messaggio inviati ai destinatari in ciascuno di tali paesi e regioni.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>
Consegne	txn-sms-delivered	Il numero di messaggi consegnati ai destinatari.

Metrica	Kpi-name	Descrizione
Consegne, raggruppate per paese	txn-sms-delivered-grouped-by-country	<p>Il numero di messaggi consegnati ai destinatari per ogni paese o regione a cui sono stati inviati i messaggi.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>
Consegne, raggruppate per data	txn-sms-delivered-grouped-by-date	<p>Il numero di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Errori di recapito	txn-sms-error-distribution	<p>Il numero di volte in cui si è verificato un errore durante il tentativo di recapitare i messaggi, per ogni tipo di errore che si è verificato.</p> <p>I risultati della query per questo parametro sono raggruppati per codice di errore, per ogni tipo di errore che si è verificato.</p>

Metrica	Kpi-name	Descrizione
Percentuale di recapiti	<code>txn-sms-delivery-rate</code>	<p>La percentuale di messaggi consegnati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati.</p>
Percentuale di recapiti, raggruppati per data	<code>txn-sms-delivery-rate-grouped-by-date</code>	<p>La percentuale di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Parti del messaggio consegnate	<code>txn-sms-delivered-by-parts</code>	<p>Numero di parti di messaggio recapitate ai partecipanti.</p> <p>Una parte di messaggio è una sezione di un messaggio SMS. Se un messaggio SMS contiene più caratteri di quelli consentiti dal protocollo SMS, Amazon Pinpoint divide il messaggio nel numero di parti necessarie per l'invio del messaggio a un destinatario.</p>
Parti di messaggio recapitate, raggruppate per paese	<code>txn-sms-delivered-by-parts-grouped-by-country</code>	<p>Numero di parti di messaggio recapitate ai destinatari per ogni paese o regione a cui sono stati inviati i messaggi.</p> <p>Una parte di messaggio è una sezione di un messaggio SMS.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>

Metrica	Kpi-name	Descrizione
Parti di messaggio inviate	txn-sms-sent-by-parts	Numero di parti di messaggio inviate. Una parte di messaggio è una sezione di un messaggio SMS. Se un messaggio SMS contiene più caratteri di quelli consentiti dal protocollo SMS, Amazon Pinpoint divide il messaggio nel numero di parti necessarie per l'invio del messaggio a un destinatario.
Parti di messaggio inviate, raggruppate per paese	txn-sms-sent-by-parts-grouped-by-country	<p>Numero di parti messaggio inviate per ogni paese o regione a cui sono stati inviati i messaggi. Una parte di messaggio è una sezione di un messaggio SMS.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>
Messages sent (Messaggi inviati)	txn-sms-sent	Il numero di messaggi che sono stati inviati.

Metrica	Kpi-name	Descrizione
Messaggi inviati, raggruppati per paese	<code>txn-sms-sent-grouped-by-country</code>	<p>Il numero di messaggi inviati per ogni paese o regione a cui sono stati inviati i messaggi.</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>
Messaggi inviati, raggruppati per data	<code>txn-sms-sent-grouped-by-date</code>	<p>Il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Prezzo totale, raggruppato per paese	txn-sms-total-price-grouped-by-country	<p>Costo totale dell'invio dei messaggi, per ogni paese o regione a cui sono stati inviati i messaggi. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 =0,645¢=\$0,00645).</p> <p>I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.</p>

## Parametri della campagna

La tabella seguente elenca e descrive i parametri standard delle campagne su cui è possibile eseguire query per valutare le performance di una singola campagna. Per eseguire query sui dati per queste metriche, utilizza la risorsa [Parametri della campagna](#) dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro kpi-name nella query.

Metrica	Kpi-name	Descrizione
Bounce rate (Percentuale di mancati recapiti (bounce))	hard-bounce-rate	Per ogni esecuzione della campagna, la percentuale di messaggi e-mail che non è stato possibile recapitare ai destinatari. Questa metrica misura solo le occorrenze di mancato recapito permanent

Metrica	Kpi-name	Descrizione
		<p>e, ovvero i messaggi in cui l'indirizzo e-mail del destinatario presenta un problema permanente che impedisce il recapito del messaggio.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail non recapitati inviati da tutte le esecuzioni della campagna, diviso il numero di messaggi e-mail inviati da tutte quelle esecuzioni della campagna.</p>

Metrica	Kpi-name	Descrizione
Percentuale di mancati recapiti, raggruppati per esecuzione della campagna	<code>hard-bounce-rate-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, la percentuale di messaggi e-mail che non è stato possibile recapitare ai destinatari. Questa metrica misura solo le occorrenze di mancato recapito permanente, ovvero i messaggi in cui l'indirizzo e-mail del destinatario presenta un problema permanente che impedisce il recapito del messaggio.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail non recapitati inviati da un'esecuzione della campagna, diviso il numero di messaggi e-mail inviati dall'esecuzione della campagna.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
Percentuale di recapiti	successful-delivery-rate	<p>Per ogni esecuzione della campagna, la percentuale di messaggi recapitati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi che sono stati inviati da tutte le esecuzioni della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati da tutte quelle esecuzioni della campagna.</p>

Metrica	Kpi-name	Descrizione
La percentuale di recapiti, raggruppati per l'esecuzione della campagna	<code>successful-delivery-rate-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, la percentuale di messaggi recapitati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati da un'esecuzione della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati tramite l'esecuzione della campagna.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
Percentuale di recapiti, raggruppati per data	successful-delivery-rate-grouped-by-date	<p>Per tutte le esecuzioni della campagna, la percentuale di messaggi recapitati ai destinatari durante ogni giorno nell'intervallo di date specificato.</p> <p>Questo parametro viene calcolato come il numero di messaggi che sono stati inviati da tutte le esecuzioni della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati da tutte quelle esecuzioni della campagna, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>

Metrica	Kpi-name	Descrizione
Percentuale di e-mail aperte	email-open-rate	<p>Per tutte le esecuzioni della campagna, la percentuale di messaggi e-mail aperti dai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail inviati da tutte le esecuzioni della campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati da tutte quelle esecuzioni della campagna e recapitati ai destinatari.</p>

Metrica	Kpi-name	Descrizione
Percentuale di e-mail aperte, raggruppate per l'esecuzione della campagna	<code>email-open-rate-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, la percentuale di messaggi e-mail aperti dai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi e-mail inviati da un'esecuzione della campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati dall'esecuzione della campagna e recapitati ai destinatari.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
E-mail aperte, raggruppate per l'esecuzione della campagna	<code>direct-email-opens-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, il numero di messaggi e-mail aperti dai destinatari.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>
Consegne endpoint	<code>unique-deliveries</code>	Per tutte le esecuzioni della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.
Consegne endpoint, raggruppate per l'esecuzione della campagna	<code>unique-deliveries-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
Consegne endpoint, raggruppate per data	<code>unique-deliveries-grouped-by-date</code>	<p>Per tutte le esecuzioni della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi, per ogni giorno nell'intervallo di date specificato.</p> <p>I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.</p>
Collegamenti selezionati, raggruppati per l'esecuzione della campagna	<code>clicks-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, il numero di volte che i destinatari hanno fatto clic su collegamenti nel messaggio e-mail. Se un singolo destinatario ha fatto clic su più collegamenti nel messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
Messaggi recapitati, raggruppati per l'esecuzione della campagna	<code>successful-deliveries-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, il numero di messaggi recapitati ai destinatari.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati da una esecuzione della campagna, meno il numero di messaggi non recapitati ai destinatari dell'esecuzione a causa di un hard bounce.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
I messaggi inviati, raggruppati per l'esecuzione della campagna	attempted-deliveries-grouped-by-campaign-activity	<p>Per ogni esecuzione della campagna, il numero di messaggi che sono stati inviati.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (CampaignActivityId ), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>
Percentuale di notifiche push aperte	push-open-rate	<p>Per ogni esecuzione della campagna, la percentuale di notifiche push aperte dai destinatari.</p> <p>Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da tutte le esecuzioni della campagna e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate da tutte quelle esecuzioni della campagna e consegnate ai destinatari.</p>

Metrica	Kpi-name	Descrizione
Percentuale di notifiche push aperte, raggruppate per l'esecuzione della campagna	<code>push-open-rate-grouped-by-campaign-activity</code>	<p>Per ogni esecuzione della campagna, la percentuale di notifiche push aperte dai destinatari.</p> <p>Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da un'esecuzione della campagna e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate dall'esecuzione della campagna e consegnate ai destinatari.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività della campagna (<code>CampaignActivityId</code>), che è una stringa che identifica in modo univoco una esecuzione della campagna.</p>

Metrica	Kpi-name	Descrizione
Totale notifiche push aperte, raggruppate per l'esecuzione della campagna	<code>direct-push-opens-grouped-by-campaign-activity</code>	Per ogni esecuzione della campagna, il numero di notifiche push aperte dai destinatari.  I risultati della query per questo parametro sono raggruppati per ID attività della campagna ( <code>CampaignActivityId</code> ), che è una stringa che identifica in modo univoco una esecuzione della campagna.
Spesa totale SMS	<code>sms-spend</code>	Per tutte le campagne, l'importo totale, in millesimi di centesimo, speso per l'invio di SMS.

## Parametri di coinvolgimento del viaggio

La tabella seguente elenca e descrive le metriche di coinvolgimento del percorso standard per cui è possibile eseguire query per monitorare le tendenze di tutti i messaggi e-mail inviati da un percorso Amazon Pinpoint. Per eseguire query sui dati relativi a queste metriche, utilizza la risorsa [Metriche di coinvolgimento del percorso](#) dell'API Amazon Pinpoint. Nella tabella, la colonna `kpi-name` indica il valore da utilizzare per il parametro `kpi-name` in una query.

Metrica	Kpi-name	Descrizione
Clic	<code>journey-emails-clicked</code>	Il numero di volte che i partecipanti hanno fatto clic sui collegamenti nei messaggi. Se un singolo partecipante ha fatto clic su più collegamenti in un messaggio o ha fatto clic

Metrica	Kpi-name	Descrizione
		sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.
Clic, raggruppati per attività	emails-clicked-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di volte in cui i partecipanti hanno fatto clic sui link nei messaggi. Se un singolo partecipante ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (JourneyActivityId ), che è una stringa che identifica in modo univoco un'attività.</p>
Reclami	journey-emails-complained	Il numero di messaggi segnalati dai partecipanti come e-mail non richieste o indesiderate.

Metrica	Kpi-name	Descrizione
Reclami, raggruppati per attività	emails-complained-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi segnalati dai partecipanti come e-mail non richieste o indesiderate.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>
Consegne	journey-emails-delivered	<p>Il numero di messaggi consegnati ai partecipanti.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati.</p>

Metrica	Kpi-name	Descrizione
Consegne raggruppate per attività	emails-delivered-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi che sono stati recapitati ai partecipanti.</p> <p>Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati, per ogni attività nel viaggio.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>
Mancati recapiti permanenti	journey-emails-hardbounced	<p>Il numero di messaggi che non è stato possibile consegnare ai partecipanti a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del partecipante non esiste.</p>

Metrica	Kpi-name	Descrizione
Hard bounce, raggruppati per attività	emails-hardbounced-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi che non possono essere recapitati ai partecipanti a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del partecipante non esiste.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>
Aperture	journey-emails-opened	Mostra il numero di messaggi che sono stati aperti dai partecipanti:
Aperture, raggruppate per attività	emails-opened-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi che sono stati aperti dai partecipanti.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>

Metrica	Kpi-name	Descrizione
Rifiuti	<code>journey-emails-rejected</code>	Il numero di messaggi non inviati ai partecipanti perché rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.
Rifiuti, raggruppati per attività	<code>emails-rejected-grouped-by-journey-activity</code>	<p>Per ogni attività del viaggio, il numero di messaggi che non sono stati inviati ai partecipanti perché sono stati rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>
Invii	<code>journey-emails-sent</code>	Il numero di messaggi che sono stati inviati.

Metrica	Kpi-name	Descrizione
Invia, raggruppati per attività	emails-sent-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi inviati.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (<code>JourneyActivityId</code>), che è una stringa che identifica in modo univoco un'attività.</p>
Soft bounce	journey-emails-softbounced	<p>Il numero di messaggi che non è stato possibile consegnare ai partecipanti a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo o impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del partecipante è piena o il server di ricezione è temporaneamente non disponibile.</p>

Metrica	Kpi-name	Descrizione
Soft bounce, raggruppati per attività	emails-softbounced-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di messaggi che non è stato possibile recapitare e ai partecipanti a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo o impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del partecipante è piena o il server di ricezione è temporaneamente non disponibile.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (JourneyActivityId ), che è una stringa che identifica in modo univoco un'attività.</p>
Annullamenti di sottoscrizioni	journey-emails-unsubscribed	<p>Il numero di volte in cui i partecipanti hanno fatto clic sui link di annullamento della sottoscrizione nei messaggi. Se un singolo partecipante ha fatto clic sullo stesso link di annullamento della sottoscrizione più di una volta, ogni clic è incluso nel conteggio.</p>

Metrica	Kpi-name	Descrizione
Annullamenti delle sottoscrizioni, raggruppati per attività	emails-unsubscribed-grouped-by-journey-activity	<p>Per ogni attività del viaggio, il numero di volte in cui i partecipanti hanno fatto clic sui link di annullamento della sottoscrizione nei messaggi. Se un singolo partecipante ha fatto clic sullo stesso link di annullamento della sottoscrizione più di una volta, ogni clic è incluso nel conteggio.</p> <p>I risultati della query per questo parametro sono raggruppati per ID attività (JourneyActivityId ), che è una stringa che identifica in modo univoco un'attività.</p>

## Parametri di esecuzione del viaggio

La tabella seguente elenca e descrive le metriche di esecuzione standard su cui è possibile eseguire query per valutare lo stato dei partecipanti in un percorso Amazon Pinpoint. Per eseguire query sui dati per questi metriche, utilizza la risorsa [Metriche di esecuzione percorso](#) dell'API Amazon Pinpoint. Nella colonna Field (Campo) nella tabella viene identificato il nome del campo visualizzato nei risultati della query per ogni parametro.

Metrica	Campo	Descrizione
Partecipanti attivi	ENDPOINT_ACTIVE	<p>Il numero di partecipanti che stanno procedendo attivamente nelle attività del viaggio.</p> <p>Questo parametro è calcolato come il numero di partecipanti che hanno iniziato il viaggio,</p>

Metrica	Campo	Descrizione
		meno il numero di partecipanti che hanno lasciato il viaggio e il numero di partecipanti che sono stati rimossi dal viaggio.
Annullamenti dei partecipanti	CANCELLED	Il numero di partecipanti che non hanno completato il viaggio perché il viaggio è stato annullato.
Partenze partecipanti	ENDPOINT_LEFT	Il numero di partecipanti che hanno abbandonato il viaggio.
Ingressi partecipanti	ENDPOINT_ENTERED	Il numero di partecipanti che hanno iniziato il viaggio.
Eccezioni dei partecipanti, limiti di rientro	REENTRY_CAP_EXCEEDED	Il numero di partecipanti che non hanno completato il viaggio perché avrebbero superato il numero massimo di volte in cui un singolo partecipante può rientrare nel viaggio.

Metrica	Campo	Descrizione
Eccezioni dei partecipanti, rifiuti	ACTIVE_ENDPOINT_REJECTED	<p>Il numero di partecipanti che non possono iniziare il viaggio perché sono già partecipanti attivi al viaggio.</p> <p>Un partecipante viene rifiutato se inizia un viaggio e in seguito la sua definizione di endpoint viene aggiornata in modo tale da compromettere la sua inclusione in un segmento (in base ai criteri del segmento) o nel viaggio (in base alle condizioni di attività).</p>

## Parametri di esecuzione delle attività di viaggio

La tabella seguente elenca e descrive le metriche di esecuzione standard su cui è possibile eseguire query per valutare lo stato dei partecipanti in ogni tipo di attività individuale in un percorso Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa [Metriche di esecuzione delle attività del percorso](#) dell'API Amazon Pinpoint. Nella colonna Metrics (Parametri) della tabella sono elencati i campi visualizzati nei risultati della query per ogni tipo di attività. Fornisce anche una breve descrizione di ciascun campo.

Tipo di attività	Metriche
Divisione/Nessuna divisione (CONDITIONAL_SPLIT )	<p>I parametri sono:</p> <ul style="list-style-type: none"> <li>Branch_FALSE _ numero di partecipanti che non hanno rispettato le condizioni dell'attività e che hanno continuato con l'attività sul percorso "No".</li> <li>Branch_TRUE : numero di partecipanti che hanno rispettato le condizioni dell'attività e</li> </ul>

Tipo di attività	Metriche
	<p>hanno continuato con l'attività sul percorso "Sì".</p> <p>I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.</p>
Controllo (HOLDOUT)	<p>I parametri sono:</p> <ul style="list-style-type: none"><li>• HOLDOUT: numero di partecipanti rimossi dal percorso come parte della percentuale di controllo per l'attività.</li><li>• PASSED: numero di partecipanti trasferiti all'attività successiva nel percorso.</li></ul>

Tipo di attività	Metriche
E-mail (MESSAGE)	<p>I parametri sono:</p> <ul style="list-style-type: none"><li>• <b>DAILY_CAP_EXCEEDED</b> : numero di messaggi non inviati perché hanno superato il numero massimo di messaggi che un singolo partecipante può ricevere durante un periodo di 24 ore.</li><li>• <b>FAILURE_PERMANENT</b> : numero di messaggi non inviati a causa di un problema permanente.</li><li>• <b>QUIET_TIME</b> : numero di messaggi non inviati perché sono stati recapitati durante il periodo di riposo per il fuso orario del partecipante.</li><li>• <b>SERVICE_FAILURE</b> : numero di messaggi non inviati a causa di un problema in Amazon Pinpoint.</li><li>• <b>SUCCESS</b>: numero di messaggi recapitati correttamente ai partecipanti.</li><li>• <b>THROTTLED</b> : numero di messaggi non inviati perché il relativo invio ha superato la quota di invio dell'account Amazon Pinpoint..</li><li>• <b>TRANSIENT_FAILURE</b> : numero di messaggi non inviati a causa di un problema temporaneo.</li><li>• <b>UNKNOWN</b>: numero di messaggi non inviati a causa di un problema sconosciuto.</li></ul>

Tipo di attività	Metriche
Divisione multivariata (MULTI_CONDITIONAL_SPLIT )	<p>Per ogni percorso dell'attività, il numero dei partecipanti che hanno proceduto all'attività sul percorso.</p> <p>I risultati della query per questa metrica sono raggruppati per percorso, <code>Branch_#</code> dove <code>#</code> è l'identificatore numerico di un percorso, ad esempio <code>Branch_1</code> per il primo percorso dell'attività.</p> <p>I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.</p>
Divisione casuale (RANDOM_SPLIT )	<p>Per ogni percorso dell'attività, il numero dei partecipanti che hanno proceduto all'attività sul percorso.</p> <p>I risultati della query per questa metrica sono raggruppati per percorso, <code>Branch_#</code> dove <code>#</code> è l'identificatore numerico di un percorso, ad esempio <code>Branch_1</code> per il primo percorso dell'attività.</p> <p>I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.</p>

Tipo di attività	Metriche
Attesa (WAIT)	<p>I parametri sono:</p> <ul style="list-style-type: none"><li>• <b>WAIT_FINISHED</b> : numero di partecipanti che hanno terminato l'attesa per il periodo di tempo specificato.</li><li>• <b>WAIT_SKIPPED</b> : numero di partecipanti che non hanno aspettato il periodo di tempo specificato, in genere perché hanno iniziato l'attività o il percorso dopo l'ora di fine pianificata per l'attività.</li><li>• <b>WAIT_STARTED</b> : numero di partecipanti che hanno iniziato ad aspettare e che non hanno saltato o terminato l'attesa per il periodo di tempo specificato.</li></ul>

Tipo di attività	Metriche
Contact Center (CONTACT_CENTER )	<p>I parametri sono:</p> <ul style="list-style-type: none"><li>• <b>CALL_QUEUED</b> : numero di partecipanti che sono stati chiamati e messi in coda su Amazon Connect. Include i tentativi di ricomposizione.</li><li>• <b>CONTINUE_WAITING</b> : numero di partecipanti che continuano ad attendere che vengano effettuati i tentativi di chiamata.</li><li>• <b>DIAL_FAILURE</b> : numero di partecipanti con tentativi giornalieri non riusciti.</li><li>• <b>DROPPED</b>: numero di partecipanti che non soddisfano più le condizioni definite nelle precedenti attività del percorso al momento della partenza.</li><li>• <b>TIMEOUT</b>: numero di partecipanti che non hanno ricevuto alcun codice di disposizione di Amazon Connect dopo diversi tentativi di chiamata.</li><li>• <b>WAIT_FINISHED</b> : numero di partecipanti che hanno terminato l'attesa per il periodo di tempo specificato.</li><li>• <b>WAIT_FOR_QUIET_HOURS</b> : numero di partecipanti che attendono il termine del periodo di silenzio per poter effettuare il recapito sul canale.</li><li>• <b>WAIT_STARTED</b> : numero di partecipanti che hanno iniziato ad aspettare e che non hanno saltato o terminato l'attesa per il periodo di tempo specificato.</li></ul>

## Metriche relative all'esecuzione di percorsi e campagne

Puoi eseguire query sulle metriche di esecuzione standard per valutare lo stato dei partecipanti a ogni tipo di attività individuale per un percorso o una campagna Amazon Pinpoint. Per eseguire query sui dati relativi a queste metriche, utilizza la risorsa [Parametri di esecuzione attività percorso](#) o [Parametri della campagna](#) dell'API Amazon Pinpoint. Nella seguente tabella sono elencati i campi visualizzati nei risultati della query per ogni tipo di attività.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
ENDPOINT_PRODUCED	Entrambi	Numero di endpoint inizialmente prodotti dal segmento o dall'evento prima dell'applicazione di qualsiasi filtro.
ENDPOINTS_FROM_USER	Entrambi	Se il cliente ha un segmento dedicato solo agli ID utente, verranno aggiunti tutti gli endpoint di tali utenti. Questa metrica misura il numero di endpoint aggiunti in questo modo.
ENDPOINT_OPT_OUT	Entrambi	L'endpoint è stato escluso mediante opt-out e non è stato inserito nella campagna o nel percorso.
ENDPOINT_INACTIVE	Entrambi	L'endpoint è inattivo e non rientra nella campagna o nel percorso.
FILTERED_OUT_BY_SEGMENT	Entrambi	L'endpoint non corrisponde ai filtri dei segmenti e non rientra nella campagna o nel percorso.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
ENDPOINT_MISSING_ADDRESS	Entrambi	All'endpoint manca un indirizzo e non rientra nella campagna o nel percorso.
ENDPOINT_MISSING_CHANNEL	Entrambi	All'endpoint manca un canale e non rientra nella campagna o nel percorso.
ENDPOINT_MISSING_TIMEZONE	Entrambi	All'endpoint manca un valore per il fuso orario ed è stato escluso. Ciò accade solo quando è richiesto un valore del fuso orario.
ENDPOINT_TIMEZONE_MISMATCH	Entrambi	L'endpoint si trova in un fuso orario attualmente non incluso nell'esecuzione.
ENDPOINT_CHANNEL_MISMATCH	Campagne	La campagna non ha un messaggio configurato per il tipo di canale dell'endpoint specifico.
DUPLICATE_ENDPOINT	Entrambi	Sono stati rilevati e deduplicati gli endpoint duplicati.
DUPLICATE_USER	Entrambi	Sono stati rilevati e deduplicati gli utenti duplicati da un segmento di soli ID utente. Se hanno lo stesso ID utente, verrà restituita una metrica pari a 1.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
PAUSED	Percorsi	Metrica rimossa dall'esecuzione perché il percorso è stato sospeso.
ENDED	Percorsi	Metrica rimossa dall'esecuzione perché il percorso è terminato.
TREATMENT_HOLDOUT	Campagne	Metrica restituita nelle campagne A/B, per gli endpoint le cui coorti non corrispondono al trattamento attuale. Ad esempio, in una suddivisione A/B 50/50, il 50% degli endpoint restituirà questa metrica per ogni trattamento
ENDPOINT_ESTIMATED_TIMEZONE	Percorsi	La stima del fuso orario è stata in grado di stimare un fuso orario per l'endpoint.

## Esecuzione di query sui dati di analisi di Amazon Pinpoint per le campagne

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, è possibile utilizzare le API di analisi di Amazon Pinpoint per eseguire query sui dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni dettagliate sulle tendenze di recapito e coinvolgimento per le campagne.

Ognuno di questi parametri è un valore misurabile, noto anche come indicatore di prestazioni chiave (KPI), che consente di monitorare e valutare le prestazioni di una o più campagne. Ad esempio, è possibile utilizzare un parametro di campagna per accedere ai dati relativi al numero di endpoint ai quali è stato inviato un messaggio della campagna o quanti di questi messaggi sono stati recapitati agli endpoint previsti.

Amazon Pinpoint raccoglie e aggrega automaticamente questi dati per tutte le campagne. Memorizza i dati per 90 giorni. Se è stata integrata un'app per dispositivi mobili a Amazon Pinpoint tramite SDK mobile AWS, Amazon Pinpoint estende questo supporto per includere ulteriori metriche, ad esempio la percentuale di notifiche push aperte dai destinatari. Per informazioni sull'integrazione di un'app per dispositivi mobili, consulta [Integrazione di Amazon Pinpoint con le applicazioni](#).

Se utilizzi le API di analisi di Amazon Pinpoint per eseguire query sui dati, puoi scegliere varie opzioni che definiscono l'ambito, i dati, il raggruppamento e i filtri per la query. A tale scopo, utilizzare i parametri che specificano il progetto, la campagna e il parametro su cui eseguire una query, oltre a eventuali filtri basati sulla data che si desidera applicare.

In questo argomento vengono illustrati e forniti esempi di come scegliere queste opzioni ed eseguire query di dati di una o più campagne.

## Prerequisiti

Prima di eseguire query sui dati di analisi di una o più campagne, consente di raccogliere le informazioni seguenti, che sono utilizzate per definire la query:

- **ID progetto:** identificatore univoco per il progetto associato alla campagna o alle campagne. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `application-id`. Nella console Amazon Pinpoint, questo valore viene visualizzato nel campo ID progetto nella pagina Tutti i progetti.
- **ID campagna:** l'identificatore univoco per la campagna, se desideri eseguire query e i dati solo per una campagna. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `campaign-id`. Questo valore non viene visualizzato sulla console.
- **Intervallo di date:** facoltativamente, la prima e l'ultima data e l'ora dell'intervallo di date in cui eseguire query sui dati. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, devono iniziare da meno di 90 giorni rispetto al giorno corrente. Se non viene specificato un intervallo di date, Amazon Pinpoint esegue automaticamente le query sui dati dei 31 giorni di calendario precedenti.
- **Tipo di metrica:** tipo di metrica a cui inviare le query. Esistono due tipi di parametri, parametri delle applicazioni e parametri delle campagne. Un parametro dell'applicazione fornisce i dati per tutte le campagne associate a un progetto, denominate anche applicazione. Un parametro della campagna fornisce dati per una sola campagna.
- **Metrica:** nome della metrica a cui inviare le query, in particolare il valore `kpi-name` della metrica. Per un elenco completo dei parametri supportati e per il valore `kpi-name` di ognuno, consulta [Parametri standard](#).

Aiuta anche a stabilire se si desidera raggruppare i dati in base a un campo pertinente. In tal caso, è possibile semplificare l'analisi e la creazione di report scegliendo un parametro progettato per raggruppare automaticamente i dati. Ad esempio, in Amazon Pinpoint sono disponibili diverse metriche standard che riportano la percentuale di messaggi recapitati ai destinatari di una campagna. Uno di questi parametri raggruppa automaticamente i dati per data (`successful-delivery-rate-grouped-by-date`). Un altro parametro raggruppa automaticamente i dati per esecuzione di campagna (`successful-delivery-rate-grouped-by-campaign-activity`). Una terza metrica restituisce semplicemente un unico valore, ovvero la percentuale di messaggi recapitati ai destinatari da tutte le esecuzioni delle campagne (`successful-delivery-rate`).

Se non è possibile trovare un parametro standard che raggruppi i dati nel modo desiderato, è possibile sviluppare una serie di query in grado di restituire i dati desiderati e quindi suddividere manualmente o combinare i risultati della query in gruppi personalizzati progettati.

Infine, è importante verificare di essere autorizzati ad accedere ai dati su cui si desidera eseguire una query. Per ulteriori informazioni, consulta [Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint](#).

## Esecuzione di query sui dati di una campagna

Per eseguire una query sui dati di una campagna, è possibile utilizzare l'API [Parametri della campagna](#) e specificare i valori per i seguenti parametri richiesti:

- `application-id`: ID del progetto, ovvero l'identificatore univoco del progetto associato alla campagna. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- `campaign-id`: identificatore univoco della campagna.
- `kpi-name`: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name` di ognuno, consulta [Parametri standard](#).

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019. Gli intervalli di date includono gli

estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Negli esempi seguenti viene illustrato come eseguire query sui dati di analisi per una campagna tramite la REST API Amazon Pinpoint, l'AWS CLI e AWS SDK for Java. È possibile utilizzare qualsiasi SDK AWS supportato per eseguire query sui dati di analisi di una campagna. Gli esempi AWS CLI sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata (\).

## REST API

Per eseguire query sui dati di analisi per una campagna tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI [Parametri della campagna](#). Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint* è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto associato alla campagna.
- *application-id* è l'identificatore univoco del progetto associato alla campagna.
- *campaign-id* è l'identificatore univoco della campagna.
- *kpi-name* è il valore kpi-name per il parametro da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori `start-time` e `end-time` all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `1234567890123456789012345example` è l'identificatore univoco del progetto associato alla campagna.
- `80b8efd84042ff8d9c96ce2f8example` è l'identificatore univoco per la campagna.
- `unique-deliveries` è il valore `kpi-name` del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## AWS CLI

Per eseguire query sui dati di analisi di una campagna utilizzando l'AWS CLI, utilizzare il comando `get-campaign-date-range-kpi` e specificare i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^  
  --application-id application-id ^  
  --campaign-id campaign-id ^  
  --kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco del progetto associato alla campagna.
- *campaign-id* è l'identificatore univoco della campagna.
- *kpi-name* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint

univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
--application-id 1234567890123456789012345example ^
--campaign-id 80b8efd84042ff8d9c96ce2f8example ^
--kpi-name unique-deliveries ^
--start-time 2019-07-19T00:00:00Z ^
--end-time 2019-07-26T23:59:59Z
```

Dove:

- 1234567890123456789012345example è l'identificatore univoco del progetto associato alla campagna.
- 80b8efd84042ff8d9c96ce2f8example è l'identificatore univoco per la campagna.
- unique-deliveries è il valore kpi-name del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- 2019-07-19T00:00:00Z rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- 2019-07-26T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## SDK for Java

Per eseguire query sui dati di analisi di una campagna con AWS SDK for Java, utilizzare il metodo `GetCampaignDateRangeKpiRequest` dell'API [Parametri della campagna](#). Specificare i valori appropriati per i parametri richiesti:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withCampaignId("campaignId")
    .withKpiName("kpiName")
```

Dove:

- *applicationId* è l'identificatore univoco del progetto associato alla campagna.
- *campaignId* è l'identificatore univoco della campagna.

- *KPIname* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
    .withApplicationId("1234567890123456789012345example")
    .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
    .withKpiName("unique-deliveries")
    .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
    .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dove:

- `1234567890123456789012345example` è l'identificatore univoco del progetto associato alla campagna.
- `80b8efd84042ff8d9c96ce2f8example` è l'identificatore univoco per la campagna.
- `unique-deliveries` è il valore `kpi-name` del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Ad esempio, la metrica della campagna Recapiti agli endpoint (`unique-deliveries`), utilizzata negli esempi precedenti, restituisce un valore, ovvero il numero di endpoint univoci a cui sono stati recapitati i messaggi, per tutte le esecuzioni di una campagna. In questo caso, la risposta JSON è la seguente:

```
{
```

```
"CampaignDateRangeKpiResponse":{
  "ApplicationId":"1234567890123456789012345example",
  "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
  "EndTime":"2019-07-26T23:59:59Z",
  "KpiName":"unique-deliveries",
  "KpiResult":{
    "Rows":[
      {
        "Values":[
          {
            "Key":"UniqueDeliveries",
            "Type":"Double",
            "Value":"123.0"
          }
        ]
      }
    ]
  },
  "StartTime":"2019-07-19T00:00:00Z"
}
```

Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta [Utilizzo dei risultati delle query](#).

## Interrogazione dei dati per più campagne

È possibile eseguire query sui dati per più campagne in due modi. Il modo migliore dipende dal fatto che si desidera eseguire una query sui dati per le campagne che sono tutte associate allo stesso progetto. In tal caso, dipende anche dal fatto che si desidera eseguire una query sui dati per tutte o solo per un sottoinsieme di tali campagne.

Per eseguire una query sui dati delle campagne associate a progetti diversi o solo sui dati di un sottoinsieme delle campagne associate allo stesso progetto, l'approccio migliore consiste nel creare ed eseguire una serie di query individuali, una per ogni campagna sui cui dati si desidera eseguire una query. Nella sezione precedente viene illustrato come eseguire una query sui dati per una sola campagna.

Per eseguire una query sui dati di tutte le campagne associate allo stesso progetto, è possibile utilizzare l'API [Parametri applicazioni](#). Specificare i valori per i parametri obbligatori seguenti.

- **application-id**: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- **kpi-name**: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name` di ognuno, consulta [Parametri standard](#).

È inoltre possibile filtrare i dati in base all'intervallo di date. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Negli esempi seguenti viene illustrato come eseguire query sui dati di analisi per una campagna tramite la REST API Amazon Pinpoint, l'AWS CLI e AWS SDK for Java. È possibile utilizzare qualsiasi SDK AWS supportato per eseguire query sui dati di analisi di una campagna. Gli esempi AWS CLI sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata (\).

## REST API

Per eseguire query sui dati di analisi per più campagne utilizzando la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI [Metriche delle applicazioni](#). Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- ***endpoint*** è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto associato alle campagne.
- ***application-id*** è l'identificatore univoco per il progetto associato alle campagne.

- *kpi-name* è il valore kpi-name per il parametro da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori `start-time` e `end-time` all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/  
kpis/daterange/unique-deliveries-grouped-by-campaign?start-  
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `1234567890123456789012345example` è l'identificatore univoco per il progetto associato alle campagne.
- `unique-deliveries-grouped-by-campaign` è il valore kpi-name per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## AWS CLI

Per eseguire query sui dati di analisi di più campagne utilizzando l'AWS CLI, utilizzare il comando `get-application-date-range-kpi` e specificare i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^
```

```
--application-id application-id ^  
--kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco per il progetto associato alle campagne.
- *kpi-name* è il valore kpi-name per il parametro da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id 1234567890123456789012345example ^  
  --kpi-name unique-deliveries-grouped-by-campaign ^  
  --start-time 2019-07-19T00:00:00Z ^  
  --end-time 2019-07-26T23:59:59Z
```

Dove:

- 1234567890123456789012345example è l'identificatore univoco del progetto associato alla campagna.
- unique-deliveries-grouped-by-campaign è il valore kpi-name per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- 2019-07-19T00:00:00Z rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- 2019-07-26T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## SDK for Java

Per eseguire query sui dati analitici per più campagne con AWS SDK for Java, utilizzare il metodo `GetApplicationDateRangeKpiRequest` dell'API [Parametri applicazioni](#). Specificare i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

Dove:

- *applicationId* è l'identificatore univoco del progetto associato alle campagne.
- *KPIname* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("1234567890123456789012345example")
    .withKpiName("unique-deliveries-grouped-by-campaign")
    .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
    .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dove:

- `1234567890123456789012345example` è l'identificatore univoco per il progetto associato alle campagne.
- `unique-deliveries-grouped-by-campaign` è il valore `kpi-name` per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.

- 2019-07-26T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Altri parametri restituiscono più valori e quei valori sono raggruppati in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Ad esempio, la metrica Recapiti agli endpoint, raggruppati per campagna (`unique-deliveries-grouped-by-campaign`), utilizzata negli esempi precedenti, restituisce più valori, ovvero il numero di endpoint univoci a cui sono stati recapitati i messaggi, per ogni campagna associata a un progetto. In questo caso, la risposta JSON è la seguente:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-07-26T23:59:59Z",
    "KpiName":"unique-deliveries-grouped-by-campaign",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"80b8efd84042ff8d9c96ce2f8example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"123.0"
            }
          ]
        }
      ],
      {
        "GroupedBy":[
          {
            "Key":"CampaignId",
```

```
        "Type": "String",
        "Value": "810c7aab86d42fb2b56c8c966example"
    }
],
"Values": [
    {
        "Key": "UniqueDeliveries",
        "Type": "Double",
        "Value": "456.0"
    }
]
},
{
    "GroupedBy": [
        {
            "Key": "CampaignId",
            "Type": "String",
            "Value": "42d8c7eb0990a57ba1d5476a3example"
        }
    ],
    "Values": [
        {
            "Key": "UniqueDeliveries",
            "Type": "Double",
            "Value": "789.0"
        }
    ]
}
],
"StartTime": "2019-07-19T00:00:00Z"
}
}
```

In questo caso, il campo `GroupedBy` indica che i valori sono raggruppati per ID campagna (`CampaignId`).

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta [Utilizzo dei risultati delle query](#).

# Esecuzione di query sui dati di analisi Amazon Pinpoint per i messaggi transazionali

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, è possibile utilizzare le API di analisi di Amazon Pinpoint per eseguire query sui dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni dettagliate sulle tendenze di recapito e coinvolgimento per i messaggi transazionali inviati a un progetto.

Ognuno di questi parametri è un valore misurabile, noto anche come indicatore di prestazioni chiave (KPI), che consente di monitorare e valutare le prestazioni di questi messaggi transazionali. Ad esempio, è possibile utilizzare una metrica per scoprire il numero di messaggi e-mail o SMS transazionali inviati o quanti di questi messaggi sono stati recapitati ai destinatari. Amazon Pinpoint raccoglie e aggrega automaticamente questi dati per tutti i messaggi e-mail e SMS transazionali inviati per un progetto. Memorizza i dati per 90 giorni.

Se utilizzi le API di analisi di Amazon Pinpoint per eseguire query sui dati, puoi scegliere varie opzioni che definiscono l'ambito, i dati, il raggruppamento e i filtri per la query. A tale scopo, utilizza i parametri che specificano il progetto e il parametro su cui eseguire una query, oltre a eventuali filtri basati sulla data che desideri applicare.

Questo argomento illustra e fornisce esempi di come scegliere queste opzioni ed eseguire query sui dati di messaggistica transazionale per un progetto.

## Prerequisiti

Prima di eseguire query sui dati di analisi per i messaggi transazionali, consente di raccogliere le informazioni seguenti, che vengono utilizzate per definire la query:

- **ID progetto:** identificatore univoco per il progetto da cui sono stati inviati i messaggi. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `application-id`. Nella console Amazon Pinpoint, questo valore viene visualizzato nel campo ID progetto nella pagina Tutti i progetti.
- **Intervallo di date:** facoltativamente, la prima e l'ultima data e l'ora dell'intervallo di date in cui eseguire query sui dati. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, devono iniziare da meno di 90 giorni rispetto al giorno corrente. Se non viene specificato un intervallo di date, Amazon Pinpoint esegue automaticamente le query sui dati dei 31 giorni di calendario precedenti.

- **Mettrica:** nome della mettrica a cui inviare le query, in particolare il valore `kpi-name` della mettrica. Per un elenco completo dei parametri supportati e per il valore `kpi-named` di ognuno, consulta [Parametri standard](#).

Aiuta anche a stabilire se si desidera raggruppare i dati in base a un campo pertinente. In tal caso, è possibile semplificare l'analisi e la creazione di report scegliendo un parametro progettato per raggruppare automaticamente i dati. Ad esempio, in Amazon Pinpoint sono disponibili diverse metriche standard che restituiscono il numero di messaggi SMS transazionali recapitati ai destinatari. Uno di questi parametri raggruppa automaticamente i dati per data (`txn-sms-delivered-grouped-by-date`). Un altro parametro raggruppa automaticamente i dati per paese o regione (`txn-sms-delivered-grouped-by-country`). Una terza mettrica restituisce semplicemente un unico valore, ovvero il numero di messaggi recapitati ai destinatari (`txn-sms-delivered`). Se non è possibile trovare un parametro standard che raggruppi i dati nel modo desiderato, è possibile sviluppare una serie di query in grado di restituire i dati desiderati e quindi suddividere manualmente o combinare i risultati della query in gruppi personalizzati progettati.

Infine, è importante verificare di essere autorizzati ad accedere ai dati su cui si desidera eseguire una query. Per ulteriori informazioni, consulta [Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint](#).

## Query dei dati per i messaggi e-mail transazionali

Per eseguire una query sui dati relativi ai messaggi e-mail transazionali inviati per un progetto, è possibile utilizzare l'API [Parametri applicazioni](#) e specificare i valori per i seguenti parametri richiesti:

- `application-id`: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- `kpi-name`: nome della mettrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-named` di ognuno, consulta [Parametri standard](#).

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio

2019-09-06T20:00:00Z per le 20:00 UTC del 6 settembre 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Negli esempi seguenti viene illustrato come eseguire query sui dati di analisi per i messaggi e-mail transazionali tramite la REST API Amazon Pinpoint, l'AWS CLI e AWS SDK for Java. È possibile utilizzare qualsiasi SDK AWS supportato per eseguire query sui dati di analisi dei messaggi transazionali. Gli esempi AWS CLI sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata (\).

## REST API

Per eseguire query sui dati di analisi per i messaggi e-mail transazionali tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI [Metriche delle applicazioni](#). Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint* è l'endpoint Amazon Pinpoint della regione AWS che ospita il progetto.
- *application-id* è l'identificatore univoco del progetto.
- *kpi-name* è il valore kpi-name per il parametro da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori start-time e end-time all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/  
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-  
time=2019-09-13T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `1234567890123456789012345example` è l'identificatore univoco per il progetto.
- `txn-emails-sent` è il valore `kpi-name` del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-13T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## AWS CLI

Per eseguire query sui dati di analisi per i messaggi e-mail transazionali utilizzando il comando AWS CLI, utilizzare il comando `get-application-date-range-kpi` e specificare i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id application-id ^  
  --kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco del progetto.
- *kpi-name* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id 1234567890123456789012345example ^  
  --kpi-name txn-emails-sent ^  
  --start-time 2019-09-06T00:00:00Z ^  
  --end-time 2019-09-13T23:59:59Z
```

**Dove:**

- 1234567890123456789012345example è l'identificatore univoco per il progetto.
- txn-emails-sent è il valore kpi-name del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- 2019-09-06T00:00:00Z rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- 2019-09-13T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

**SDK for Java**

Per eseguire query sui dati analitici per i messaggi e-mail transazionali con AWS SDK for Java, utilizzare il metodo `GetApplicationDateRangeKpiRequest` dell'API [Parametri applicazioni](#). Specificare i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

**Dove:**

- *applicationId* è l'identificatore univoco per il progetto.
- *KPIName* è il valore kpi-name per il parametro da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("1234567890123456789012345example")
    .withKpiName("txn-emails-sent")
    .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
    .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

**Dove:**

- 1234567890123456789012345example è l'identificatore univoco per il progetto.
- txn-emails-sent è il valore kpi-name del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- 2019-09-06T00:00:00Z rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- 2019-09-13T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Ad esempio, la metrica Invii (txn-emails-sent), utilizzata negli esempi precedenti, restituisce un valore, ovvero il numero di messaggi e-mail transazionali inviati da un progetto. In questo caso, la risposta JSON è la seguente:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-09-13T23:59:59Z",
    "KpiName":"txn-emails-sent",
    "KpiResult":{
      "Rows":[
        {
          "Values":[
            {
              "Key":"TxnEmailsSent",
              "Type":"Double",
              "Value":"62.0"
            }
          ]
        }
      ]
    },
    "StartTime":"2019-09-06T00:00:00Z"
  }
}
```

Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta [Utilizzo dei risultati delle query](#).

## Query dei dati per i messaggi SMS transazionali

Per eseguire una query sui dati relativi ai messaggi SMS transazionali inviati per un progetto, è possibile utilizzare l'API [Parametri applicazioni](#) e specificare i valori per i seguenti parametri richiesti:

- **application-id**: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- **kpi-name**: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-namedi` di ognuno, consulta [Parametri standard](#).

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-09-06T20:00:00Z` per le 20:00 UTC del 6 settembre 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Negli esempi seguenti viene illustrato come eseguire query sui dati di analisi per i messaggi SMS transazionali tramite la REST API Amazon Pinpoint, l'AWS CLI e AWS SDK for Java. È possibile utilizzare qualsiasi SDK AWS supportato per eseguire query sui dati di analisi dei messaggi transazionali. Gli esempi AWS CLI sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata (\).

### REST API

Per eseguire query sui dati di analisi per i messaggi SMS transazionali tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI [Metriche delle applicazioni](#). Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint* è l'endpoint Amazon Pinpoint della regione AWS che ospita il progetto.
- *application-id* è l'identificatore univoco del progetto.
- *kpi-name* è il valore *kpi-name* per il parametro da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori *start-time* e *end-time* all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-time=2019-09-08T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `1234567890123456789012345example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore *kpi-name* del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## AWS CLI

Per eseguire query sui dati di analisi per i messaggi SMS transazionali utilizzando il comando AWS CLI, utilizzare il comando `get-application-date-range-kpi` e specificare i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^
  --application-id application-id ^
  --kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco del progetto.
- *kpi-name* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
  --application-id 1234567890123456789012345example ^
  --kpi-name txn-sms-sent-grouped-by-date ^
  --start-time 2019-09-06T00:00:00Z ^
  --end-time 2019-09-08T23:59:59Z
```

Dove:

- `1234567890123456789012345example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore `kpi-name` del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

## SDK for Java

Per eseguire query sui dati di analisi per i messaggi SMS transazionali utilizzando AWS SDK for Java, utilizzare il metodo `GetApplicationDateRangeKpiRequest` dell'API [Parametri applicazione](#) e specificare i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

Dove:

- *applicationId* è l'identificatore univoco per il progetto.
- *KPIName* è il valore `kpi-name` per il parametro da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("1234567890123456789012345example")
    .withKpiName("txn-sms-sent-grouped-by-date")
    .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
    .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

Dove:

- `1234567890123456789012345example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore `kpi-name` del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Ad esempio, la metrica `Invi`, raggruppati per data (`txn-sms-sent-grouped-by-date`), utilizzata negli esempi precedenti, restituisce più valori, ovvero il numero di messaggi SMS transazionali inviati durante ogni giorno dell'intervallo di date specificato. In questo caso, la risposta JSON è la seguente:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-09-08T23:59:59Z",
    "KpiName":"txn-sms-sent-grouped-by-date",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"Date",
              "Type":"String",
              "Value":"2019-09-06"
            }
          ],
          "Values":[
            {
              "Key":"TxnSmsSent",
              "Type":"Double",
              "Value":"29.0"
            }
          ]
        },
        {
          "GroupedBy":[
            {
              "Key":"Date",
              "Type":"String",
              "Value":"2019-09-07"
            }
          ],
          "Values":[
            {
```

```
        "Key": "TxnSmsSent",
        "Type": "Double",
        "Value": "35.0"
      }
    ],
  },
  {
    "GroupedBy": [
      {
        "Key": "Date",
        "Type": "String",
        "Value": "2019-09-08"
      }
    ],
    "Values": [
      {
        "Key": "TxnSmsSent",
        "Type": "Double",
        "Value": "10.0"
      }
    ]
  }
],
"StartTime": "2019-09-06T00:00:00Z"
}
```

In questo caso, il campo `GroupedBy` indica che i valori sono raggruppati per giorno di calendario (`Date`). Ciò significa che:

- 29 messaggi sono stati inviati il 6 settembre 2019.
- 35 messaggi sono stati inviati il 7 settembre 2019.
- 10 messaggi sono stati inviati l'8 settembre 2019.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta [Utilizzo dei risultati delle query](#).

# Utilizzo dei risultati delle query sui dati di analisi di Amazon Pinpoint

Quando si utilizzano le API di analisi Amazon Pinpoint per eseguire query sui dati di analisi, Amazon Pinpoint restituisce i risultati in una risposta JSON. Per le metriche delle applicazioni, le metriche delle campagne e le metriche di coinvolgimento del percorso, i dati nella risposta sono conformi a uno schema JSON standard per la creazione di report di dati di analisi di Amazon Pinpoint.

Ciò significa che è possibile utilizzare il linguaggio di programmazione o lo strumento di propria scelta per implementare una soluzione personalizzata che esegua query sui dati per uno o più di questi parametri, acquisisca i risultati di ogni query e quindi scriva i risultati in una tabella, un oggetto o un'altra posizione. È quindi possibile utilizzare i risultati delle query in tale posizione utilizzando un altro servizio o applicazione.

Ad esempio, puoi:

- Creare un dashboard personalizzato che esegua regolarmente le query su un set di parametri e visualizzi i risultati utilizzando il framework di visualizzazione dei dati preferito.
- Creare un report che tenga traccia dei tassi di coinvolgimento eseguendo una query sui parametri appropriati e visualizzando i risultati in un grafico o in un altro tipo di report che si progetta.
- Analizzare e scrivere i dati di analisi in un particolare formato di storage e quindi eseguire il port dei risultati in una soluzione di storage a lungo termine.

Tieni presente che le API di analisi di Amazon Pinpoint non sono progettate per creare o archiviare oggetti persistenti che è possibile leggere o utilizzare successivamente in un progetto Amazon Pinpoint o nell'account Amazon Pinpoint. Le API sono invece progettate per aiutarti a recuperare i dati di analisi e a trasferirli ad altri servizi e applicazioni per ulteriori analisi, storage o report. Lo fanno in parte utilizzando la stessa struttura di risposta JSON e lo stesso schema per tutti i dati di analisi su cui è possibile eseguire query in modo programmatico per i parametri dell'applicazione, i parametri della campagna e i parametri di coinvolgimento del viaggio.

In questo argomento vengono illustrati la struttura, gli oggetti e i campi in una risposta JSON a una query per un parametro dell'applicazione, un parametro della campagna o un parametro di coinvolgimento del viaggio. Per informazioni sui campi in una risposta JSON a una query relativa a un parametro di esecuzione del viaggio o a un parametro di esecuzione dell'attività di viaggio, consulta [Metriche di analisi standard di Amazon Pinpoint](#).

## Struttura JSON

Per semplificare l'analisi e l'utilizzo dei risultati delle query, le API di analisi di Amazon Pinpoint utilizzano la stessa struttura di risposta JSON per tutti i dati di analisi Amazon Pinpoint sui quali è possibile eseguire query a livello di programmazione relativamente a metriche delle applicazioni, metriche delle campagne e metriche di coinvolgimento del percorso. Ogni risposta JSON specifica i valori che hanno definito la query, ad esempio l'ID del progetto (`ApplicationId`). La risposta include anche un (e solo un) oggetto `KpiResult`. L'oggetto `KpiResult` contiene il set di risultati complessivo per una query.

Ogni oggetto `KpiResult` contiene un oggetto `Rows`. Si tratta di un array di oggetti che contengono risultati di query e metadati pertinenti sui valori in tali risultati. La struttura e il contenuto di un oggetto `Rows` presentano le seguenti caratteristiche generali:

- Ogni riga di risultati di query è un oggetto JSON separato, denominato `Values`, nell'oggetto `Rows`. Ad esempio, se una query restituisce tre valori, l'oggetto `Rows` contiene tre oggetti `Values`. Ogni oggetto `Values` contiene un singolo risultato per la query.
- Ogni colonna dei risultati della query è una proprietà dell'oggetto `Values` a cui si applica. Il nome della colonna viene archiviato nel campo `Key` dell'oggetto `Values`.
- Per i risultati delle query raggruppate, ogni oggetto `Values` ha un oggetto `GroupedBy` associato. L'oggetto `GroupedBy` indica quale campo è stato utilizzato per raggruppare i risultati. Fornisce inoltre il valore di raggruppamento per l'oggetto `Values` associato.
- Se i risultati della query per un parametro sono null, l'oggetto `Rows` è vuoto.

Al di là di queste caratteristiche generali, la struttura e il contenuto dell'oggetto `Rows` variano a seconda del parametro. Questo perché Amazon Pinpoint supporta due tipi di metriche, ovvero metriche a valore singolo e metriche a valore multiplo.

Un parametro a valore singolo fornisce un solo valore cumulativo. Un esempio è la percentuale di messaggi recapitati ai destinatari da tutte le esecuzioni di una campagna. Un parametro a valore multiplo fornisce più di un valore e raggruppa tali valori in base a un campo pertinente. Un esempio è la percentuale di messaggi recapitati ai destinatari per ogni esecuzione di una campagna, raggruppati per esecuzione di campagna.

È possibile stabilire rapidamente se un parametro è un parametro a valore singolo o un parametro a valore multiplo facendo riferimento al nome del parametro. Se il nome non contiene `grouped-by`, si tratta di un parametro a valore singolo. Se lo fa, è un parametro a valore multiplo. Per un

elenco completo dei parametri su cui è possibile eseguire query a livello di programmazione, consulta [Metriche di analisi standard di Amazon Pinpoint](#).

## Parametri a valore singolo

Per un parametro a valore singolo, l'oggetto Rows contiene un oggetto Values che:

- Specifica il nome descrittivo del parametro su cui è stata eseguita la query.
- Fornisce il valore per il parametro su cui è stata eseguita la query.
- Identifica il tipo di dati del valore restituito.

Ad esempio, la seguente risposta JSON contiene i risultati della query per un parametro a valore singolo. Questo parametro riporta il numero di endpoint univoci a cui sono stati recapitati i messaggi da tutte le campagne associate a un progetto, dal 1° agosto 2019 al 31 agosto 2019:

```
{
  "ApplicationDateRangeKpiResponse": {
    "ApplicationId": "1234567890123456789012345example",
    "EndTime": "2019-08-31T23:59:59Z",
    "KpiName": "unique-deliveries",
    "KpiResult": {
      "Rows": [
        {
          "Values": [
            {
              "Key": "UniqueDeliveries",
              "Type": "Double",
              "Value": "1368.0"
            }
          ]
        }
      ]
    },
    "StartTime": "2019-08-01T00:00:00Z"
  }
}
```

In questo esempio, la risposta indica che tutte le campagne del progetto hanno recapitato messaggi a 1.368 endpoint univoci dal 1 agosto 2019 al 31 agosto 2019, dove:

- `Key` è il nome descrittivo del parametro il cui valore è specificato nel campo `Value` (`UniqueDeliveries`).
- `Type` è il tipo di dati del valore specificato nel campo `Value` (`Double`).
- `Value` è il valore effettivo per il parametro su cui è stata eseguita la query, inclusi eventuali filtri applicati (`1368.0`).

Se il risultato della query per una metrica a valore singolo è nullo (non maggiore o uguale a zero), l'oggetto `Rows` è vuoto. Amazon Pinpoint restituisce un valore nullo per una metrica se non ci sono dati da restituire per la metrica. Ad esempio:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"2345678901234567890123456example",
    "EndTime":"2019-08-31T23:59:59Z",
    "KpiName":"unique-deliveries",
    "KpiResult":{
      "Rows":[

    ]
    },
    "StartTime":"2019-08-01T00:00:00Z"
  }
}
```

## Parametri a più valori

La struttura e il contenuto dell'oggetto `Rows` per un parametro a più valori sono per lo più uguali a un parametro a valore singolo. L'oggetto `Rows` per un parametro a valore multiplo contiene anche un oggetto `Values`. L'oggetto `Values` specifica il nome descrittivo del parametro con query, fornisce il valore per tale parametro e identifica il tipo di dati di tale valore.

Tuttavia, l'oggetto `Rows` per un parametro a più valori contiene anche uno o più oggetti `GroupedBy`. Esiste un oggetto `GroupedBy` per ogni oggetto `Values` nei risultati della query. L'oggetto `GroupedBy` indica quale campo è stato utilizzato per raggruppare i dati nei risultati e il tipo di dati di quel campo. Indica anche il valore di raggruppamento per quel campo (per l'oggetto `Values` associato).

Ad esempio, la seguente risposta JSON contiene i risultati della query per un parametro a più valori che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi per ogni campagna associata a un progetto, dal 1 agosto 2019 al 31 agosto 2019:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-08-31T23:59:59Z",
    "KpiName":"unique-deliveries-grouped-by-campaign",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"80b8efd84042ff8d9c96ce2f8example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"123.0"
            }
          ]
        },
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"810c7aab86d42fb2b56c8c966example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"456.0"
            }
          ]
        }
      ]
    }
  },
}
```

```
{
  "GroupedBy": [
    {
      "Key": "CampaignId",
      "Type": "String",
      "Value": "42d8c7eb0990a57ba1d5476a3example"
    }
  ],
  "Values": [
    {
      "Key": "UniqueDeliveries",
      "Type": "Double",
      "Value": "789.0"
    }
  ]
},
"StartTime": "2019-08-01T00:00:00Z"
}
```

In questo esempio, la risposta indica che tre delle campagne del progetto hanno inviato messaggi a endpoint univoci dal 1 agosto 2019 al 31 agosto 2019. Per ciascuna di queste campagne, la ripartizione dei conteggi dei recapiti è:

- La campagna `80b8efd84042ff8d9c96ce2f8example` ha recapitato messaggi a 123 endpoint univoci.
- La campagna `810c7aab86d42fb2b56c8c966example` ha recapitato messaggi a 456 endpoint univoci.
- La campagna `42d8c7eb0990a57ba1d5476a3example` ha recapitato messaggi a 789 endpoint univoci.

Dove la struttura generale degli oggetti e dei campi è:

- `GroupedBy.Key`: nome della proprietà o del campo che archivia il valore di raggruppamento specificato nel campo `GroupedBy.Value` (`CampaignId`).
- `GroupedBy.Type`: tipo di dati del valore specificato nel campo `GroupedBy.Value` (`String`).

- `GroupedBy.Value`: valore effettivo del campo utilizzato per raggruppare i dati, come specificato nel campo `GroupedBy.Key` (ID campagna).
- `Values.Key`: nome descrittivo della metrica il cui valore è specificato nel campo `Values.Value` (`UniqueDeliveries`).
- `Values.Type`: tipo di dati del valore specificato nel campo `Values.Value` (`Double`).
- `Values.Value`: valore effettivo della metrica su cui è stata eseguita la query, inclusi tutti i filtri applicati.

Se i risultati della query per una metrica a più valori sono null (non maggiori di o uguali a zero) per un progetto, una campagna o un'altra risorsa specifica, Amazon Pinpoint non restituisce alcun oggetto o campo per la risorsa. Se i risultati della query per una metrica a più valori sono null per tutte le risorse, Amazon Pinpoint restituisce un oggetto `Rows` vuoto.

## Oggetti e campi JSON

Oltre a specificare i valori che hanno definito una query, ad esempio l'ID del progetto (`ApplicationId`), ogni risposta JSON a una query per un parametro dell'applicazione, il parametro della campagna o il parametro del viaggio include un oggetto `KpiResult`. Questo oggetto contiene il set di risultati complessivo per una query, che è possibile analizzare per inviare dati di analisi analitici a un altro servizio o applicazione. Ogni oggetto `KpiResult` contiene alcuni o tutti i seguenti oggetti e campi standard, a seconda del parametro.

Oggetto o campo	Descrizione
<code>Rows</code>	Un array di oggetti che contiene il set di risultati per una query.
<code>Rows.GroupedBy</code>	Per un parametro a più valori, un parametro di campi che definisce il campo e i valori utilizzati per raggruppare i dati nei risultati delle query.
<code>Rows.GroupedBy.Key</code>	Per un parametro a più valori, il nome della proprietà o del campo che memorizza il valore specificato nel campo <code>GroupedBy.Value</code> .

Oggetto o campo	Descrizione
<code>Rows.GroupedBy.Type</code>	Per un parametro a più valori, il tipo di dati del valore specificato nel campo <code>GroupedBy.Value</code> .
<code>Rows.GroupedBy.Value</code>	Per un parametro a più valori, il valore effettivo per il campo utilizzato per raggruppare i dati nei risultati delle query. Questo valore è correlato a un oggetto <code>Values</code> associato.
<code>Rows.Values</code>	Un array di campi che contiene i risultati delle query.
<code>Rows.Values.Key</code>	Il nome descrittivo del parametro su cui è stata eseguita la query. Il valore del parametro è specificato nel campo <code>Values.Value</code> .
<code>Rows.Values.Type</code>	Il tipo di dati del valore specificato nel campo <code>Values.Value</code> .
<code>Rows.Values.Value</code>	Il valore effettivo per il parametro su cui è stata eseguita la query, inclusi tutti i filtri applicati.

Per informazioni sui campi in una risposta JSON a una query relativa a un parametro di esecuzione del viaggio o a un parametro di esecuzione dell'attività di viaggio, consulta [Metriche di analisi standard di Amazon Pinpoint](#).

# Registrazione delle chiamate API Amazon Pinpoint con AWS CloudTrail

Amazon Pinpoint è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Pinpoint. CloudTrail acquisisce le chiamate API per Amazon Pinpoint come eventi. Le chiamate acquisite includono le chiamate dalla console Amazon Pinpoint e le chiamate di codice alle operazioni delle API di Amazon Pinpoint.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon Simple Storage Service (Amazon S3), inclusi gli eventi per Amazon Pinpoint. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti utilizzando la cronologia degli eventi sulla console. CloudTrail Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata ad Amazon Pinpoint, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la [Guida per l'AWS CloudTrail utente](#).

## Informazioni su Amazon Pinpoint in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in Amazon Pinpoint, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Pinpoint, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)

- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di stabilire:

- Se la richiesta è stata effettuata con credenziali root o AWS Identity and Access Management utente.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta [Elemento userIdentity di CloudTrail](#).

Puoi creare un trail e archiviare i file di log nel bucket Amazon S3 per il periodo di tempo desiderato. Puoi anche definire regole del ciclo di vita di Amazon S3 per archiviare o eliminare file di log automaticamente. Per impostazione predefinita, i file di log sono crittografati mediante la crittografia lato server (SSE) di Amazon S3.

Per ricevere una notifica della consegna dei file di log, configura CloudTrail la pubblicazione di notifiche Amazon SNS quando vengono consegnati nuovi file di log. Per ulteriori informazioni, consulta [Configurazione delle notifiche di Amazon SNS](#) per CloudTrail

Puoi anche aggregare i file di log di Amazon Pinpoint da AWS più regioni e AWS più account in un unico bucket Amazon S3. Per ulteriori informazioni, consulta [Ricezione di file di CloudTrail registro da più regioni](#) e [Ricezione di file di CloudTrail registro](#) da più account.

Puoi utilizzarlo CloudTrail per registrare le azioni per le seguenti API Amazon Pinpoint:

- [API Amazon Pinpoint](#)
- [API SMS e Voce di Amazon Pinpoint](#)

## Azioni dell'API Amazon Pinpoint che possono essere registrate da CloudTrail

L'API Amazon Pinpoint supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- [CreateApp](#)
- [CreateCampaign](#)
- [CreateEmailTemplate](#)
- [CreateExportJob](#)
- [CreateImportJob](#)
- [CreateJourney](#)
- [CreatePushTemplate](#)
- [CreateRecommenderConfiguration](#)
- [CreateSegment](#)
- [CreateSmsTemplate](#)
- [CreateVoiceTemplate](#)
- [DeleteAdmChannel](#)
- [DeleteApnsChannel](#)
- [DeleteApnsSandboxChannel](#)
- [DeleteApnsVoipChannel](#)
- [DeleteApnsVoipSandboxChannel](#)
- [DeleteApp](#)
- [DeleteBaiduChannel](#)
- [DeleteCampaign](#)
- [DeleteEmailChannel](#)
- [DeleteEmailTemplate](#)
- [DeleteEndpoint](#)
- [DeleteEventStream](#)
- [DeleteGcmChannel](#)
- [DeleteJourney](#)
- [DeletePushTemplate](#)
- [DeleteRecommenderConfiguration](#)
- [DeleteSegment](#)

- [DeleteSmsChannel](#)
- [DeleteSmsTemplate](#)
- [DeleteUserEndpoints](#)
- [DeleteVoiceChannel](#)
- [DeleteVoiceTemplate](#)
- [GetAdmChannel](#)
- [GetApnsChannel](#)
- [GetApnsSandboxChannel](#)
- [GetApnsVoipChannel](#)
- [GetApnsVoipSandboxChannel](#)
- [GetApp](#)
- [GetApplicationDateRangeKpi](#)
- [GetApplicationSettings](#)
- [GetApps](#)
- [GetBaiduChannel](#)
- [GetCampaign](#)
- [GetCampaignActivities](#)
- [GetCampaignDateRangeKpi](#)
- [GetCampaignVersion](#)
- [GetCampaignVersions](#)
- [GetCampaigns](#)
- [GetChannels](#)
- [GetEmailChannel](#)
- [GetEmailTemplate](#)
- [GetEndpoint](#)
- [GetEventStream](#)
- [GetExportJob](#)
- [GetExportJobs](#)

- [GetGcmChannel](#)
- [GetImportJob](#)
- [GetImportJobs](#)
- [GetJourney](#)
- [GetJourneyDateRangeKpi](#)
- [GetJourneyExecutionActivityMetrics](#)
- [GetJourneyExecutionMetrics](#)
- [GetPushTemplate](#)
- [GetRecommenderConfiguration](#)
- [GetRecommenderConfigurations](#)
- [GetSegment](#)
- [GetSegmentExportJobs](#)
- [GetSegmentImportJobs](#)
- [GetSegmentVersion](#)
- [GetSegmentVersions](#)
- [GetSegments](#)
- [GetSmsChannel](#)
- [GetSmsTemplate](#)
- [GetUserEndpoints](#)
- [GetVoiceChannel](#)
- [GetVoiceTemplate](#)
- [ListJourneys](#)
- [ListTagsForResource](#)
- [ListTemplates](#)
- [ListTemplateVersions](#)
- [PhoneNumberValidate](#)
- [PutEventStream](#)
- [RemoveAttributes](#)

- [TagResource](#)
- [UntagResource](#)
- [UpdateAdmChannel](#)
- [UpdateApnsChannel](#)
- [UpdateApnsSandboxChannel](#)
- [UpdateApnsVoipChannel](#)
- [UpdateApnsVoipSandboxChannel](#)
- [UpdateApplicationSettings](#)
- [UpdateBaiduChannel](#)
- [UpdateCampaign](#)
- [UpdateEmailChannel](#)
- [UpdateEmailTemplate](#)
- [UpdateEndpoint](#)
- [UpdateEndpointsBatch](#)
- [UpdateGcmChannel](#)
- [UpdateJourney](#)
- [UpdateJourneyState](#)
- [UpdatePushTemplate](#)
- [UpdateRecommenderConfiguration](#)
- [UpdateSegment](#)
- [UpdateSmsChannel](#)
- [UpdateSmsTemplate](#)
- [UpdateTemplateActiveVersion](#)
- [UpdateVoiceChannel](#)
- [UpdateVoiceTemplate](#)

Le seguenti azioni dell'API Amazon Pinpoint non sono registrate: CloudTrail

- PutEvents
- SendMessages

- [SendUsersMessages](#)

## Azioni dell'API e-mail di Amazon Pinpoint che possono essere registrate da CloudTrail

L'API Amazon Pinpoint Email supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- [CreateConfigurationSet](#)
- [CreateConfigurationSetEventDestination](#)
- [CreateDedicatedIpPool](#)
- [CreateEmailIdentity](#)
- [DeleteConfigurationSet](#)
- [DeleteConfigurationSetEventDestination](#)
- [DeleteDedicatedIpPool](#)
- [DeleteEmailIdentity](#)
- [GetAccount](#)
- [GetConfigurationSet](#)
- [GetConfigurationSetEventDestinations](#)
- [GetDedicatedIp](#)
- [GetDedicatedIps](#)
- [GetEmailIdentity](#)
- [ListConfigurationSets](#)
- [ListDedicatedIpPools](#)
- [ListEmailIdentities](#)
- [PutAccountDedicatedIpWarmupAttributes](#)
- [PutAccountSendingAttributes](#)
- [PutConfigurationSetDeliveryOptions](#)
- [PutConfigurationSetReputationOptions](#)
- [PutConfigurationSetSendingOptions](#)
- [PutConfigurationSetTrackingOptions](#)

- [PutDedicatedIpInPool](#)
- [PutDedicatedIpWarmupAttributes](#)
- [PutEmailIdentityDkimAttributes](#)
- [PutEmailIdentityFeedbackAttributes](#)
- [PutEmailIdentityMailFromAttributes](#)
- [UpdateConfigurationSetEventDestination](#)

La seguente azione dell'API Amazon Pinpoint Email non è stata registrata: CloudTrail

- SendEmail

## Azioni dell'API vocale e SMS di Amazon Pinpoint versione 1 che possono essere registrate da CloudTrail

L'API Amazon Pinpoint SMS and Voice versione 1 supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- [CreateConfigurationSet](#)
- [CreateConfigurationSetEventDestination](#)
- [DeleteConfigurationSet](#)
- [DeleteConfigurationSetEventDestination](#)
- [GetConfigurationSetEventDestinations](#)
- [UpdateConfigurationSetEventDestination](#)

La seguente azione API Amazon Pinpoint SMS and Voice versione 1 non è stata registrata: CloudTrail

- SendVoiceMessage

## Esempi: voci del file di log Amazon Pinpoint

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro.

Un evento rappresenta una singola richiesta da un'origine. Include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra CreateCampaign le azioni GetCampaigns e le azioni dell'API Amazon Pinpoint.

```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpURLConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "type": "Root"
    }
  ],
  {
```

```
"awsRegion": "us-east-1",
"eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
"eventName": "CreateCampaign",
"eventSource": "pinpoint.amazonaws.com",
"eventTime": "2018-02-03T01:05:16Z",
"eventType": "AwsApiCall",
"eventVersion": "1.05",
"readOnly": false,
"recipientAccountId": "123456789012",
"requestID": "example1-b9bb-50fa-abdb-80f274981d60",
"requestParameters": {
  "Description": "****",
  "HoldoutPercent": 0,
  "IsPaused": false,
  "MessageConfiguration": "****",
  "Name": "****",
  "Schedule": {
    "Frequency": "ONCE",
    "IsLocalTime": true,
    "StartTime": "2018-02-03T00:00:00-08:00",
    "Timezone": "utc-08"
  },
  "SegmentId": "exampleda204adf991a80281aa0e591",
  "SegmentVersion": 1,
  "application-id": "example71dfa4c1aab66332a5839798f"
},
"responseElements": {
  "ApplicationId": "example71dfa4c1aab66332a5839798f",
  "CreationDate": "2018-02-03T01:05:16.425Z",
  "Description": "****",
  "HoldoutPercent": 0,
  "Id": "example54a654f80948680cbba240ede",
  "IsPaused": false,
  "LastModifiedDate": "2018-02-03T01:05:16.425Z",
  "MessageConfiguration": "****",
  "Name": "****",
  "Schedule": {
    "Frequency": "ONCE",
    "IsLocalTime": true,
    "StartTime": "2018-02-03T00:00:00-08:00",
    "Timezone": "utc-08"
  },
  "SegmentId": "example4da204adf991a80281example",
  "SegmentVersion": 1,
```

```

    "State": {
      "CampaignStatus": "SCHEDULED"
    },
    "Version": 1
  },
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
  "userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/userName",
    "principalId": "AIDAIHTRCDA62EXAMPLE",
    "type": "IAMUser",
    "userName": "userName"
  }
}
]
}

```

L'esempio seguente mostra una voce di CloudTrail registro che illustra le `CreateConfigurationSetEventDestination` azioni `CreateConfigurationSet` e nell'API SMS and Voice di Amazon Pinpoint.

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIHTRCDA62EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/SampleUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "SampleUser"
      },
      "eventTime": "2018-11-06T21:45:55Z",
      "eventSource": "sms-voice.amazonaws.com",
      "eventName": "CreateConfigurationSet",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.0.1",
      "userAgent": "PostmanRuntime/7.3.0",
      "requestParameters": {
        "ConfigurationSetName": "MyConfigurationSet"
      }
    }
  ]
}

```

```

    },
    "responseElements":null,
    "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
    "eventID":"725843fc-8846-41f4-871a-7c52dexample",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "recipientAccountId":"123456789012"
  },
  {
    "eventVersion":"1.05",
    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDAIHTRCDA62EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/SampleUser",
      "accountId":"111122223333",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"SampleUser"
    },
    "eventTime":"2018-11-06T21:47:08Z",
    "eventSource":"sms-voice.amazonaws.com",
    "eventName":"CreateConfigurationSetEventDestination",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"192.0.0.1",
    "userAgent":"PostmanRuntime/7.3.0",
    "requestParameters":{
      "EventDestinationName":"CloudWatchEventDestination",
      "ConfigurationSetName":"MyConfigurationSet",
      "EventDestination":{
        "Enabled":true,
        "MatchingEventTypes":[
          "INITIATED_CALL",
          "INITIATED_CALL"
        ],
        "CloudWatchLogsDestination":{
          "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
          "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
        }
      }
    }
  },
  "responseElements":null,
  "requestID":"81de1e73-e20d-11e8-b158-d5536example",
  "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
  "readOnly":false,

```

```
    "eventType": "AwsApiCall",  
    "recipientAccountId": "111122223333"  
  }  
]  
}
```

# Assegnazione di tag alle risorse Amazon Pinpoint

Un tag è un'etichetta che è possibile definire e associare alle risorse AWS, inclusi alcuni tipi di risorse Amazon Pinpoint. I tag consentono di categorizzare e gestire le risorse in diversi modi, ad esempio in base allo scopo, al proprietario, all'ambiente o ad altri criteri. Ad esempio, è possibile utilizzare i tag per applicare le policy o l'automazione, nonché per identificare le risorse che sono soggette a determinati requisiti di conformità. Puoi aggiungere tag ai seguenti tipi di risorse Amazon Pinpoint:

- Campagne
- Modelli dei messaggi
- Progetti (applicazioni)
- Segmenti

Una risorsa può avere fino a 50 tag.

## Gestione dei tag

Ogni tag è composto da una chiave di tag obbligatoria e da un valore di tag opzionale, entrambi definibili dall'utente. Una chiave di tag è un'etichetta generale che funge da categoria per più valori di tag specifici. Un valore di tag funge da descrittore di una chiave di tag.

Una chiave di tag può contenere fino a un massimo di 128 caratteri. Un valore di tag può contenere fino a un massimo di 256 caratteri. I caratteri possono essere lettere Unicode, numeri, spazi o uno dei seguenti simboli: `_ . : / = + -`. Ai tag si applicano le seguenti ulteriori limitazioni:

- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Per ogni risorsa associata, ciascuna chiave di tag deve essere univoca e avere un solo valore.
- Il prefisso `aws:` è riservato all'uso da parte di AWS; non è possibile utilizzarlo in chiavi e valori di tag definibili dall'utente. Inoltre, non è possibile modificare o rimuovere chiavi o valori di tag che utilizzano questo prefisso. I tag che utilizzano questo prefisso non vengono conteggiati per la quota di 50 tag per ogni risorsa.
- Non è possibile aggiornare o eliminare una risorsa solo in base ai suoi tag. Inoltre, è necessario specificare l'ARN (Amazon Resource Name) o l'ID della risorsa, a seconda dell'operazione utilizzata.

- È possibile associare i tag a risorse pubbliche o condivise. I tag, però, sono disponibili solo per l'account AWS dell'utente e non per gli altri account che condividono la risorsa. Inoltre, i tag sono disponibili solo per le risorse che si trovano nella regione AWS specificata per l'account AWS dell'utente.

Per aggiungere, visualizzare, aggiornare e rimuovere chiavi e valori di tag dalle risorse Amazon Pinpoint, è possibile utilizzare l'AWS Command Line Interface (AWS CLI), l'API Amazon Pinpoint, l'API per l'assegnazione di tag a AWS Resource Groups o AWS SDK. Per gestire chiavi e valori di tag in tutte le risorse AWS che si trovano in una regione AWS specifica per il tuo account AWS, incluse le risorse Amazon Pinpoint, puoi utilizzare [l'API per l'assegnazione di tag a AWS Resource Groups](#).

## Utilizzo di tag nelle policy IAM

Dopo aver avviato l'implementazione dei tag, è possibile applicare autorizzazioni a livello di risorse basate su tag alle policy AWS Identity and Access Management (IAM) e alle operazioni API. Ciò include le operazioni che supportano l'aggiunta di tag alle risorse al momento della loro creazione. Utilizzando i tag in questo modo, è possibile implementare il controllo granulare dei gruppi e degli utenti nell'account AWS che sono autorizzati a creare e contrassegnare con tag risorse e più in generale dei gruppi e degli utenti che sono autorizzati a creare, aggiornare e rimuovere tag.

Ad esempio, è possibile creare una policy che consenta a un utente di avere accesso completo a tutte le risorse Amazon Pinpoint il cui nome è un valore nel tag Owner della risorsa:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "mobiletargeting:*",
      "Resource": "*",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

Se vengono definite autorizzazioni a livello di risorsa basate su tag, le autorizzazioni diventano subito effettive. Ciò significa che le risorse sono più sicure non appena vengono create e che è possibile avviare rapidamente l'applicazione di tag alle nuove risorse. È inoltre possibile utilizzare le autorizzazioni a livello di risorsa per controllare quali chiavi e valori di tag possono essere associati a risorse nuove ed esistenti. Per ulteriori informazioni, consulta [Controllo dell'accesso alle risorse AWS mediante i tag](#) nella Guida per l'utente di AWS.

## Aggiunta di tag alle risorse

Nei seguenti esempi viene illustrato come aggiungere un tag a una risorsa Amazon Pinpoint utilizzando la [AWS CLI](#) e la [REST API Amazon Pinpoint](#). È inoltre possibile utilizzare qualsiasi SDK AWS supportato per aggiungere un tag a una risorsa.

Per aggiungere un tag a più risorse Amazon Pinpoint in un'unica operazione, utilizza le operazioni di assegnazione di tag a gruppi di risorse della AWS CLI o l'[API per l'assegnazione di tag a AWS Resource Groups](#).

## Aggiunta di tag mediante l'API

Per creare una nuova risorsa e aggiungere un tag utilizzando la REST API Amazon Pinpoint, invia una richiesta POST all'URI di risorsa appropriato. Includi il parametro `tags` e i valori nel corpo della richiesta. L'esempio seguente mostra come specificare un tag quando si crea un nuovo progetto.

```
POST /v1/apps HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Cache-Control: no-cache

{
  "Name": "MyProject",
  "tags": {
    "key1": "value1"
  }
}
```

Per aggiungere un tag a una risorsa esistente, invia una richiesta POST all'URI [Tag](#). Includi l'Amazon Resource Name (ARN) della risorsa nell'URI. L'ARN deve essere codificato in formato URL. Nel corpo della richiesta, includi il parametro `tags` e i valori, come mostrato nel seguente esempio.

```
POST /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

```
{
  "tags":{
    "key1":"value1"
  }
}
```

## Aggiunta di tag utilizzando la AWS CLI

Per creare una nuova risorsa e aggiungervi un tag utilizzando l'AWS CLI, usa il comando `create` appropriato per la risorsa. Includi il parametro `tags` e i valori. L'esempio seguente mostra come specificare i tag quando si crea un nuovo progetto.

Linux, macOS, or Unix

```
$ aws pinpoint create-app \  
  --create-application-request '{  
    "Name": "MyProject",  
    "tags": {  
      "key1": "value1",  
      "key2": "value2"  
    }  
  }'
```

Windows Command prompt

```
C:\> aws pinpoint create-app ^  
  --create-application-request Name=MyProject,tags={key1=value1,key2=value2}
```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci *MyProject* con il nome da assegnare al progetto.
- Sostituisci *key1* e *key2* con le chiavi dei tag che desideri aggiungere alla risorsa.

- Sostituisci *value1* e *value2* con i valori dei tag che desideri aggiungere alle rispettive chiavi.

Per informazioni sui comandi utilizzabili per creare una risorsa Amazon Pinpoint, consulta la [documentazione di riferimento dei comandi della AWS CLI](#).

Per aggiungere un tag a una risorsa esistente, utilizzare il comando `tag-resource` e specificare i valori appropriati per i parametri richiesti:

Linux, macOS, or Unix

```
$ aws pinpoint tag-resource \  
  --resource-arn resource-arn \  
  --tags-model '{  
    "tags": {  
      "key1": "value1",  
      "key2": "value2"  
    }  
  }'
```

Windows Command Prompt

```
C:\> aws pinpoint tag-resource ^  
  --resource-arn resource-arn ^  
  --tags-model tags={key1=value1,key2=value2}
```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci *resource-arn* con il nome della risorsa Amazon (ARN) della risorsa a cui desideri aggiungere un tag.
- Sostituisci *key1* e *key2* con le chiavi dei tag che desideri aggiungere alla risorsa.
- Sostituisci *value1* e *value2* con i valori dei tag che desideri aggiungere alle rispettive chiavi.

## Visualizzazione dei tag per le risorse

I seguenti esempi mostrano come utilizzare la [AWS CLI](#) e la [REST API Amazon Pinpoint](#) per visualizzare l'elenco di tutti i tag (chiavi e valori) associati a una risorsa Amazon Pinpoint. È inoltre possibile utilizzare qualsiasi versione supportata di AWS SDK per visualizzare tutti i tag associati a una risorsa.

## Visualizzazione dei tag mediante l'API

Per utilizzare la REST API Amazon Pinpoint per visualizzare tutti i tag associati a una risorsa specifica, inviare una richiesta GET all'URI [Tags](#) e includere il nome della risorsa Amazon (ARN) della risorsa nell'URI. L'ARN deve essere codificato in formato URL. Ad esempio, la richiesta seguente recupera tutti i tag associati a una campagna specifica (*resource-arn*):

```
GET /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

La risposta JSON alla richiesta include un oggetto `tags`. L'oggetto `tags` elenca tutte le chiavi e i valori di tag associati alla campagna.

Per visualizzare tutti i tag associati a più di una risorsa dello stesso tipo, inviare una richiesta GET all'URI appropriato per quel tipo di risorsa. Ad esempio, la seguente richiesta recupera le informazioni su tutte le campagne nel progetto specificato (*application-id*):

```
GET /v1/apps/application-id/campaigns HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

La risposta JSON alla richiesta elenca tutte le campagne nel progetto. L'oggetto `tags` di ciascuna campagna elenca tutte le chiavi e i valori di tag associati alla campagna.

## Visualizzazione dei tag mediante la AWS CLI

Per utilizzare la AWS CLI per visualizzare l'elenco dei tag associati a una risorsa specifica, esegui il comando `list-tags-for-resource` e specifica il nome della risorsa Amazon (ARN) della risorsa per il parametro `resource-arn`, come mostrato nell'esempio seguente.

Linux, macOS, or Unix

```
$ aws pinpoint list-tags-for-resource \  
  --resource-arn resource-arn
```

## Windows Command Prompt

```
C:\> aws pinpoint list-tags-for-resource ^  
    --resource-arn resource-arn
```

Per visualizzare l'elenco di tutte le risorse Amazon Pinpoint con tag e di tutti i tag associati a ciascuna di tali risorse, utilizza il comando [get-resources](#) dell'API per l'assegnazione di tag a AWS Resource Groups. Imposta il parametro `resource-type-filters` su `mobiletargeting`, come mostrato nell'esempio seguente.

## Linux, macOS, or Unix

```
$ aws resourcegroupstaggingapi get-resources \  
    --resource-type-filters "mobiletargeting"
```

## Windows Command Prompt

```
C:\> aws resourcegroupstaggingapi get-resources ^  
    --resource-type-filters "mobiletargeting"
```

L'output del comando è un elenco di ARN per tutte le risorse Amazon Pinpoint con tag. L'elenco include tutte le chiavi e valori di tag che sono associati a ogni risorsa.

## Aggiornamento dei tag per le risorse

Sono disponibili diversi modi per aggiornare (sovrascrivere) un tag di una risorsa Amazon Pinpoint. Il modo migliore per aggiornare un tag dipende da:

- Il tipo di risorsa di cui si desidera aggiornare i tag.
- Se si desidera aggiornare un tag per una o più risorse contemporaneamente.
- Se si desidera aggiornare una chiave di tag, un valore di tag o entrambi.

Per aggiornare un tag per un progetto Amazon Pinpoint o per più risorse contemporaneamente, utilizza le operazioni di assegnazione di tag a gruppi di risorse di AWS CLI o [l'API per l'assegnazione di tag a AWS Resource Groups](#). L'API Amazon Pinpoint attualmente non fornisce supporto diretto per nessuna di queste attività.

Per aggiornare un tag per una risorsa, è possibile [rimuovere il tag attuale](#) e [aggiungere un nuovo tag](#) tramite l'API Amazon Pinpoint.

## Rimozione dei tag dalle risorse

Negli esempi seguenti viene illustrato come rimuovere un tag (sia la chiave che il valore) da una risorsa Amazon Pinpoint utilizzando la [AWS CLI](#) e la [REST API Amazon Pinpoint](#). È inoltre possibile utilizzare qualsiasi SDK AWS supportato per rimuovere un tag da una risorsa.

Per rimuovere un tag da più risorse Amazon Pinpoint in un'unica operazione, utilizza le operazioni di assegnazione di tag a gruppi di risorse della AWS CLI o l'[API per l'assegnazione di tag a AWS Resource Groups](#). Per rimuovere da una risorsa solo un determinato valore di tag (non la chiave di tag), [aggiornare i tag per la risorsa](#).

## Rimozione dei tag mediante l'API

Per rimuovere un tag da una risorsa mediante la REST API Amazon Pinpoint, invia una richiesta DELETE all'URI [Tags](#). Nell'URI, includere l'Amazon Resource Name (ARN) della risorsa da cui si desidera rimuovere un tag, seguita dal parametro `tagKeys` e dal tag da rimuovere. Ad esempio:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key
```

Dove:

- *endpoint* è l'endpoint Amazon Pinpoint della regione AWS che ospita la risorsa.
- *resource-arn* è l'ARN della risorsa da cui si desidera rimuovere un tag.
- *key* è il tag che si desidera rimuovere dalla risorsa.

Tutti i parametri devono essere codificati URL.

Per rimuovere da una risorsa più chiavi di tag e i relativi valori associati, aggiungere il parametro e l'argomento `tagKeys` per ogni ulteriore tag da rimuovere, separati da una e commerciale (&). Ad esempio:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key1&tagKeys=key2
```

Tutti i parametri devono essere codificati URL.

## Rimozione dei tag mediante la AWS CLI

Per rimuovere un tag da una risorsa utilizzando l'AWS CLI, eseguire il comando `untag-resource`. Includi il parametro `tag-keys`, come mostrato nell'esempio seguente.

Linux, macOS, or Unix

```
$ aws pinpoint untag-resource \  
--resource-arn resource-arn \  
--tag-keys key1 key2
```

Windows Command Prompt

```
C:\> aws pinpoint untag-resource ^  
--resource-arn resource-arn ^  
--tag-keys key1 key2
```

In questo esempio, apporta le modifiche seguenti:

- Sostituisci *resource-arn* con l'ARN della risorsa da cui desideri rimuovere i tag.
- Sostituisci *key1* e *key2* con le chiavi dei tag che desideri rimuovere dalla risorsa.

## Informazioni correlate

Per ulteriori informazioni sui comandi CLI che è possibile utilizzare per gestire le risorse Amazon Pinpoint, consulta la sezione relativa ad Amazon Pinpoint della [documentazione di riferimento dei comandi della AWS CLI](#).

Per ulteriori informazioni sulle risorse nell'API Amazon Pinpoint, compresi i metodi HTTP(S), i parametri e gli schemi supportati, consulta la [documentazione di riferimento dell'API Amazon Pinpoint](#).

# Personalizzazione dei suggerimenti con AWS Lambda

In Amazon Pinpoint, puoi recuperare le raccomandazioni personalizzate da un modello di raccomandazione e aggiungerle ai messaggi inviati da campagne e percorsi. Un modello della funzione di suggerimento è un tipo di modello di machine learning (ML) che individua pattern nei dati e genera previsioni e suggerimenti in base ai pattern individuati. Predice ciò che un particolare utente preferirà da un determinato set di prodotti o articoli, e fornisce tali informazioni come una serie di suggerimenti per l'utente.

Utilizzando i modelli di raccomandazione con Amazon Pinpoint, puoi inviare raccomandazioni personalizzate ai destinatari dei messaggi in base agli attributi e al comportamento di ciascun destinatario. Con AWS Lambda, puoi anche personalizzare e migliorare questi suggerimenti. Ad esempio, puoi trasformare dinamicamente un suggerimento da un singolo valore di testo (ad esempio un nome prodotto o un ID) a contenuti più sofisticati (ad esempio un nome prodotto, una descrizione e un'immagine). Puoi inoltre farlo in tempo reale, quando Amazon Pinpoint invia il messaggio.

Questa funzionalità è disponibile nelle regioni AWS seguenti: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Mumbai), Asia Pacifico (Sydney) ed Europa (Irlanda).

## Argomenti

- [Utilizzo dei suggerimenti nei messaggi](#)
- [Creazione di una funzione Lambda](#)
- [Assegnazione di una policy di funzione Lambda](#)
- [Autorizzazione di Amazon Pinpoint a richiamare la funzione](#)
- [Configurazione del modello della funzione di suggerimento](#)

## Utilizzo dei suggerimenti nei messaggi

Per utilizzare un modello di raccomandazione con Amazon Pinpoint, crea una soluzione Amazon Personalize e quindi implementala come campagna Amazon Personalize. Quindi, puoi creare una configurazione per il modello di raccomandazione in Amazon Pinpoint. Nella configurazione è necessario specificare le impostazioni che determinano come recuperare ed elaborare i dati relativi alle raccomandazioni dalla campagna Amazon Personalize. Ciò include se invocare o meno una funzione AWS Lambda per eseguire un'ulteriore elaborazione dei dati recuperati.

Amazon Personalize è un servizio AWS progettato per aiutarti a creare modelli di machine learning che forniscono raccomandazioni personalizzate in tempo reale per i clienti che utilizzano le tue applicazioni. Grazie a procedure semplificate, Amazon Personalize supporta l'utente nel processo di creazione e addestramento di un modello di machine learning, quindi nella preparazione e implementazione del modello come campagna Amazon Personalize. Puoi quindi recuperare i suggerimenti personalizzati in tempo reale dalla campagna. Per ulteriori informazioni su Amazon Personalize, consulta la [Guida per gli sviluppatori di Amazon Personalize](#).

AWS Lambda è un servizio di calcolo che consente di eseguire il codice senza gestire i server o effettuare il provisioning. Devi pacchettizzare il tuo codice e caricarlo su AWS Lambda come funzione Lambda. Quindi AWS Lambda esegue la funzione quando questa viene richiamata. Puoi richiamare una funzione manualmente, come risposta automatica agli eventi o in risposta a richieste provenienti da applicazioni o servizi, compreso Amazon Pinpoint. Per ulteriori informazioni su come creare e richiamare le funzioni Lambda, consulta [Guida per gli sviluppatori di AWS Lambda](#).

Dopo aver creato una configurazione Amazon Pinpoint per un modello di raccomandazione, puoi aggiungere raccomandazioni dal modello ai messaggi inviati da campagne e percorsi. Puoi eseguire questa operazione utilizzando i modelli di messaggio che contengono variabili di messaggio per gli attributi suggeriti. Un attributo suggerito è un endpoint dinamico o un attributo utente progettato per memorizzare i dati dei suggerimenti. Puoi definire questi attributi crei la configurazione per un modello della funzione di suggerimento.

Puoi utilizzare le variabili per gli attributi suggeriti nei seguenti tipi di modelli di messaggio:

- Modelli di posta elettronica, per i messaggi di posta elettronica inviati da campagne o viaggi.
- Modelli di notifica push, per le notifiche push inviate dalle campagne.
- Modelli SMS, per messaggi di testo SMS inviati da campagne.

Per ulteriori informazioni sull'utilizzo dei modelli di raccomandazione con Amazon Pinpoint, consulta [Modelli di machine learning](#) nella Guida per l'utente di Amazon Pinpoint.

Se configuri Amazon Pinpoint per richiamare una funzione Lambda che elabora i dati delle raccomandazioni, Amazon Pinpoint esegue le seguenti attività generali ogni volta che invia raccomandazioni personalizzate in un messaggio per una campagna o un percorso:

1. Valuta ed elabora le impostazioni di configurazione e il contenuto del messaggio e del modello di messaggio.
2. Determina che il modello di messaggio è connesso a un modello della funzione di suggerimento.

3. Valuta le impostazioni di configurazione per la connessione e l'utilizzo del modello. Queste impostazioni sono definite dalla risorsa del [Modello della funzione di suggerimento](#) per il modello.
4. Rileva una o più variabili di messaggio per gli attributi suggeriti definiti dalle impostazioni di configurazione per il modello.
5. Recupera i dati delle raccomandazioni dalla campagna Amazon Personalize specificata nelle impostazioni di configurazione per il modello. Utilizza l'operazione [GetRecommendations](#) dell'API runtime Amazon Personalize per eseguire questa attività.
6. Aggiunge i dati suggeriti appropriati a un attributo dinamico suggerito (`RecommendationItems`) per ogni destinatario del messaggio.
7. Richiama la funzione Lambda e invia i dati delle raccomandazioni per ciascun destinatario a tale funzione per l'elaborazione.

I dati vengono inviati come oggetto JSON che contiene la definizione dell'endpoint per ogni destinatario. Ogni definizione di endpoint include un campo `RecommendationItems` contenente un array ordinato contenente da 1 a 5 valori. Il numero di valori nella matrice dipende dalle impostazioni di configurazione del modello.

8. Attende che la funzione Lambda elabori i dati e restituisca i risultati.

I risultati consistono in un oggetto JSON che contiene una definizione dell'endpoint aggiornata per ogni destinatario. Ogni definizione dell'endpoint aggiornata contiene un nuovo oggetto `Recommendations`. Questo oggetto contiene da 1 a 10 campi, uno per ogni attributo consigliato personalizzato definito nelle impostazioni di configurazione per il modello. Ciascuno di questi campi memorizza dati dei suggerimenti avanzati per l'endpoint.

9. Utilizza la definizione dell'endpoint aggiornata per ciascun destinatario per sostituire ogni variabile di messaggio con il valore appropriato per tale destinatario.
10. Invia una versione del messaggio contenente i suggerimenti personalizzati per ciascun destinatario del messaggio.

Per personalizzare e migliorare le raccomandazioni in questo modo, crea una funzione Lambda che elabori le definizioni di endpoint inviate da Amazon Pinpoint e restituisca le definizioni aggiornate. Quindi, assegna una policy di funzione Lambda alla funzione e autorizza Amazon Pinpoint a richiamare la funzione. Quindi, configura il modello di raccomandazione in Amazon Pinpoint. Quando configuri il modello, specifica la funzione da richiamare e definisci gli attributi suggeriti da utilizzare.

# Creazione di una funzione Lambda

Per informazioni su come creare una funzione Lambda, consulta [Guida introduttiva](#) nella Guida per gli sviluppatori di AWS Lambda. Quando progetti e sviluppi la tua funzione, tiene a mente i requisiti e le linee guida seguenti.

## Dati dell'evento di input

Quando Amazon Pinpoint richiama una funzione Lambda per un modello di raccomandazione, invia un payload contenente la configurazione e altre impostazioni per la campagna o il percorso di invio del messaggio. Il payload include un oggetto `Endpoints`, ovvero una mappa che associa gli ID endpoint alle definizioni degli endpoint per i destinatari dei messaggi.

Le definizioni di endpoint utilizzano la struttura definita dalla risorsa [Endpoint](#) dell'API Amazon Pinpoint. Tuttavia, includono anche un campo per un attributo dinamico suggerito denominato `RecommendationItems`. Il campo `RecommendationItems` contiene uno o più elementi suggeriti per l'endpoint, come restituiti dalla campagna Amazon Personalize. Il valore di questo campo è un array ordinato contenente da 1 a 5 elementi suggeriti (come stringhe). Il numero di elementi nell'array dipende dal numero di elementi suggeriti configurati per il recupero da parte di Amazon Pinpoint per ciascun endpoint o utente.

Ad esempio:

```
"Endpoints": {
  "endpointIDexample-1":{
    "ChannelType":"EMAIL",
    "Address":"sofiam@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.875Z",
    "Attributes":{
      "AddressType":[
        "primary"
      ]
    },
    "User":{
      "UserId":"SofiaMartínez",
      "UserAttributes":{
        "LastName":[
          "Martínez"
        ],

```

```
        "FirstName":[
            "Sofia"
        ],
        "Neighborhood":[
            "East Bay"
        ]
    }
},
"RecommendationItems":[
    "1815",
    "2009",
    "1527"
],
"CreationDate":"2020-02-26T18:56:24.875Z"
},
"endpointIDexample-2":{
    "ChannelType":"EMAIL",
    "Address":"alejandr@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.897Z",
    "Attributes":{
        "AddressType":[
            "primary"
        ]
    },
    "User":{
        "UserId":"AlejandroRosalez",
        "UserAttributes":{
            "LastName":[
                "Rosalez"
            ],
            "FirstName":[
                "Alejandro"
            ],
            "Neighborhood":[
                "West Bay"
            ]
        }
    },
    "RecommendationItems":[
        "1210",
        "6542",
        "4582"
    ]
}
```

```
    ],  
    "CreationDate": "2020-02-26T18:56:24.897Z"  
  }  
}
```

Nell'esempio precedente, le impostazioni Amazon Pinpoint pertinenti sono:

- Il modello della funzione di suggerimento è configurato per recuperare tre elementi suggeriti per ogni endpoint o utente. (Il valore della proprietà `RecommendationsPerMessage` è impostato su 3.) Con questa impostazione, Amazon Pinpoint recupera e aggiunge solo i primi, secondi e terzi elementi suggeriti per ogni endpoint o utente.
- Il progetto è configurato per utilizzare attributi utente personalizzati che memorizzano il nome, il cognome e il quartiere in cui vive ogni utente. (L'oggetto `UserAttributes` contiene i valori per questi attributi.)
- Il progetto è configurato per utilizzare un attributo endpoint personalizzato (`AddressType`) che indica se l'endpoint è l'indirizzo (canale) preferito dall'utente per ricevere messaggi dal progetto. (L'oggetto `Attributes` contiene il valore per questo attributo.)

Quando Amazon Pinpoint richiama la funzione Lambda e invia il payload come dati dell'evento, AWS Lambda passa i dati alla funzione Lambda per l'elaborazione.

Ogni payload può contenere dati per un massimo di 50 endpoint. Se un segmento contiene più di 50 endpoint, Amazon Pinpoint richiama la funzione ripetutamente, per un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti i dati.

## Dati e requisiti di risposta

Quando progetti e sviluppi la funzione Lambda, considera le [quote per i modelli di machine learning](#). Se la funzione non soddisfa le condizioni definite da queste quote, Amazon Pinpoint non sarà in grado di elaborare e inviare il messaggio.

Tieni a mente anche i seguenti requisiti:

- La funzione deve restituire le definizioni degli endpoint aggiornate nello stesso formato fornito dai dati degli eventi di input.
- Ogni definizione di endpoint aggiornata può contenere da 1 a 10 attributi consigliati personalizzati per l'endpoint o l'utente. I nomi di questi attributi devono corrispondere ai nomi degli attributi specificati quando configuri il modello di raccomandazione in Amazon Pinpoint.

- Tutti gli attributi suggeriti personalizzati devono essere restituiti in un singolo oggetto `Recommendations` per ogni endpoint o utente. Questo requisito consente di garantire che i conflitti di denominazione non si verifichino. Puoi aggiungere l'oggetto `Recommendations` in qualsiasi posizione in una definizione di endpoint.
- Il valore per ogni attributo suggerito personalizzato deve essere una stringa (valore singolo) o una matrice di stringhe (valori multipli). Se il valore è un array di stringhe, è consigliabile mantenere l'ordine degli elementi suggeriti restituiti da Amazon Personalize, come indicato nel campo `RecommendationItems`. In caso contrario, il contenuto potrebbe non riflettere le previsioni del modello per un endpoint o un utente.
- La funzione non deve modificare altri elementi nei dati dell'evento, inclusi altri valori di attributo per un endpoint o un utente. Deve aggiungere e restituire solo valori per gli attributi consigliati personalizzati. Amazon Pinpoint non accetterà aggiornamenti a nessun altro valore nella risposta della funzione.
- La funzione deve essere ospitata nella stessa regione AWS del progetto Amazon Pinpoint che richiama la funzione. Se la funzione e il progetto non si trovano nella stessa regione, Amazon Pinpoint non è in grado di inviare i dati degli eventi alla funzione.

Se uno dei requisiti precedenti non viene soddisfatto, Amazon Pinpoint non sarà in grado di elaborare e inviare il messaggio a uno o più endpoint. Ciò potrebbe causare un fallimento di una campagna o di un'attività di viaggio.

Infine, si consiglia di riservare 256 esecuzioni simultanee per la funzione.

Nel complesso, la funzione Lambda deve elaborare i dati degli eventi inviati da Amazon Pinpoint e restituire le definizioni di endpoint modificate. È possibile eseguire questa operazione mediante l'iterazione di ciascun endpoint nell'oggetto `Endpoints` e, per ogni endpoint, creando e impostando valori per gli attributi suggeriti personalizzati che si desidera utilizzare. Il seguente handler di esempio, scritto in Python e che prosegue l'esempio precedente di dati degli eventi di input, mostra quanto segue:

```
import json
import string

def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
    print("Received context: " + str(context))
    segment_endpoints = event["Endpoints"]
    new_segment = dict()
```

```
for endpoint_id in segment_endpoints.keys():
    endpoint = segment_endpoints[endpoint_id]
    if supported_endpoint(endpoint):
        new_segment[endpoint_id] = add_recommendation(endpoint)

print("Returning endpoints: " + json.dumps(new_segment))
return new_segment

def supported_endpoint(endpoint):
    return True

def add_recommendation(endpoint):
    endpoint["Recommendations"] = dict()

    customTitleList = list()
    customGenreList = list()
    for i,item in enumerate(endpoint["RecommendationItems"]):
        item = int(item)
        if item == 1210:
            customTitleList.insert(i, "Hanna")
            customGenreList.insert(i, "Action")
        elif item == 1527:
            customTitleList.insert(i, "Catastrophe")
            customGenreList.insert(i, "Comedy")
        elif item == 1815:
            customTitleList.insert(i, "Fleabag")
            customGenreList.insert(i, "Comedy")
        elif item == 2009:
            customTitleList.insert(i, "Late Night")
            customGenreList.insert(i, "Drama")
        elif item == 4582:
            customTitleList.insert(i, "Agatha Christie\'s The ABC Murders")
            customGenreList.insert(i, "Crime")
        elif item == 6542:
            customTitleList.insert(i, "Hunters")
            customGenreList.insert(i, "Drama")

    endpoint["Recommendations"]["Title"] = customTitleList
    endpoint["Recommendations"]["Genre"] = customGenreList

    return endpoint
```

Nell'esempio precedente, AWS Lambda passa i dati dell'evento all'handler come parametro `event`. L'handler esegue l'iterazione di ciascun endpoint nell'oggetto `Endpoints` e imposta i valori per gli attributi suggeriti personalizzati denominati `Recommendations.Title` e `Recommendations.Genre`. L'istruzione `return` restituisce ogni definizione di endpoint aggiornata ad Amazon Pinpoint.

Continuando con l'esempio precedente di dati degli eventi di input, le definizioni degli endpoint aggiornati sono:

```
"Endpoints":{
  "endpointIDexample-1":{
    "ChannelType":"EMAIL",
    "Address":"sofiam@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.875Z",
    "Attributes":{
      "AddressType":[
        "primary"
      ]
    },
    "User":{
      "UserId":"SofiaMartínez",
      "UserAttributes":{
        "LastName":[
          "Martínez"
        ],
        "FirstName":[
          "Sofia"
        ],
        "Neighborhood":[
          "East Bay"
        ]
      }
    },
    "RecommendationItems":[
      "1815",
      "2009",
      "1527"
    ],
    "CreationDate":"2020-02-26T18:56:24.875Z",
    "Recommendations":{
      "Title":[
```

```
        "Fleabag",
        "Late Night",
        "Catastrophe"
    ],
    "Genre": [
        "Comedy",
        "Comedy",
        "Comedy"
    ]
}
},
"endpointIDexample-2": {
    "ChannelType": "EMAIL",
    "Address": "alejandr@example.com",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "EffectiveDate": "2020-02-26T18:56:24.897Z",
    "Attributes": {
        "AddressType": [
            "primary"
        ]
    },
    "User": {
        "UserId": "AlejandroRosalez",
        "UserAttributes": {
            "LastName": [
                "Rosalez"
            ],
            "FirstName": [
                "Alejandro"
            ],
            "Neighborhood": [
                "West Bay"
            ]
        }
    },
    "RecommendationItems": [
        "1210",
        "6542",
        "4582"
    ],
    "CreationDate": "2020-02-26T18:56:24.897Z",
    "Recommendations": {
        "Title": [
```

```
        "Hanna",
        "Hunters",
        "Agatha Christie\'s The ABC Murders"
    ],
    "Genre": [
        "Action",
        "Drama",
        "Crime"
    ]
}
}
```

Nell'esempio precedente, la funzione ha modificato l'oggetto `Endpoints` ricevuto e restituito i risultati. L'oggetto `Endpoint` per ogni endpoint ora contiene un nuovo oggetto `Recommendations`, che contiene i campi `Title` e `Genre`. Ciascuno di questi campi memorizza una matrice ordinata di tre valori (come stringhe), in cui ogni valore fornisce contenuto migliorato per un elemento suggerito corrispondente nel campo `RecommendationItems`.

## Assegnazione di una policy di funzione Lambda

Prima di poter utilizzare la funzione Lambda per elaborare i dati delle raccomandazioni, devi autorizzare Amazon Pinpoint a richiamare la funzione. Per concedere l'autorizzazione per l'invocazione, assegna una policy di funzione Lambda alla funzione. Una policy di funzione Lambda è una policy di autorizzazione basata su risorse che designa le entità che possono utilizzare la funzione e le azioni che tali entità possono eseguire. Per ulteriori informazioni, consulta [Uso delle policy basate sulle risorse per AWS Lambda](#) nella Guida per gli sviluppatori di AWS Lambda.

La policy di esempio seguente consente al principale del servizio Amazon Pinpoint di utilizzare l'azione `lambda:InvokeFunction` per una particolare campagna Amazon Pinpoint (*campaignId*) in un particolare progetto Amazon Pinpoint (*projectId*):

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
```

```

"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
}

```

La policy di funzione richiede un blocco `Condition` che includa una chiave `AWS:SourceArn`. Questa chiave specifica quale risorsa può invocare la funzione. Nell'esempio precedente, la policy consente a una particolare campagna di richiamare la funzione.

Puoi anche scrivere una policy che consenta al principale del servizio Amazon Pinpoint di utilizzare l'azione `lambda:InvokeFunction` per tutte le campagne e tutti i percorsi in un progetto Amazon Pinpoint specifico (*projectId*). Di seguito viene illustrato un esempio di policy che mostra questo approccio:

```

{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
}

```

A differenza del primo esempio, la chiave `AWS:SourceArn` nel blocco `Condition` di questo esempio consente a un particolare progetto di invocare la funzione. Questa autorizzazione si applica a tutte le campagne e viaggi nel progetto.

Per scrivere una policy più generica, puoi utilizzare un carattere jolly di corrispondenza con più caratteri (\*). Ad esempio, puoi utilizzare il seguente blocco `Condition` per consentire a qualsiasi progetto Amazon Pinpoint di richiamare la funzione:

```

"Condition": {
  "ArnLike": {

```

```

    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}

```

Se desideri utilizzare la funzione Lambda con tutti i progetti per l'account Amazon Pinpoint, è consigliabile configurare il blocco `Condition` della policy come indicato in precedenza. Tuttavia, come procedura consigliata, dovresti creare policy che includano solo le autorizzazioni necessarie per eseguire un'operazione specifica su una risorsa specifica.

## Autorizzazione di Amazon Pinpoint a richiamare la funzione

Dopo aver assegnato una policy di funzione Lambda alla funzione, puoi aggiungere autorizzazioni che consentono ad Amazon Pinpoint di richiamare la funzione per un progetto, una campagna o un percorso specifico. A tale scopo, utilizza la AWS Command Line Interface (AWS CLI) e il comando Lambda [add-permission](#). L'esempio seguente mostra come eseguire questa operazione per un progetto specifico (*projectId*):

```

$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*

```

L'esempio precedente è formattato per Unix, Linux e macOS. Per Microsoft Windows, sostituire il carattere di continuazione della riga (`\`) con un accento circonflesso (`^`).

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```

{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
          \"arn:aws:mobiletargeting:us-east-1:111122223333:recommenders/*\"}}}"
}

```

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy della funzione Lambda.

## Configurazione del modello della funzione di suggerimento

Per configurare Amazon Pinpoint per richiamare la funzione Lambda per un modello di raccomandazione, specifica le seguenti impostazioni di configurazione specifiche di Lambda per il modello:

- `RecommendationTransformerUri`: questa proprietà specifica il nome o il nome della risorsa Amazon (ARN) della funzione Lambda.
- `Attributes`: questo oggetto è una mappa che definisce gli attributi consigliati personalizzati che la funzione aggiunge a ciascuna definizione di endpoint. Ciascuno di questi attributi può essere utilizzato come variabile di messaggio in un modello di messaggio.

Puoi specificare queste impostazioni utilizzando la risorsa [Modelli di raccomandazione](#) dell'API Amazon Pinpoint (quando crei la configurazione per un modello) o la risorsa [Modello di raccomandazione](#) dell'API Amazon Pinpoint (se aggiorni la configurazione per un modello). Puoi inoltre definire queste impostazioni utilizzando la console Amazon Pinpoint.

Per ulteriori informazioni sull'utilizzo dei modelli di raccomandazione con Amazon Pinpoint, consulta [Modelli di machine learning](#) nella Guida per l'utente di Amazon Pinpoint.

# Eliminazione di dati da Amazon Pinpoint

A seconda di come viene utilizzato, Amazon Pinpoint può archiviare determinati dati che potrebbero essere considerati personali. Ad esempio, un endpoint in Amazon Pinpoint contiene le informazioni di contatto di un utente finale, come l'indirizzo e-mail o il numero di cellulare.

Puoi utilizzare la console o l'API Amazon Pinpoint per eliminare definitivamente i dati personali. Questo argomento illustra le procedure per l'eliminazione di vari tipi di dati che possono essere considerati dati personali.

## Eliminazione di endpoint

Un endpoint rappresenta un metodo per contattare uno dei clienti. Ogni endpoint può fare riferimento all'indirizzo e-mail di un cliente, all'identificatore del dispositivo mobile, al numero di telefono o ad un altro tipo di destinazione a cui è possibile inviare messaggi. In molte giurisdizioni, le informazioni di questo tipo sono considerate personali.

Per eliminare tutti i dati per un determinato endpoint, è possibile utilizzare l'API Amazon Pinpoint per eliminare l'endpoint. Nella procedura seguente viene spiegato come eliminare un endpoint utilizzando l'AWS CLI per interagire con l'API Amazon Pinpoint. Questa procedura si basa sul presupposto che l'AWS CLI sia già stata installata e configurata. Per ulteriori informazioni, consulta [Installazione dell'AWS CLI](#) nella Guida per l'utente dell'AWS Command Line Interface .

Per eliminare un endpoint utilizzando AWS CLI

- Nella riga di comando, inserisci il comando seguente:

```
aws pinpoint delete-endpoint --application-id 810c7aab86d42fb2b56c8c966example --  
endpoint-id ad015a3bf4f1b2b0b82example
```

Nel comando precedente, sostituisci *810c7aab86d42fb2b56c8c966example* con l'ID del progetto a cui è associato l'endpoint. Sostituisci anche *ad015a3bf4f1b2b0b82example* con l'ID univoco dell'endpoint.

Per trovare l'ID di un determinato endpoint, stabilisci a quale segmento appartiene l'endpoint e quindi esporta il segmento da Amazon Pinpoint. I dati esportati includono l'ID endpoint per ciascun endpoint. È possibile esportare un segmento in un file utilizzando la console Amazon Pinpoint. Per

ulteriori informazioni, consulta [Esportazione di segmenti](#) nella Guida per l'utente di Amazon Pinpoint. Puoi anche esportare un segmento in un bucket Amazon Simple Storage Service (Amazon S3) tramite l'API Amazon Pinpoint. Per la procedura da seguire a questo scopo, consulta [Esportazione di endpoint](#) in questa guida.

## Eliminazione dei dati relativi a segmenti ed endpoint archiviati in Amazon S3

È possibile importare segmenti da un file archiviato in un bucket Amazon S3 utilizzando la console o l'API Amazon Pinpoint. È anche possibile esportare dati relativi all'applicazione, ai segmenti o agli endpoint da Amazon Pinpoint in un bucket Amazon S3. I file importati ed esportati possono contenere dati personali, tra cui indirizzi e-mail, numeri di cellulare e informazioni sulla posizione fisica di un endpoint. È possibile eliminare questi file da Amazon S3.

I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi [Come svuotare un bucket S3?](#) o [Come eliminare un bucket S3?](#).

## Eliminazione di tutti i dati del progetto

È possibile eliminare definitivamente tutti i dati archiviati per un progetto Amazon Pinpoint. È possibile farlo eliminando il progetto.

### Warning

Se elimini un progetto, Amazon Pinpoint elimina tutte le impostazioni specifiche e i dati del progetto. Le informazioni non possono essere recuperate.

Quando elimini un progetto, Amazon Pinpoint elimina tutte le impostazioni specifiche del progetto per la notifica push e i canali di messaggistica SMS bidirezionale, nonché tutti i segmenti, le campagne, i percorsi e i dati analitici specifici del progetto archiviati in Amazon Pinpoint, ad esempio:

- **Segmenti:** tutte le impostazioni e i dati dei segmenti. Per segmenti dinamici, questo include i gruppi di segmenti e i filtri definiti. Per segmenti importati, questo include endpoint, ID utente e altri dati importati, nonché gli eventuali filtri applicati.
- **Campagne:** tutti i messaggi, i trattamenti e le variabili dei messaggi, i dati analitici, le pianificazioni e altre impostazioni.

- Percorsi: tutte le attività, i dati analitici, gli orari e altre impostazioni.
- Analisi: dati analitici per tutte le metriche di coinvolgimento, ad esempio il numero di messaggi inviati e recapitati per campagne e percorsi e tutte le metriche di esecuzione dei percorsi. Per le app mobili e Web, tutti i dati sugli eventi che non sono stati trasmessi in streaming a un altro AWS servizio come Amazon Kinesis, tutte le canalizzazioni e i dati relativi all'utilizzo delle applicazioni, alle entrate e ai parametri demografici. Prima di eliminare un progetto, ti consigliamo di esportare questi dati in un'altra posizione.

Puoi eliminare un progetto utilizzando la console Amazon Pinpoint. Per ulteriori informazioni, consulta [Eliminazione di un progetto](#) nella Guida per l'utente di Amazon Pinpoint. Puoi anche eliminare un progetto a livello di programmazione utilizzando la risorsa [App](#) dell'API Amazon Pinpoint.

## Eliminazione di tutti i dati chiudendo l'account AWS

Per eliminare tutti i dati archiviati in Amazon Pinpoint, è anche possibile chiudere l'account AWS . Tuttavia, questa azione elimina anche tutti gli altri dati, personali o non personali, che hai archiviato in ogni altro servizio. AWS Dopo il periodo successivo alla chiusura, chiude AWS definitivamente il tuo AWS account e non puoi più riaprirlo. Tutti i contenuti che non hai eliminato vengono eliminati definitivamente e tutti i servizi che non hai interrotto vengono interrotti. Per ulteriori informazioni, consulta [Chiudere un AWS account](#) nella Guida AWS Account Management di riferimento.

### Warning

La procedura seguente rimuove completamente tutti i dati archiviati nel tuo AWS account in tutti i AWS servizi e le AWS aree geografiche.

Puoi chiudere il tuo AWS account utilizzando il AWS Management Console.

Per chiudere il tuo AWS account

1. Apri il file AWS Management Console all'[indirizzo https://console.aws.amazon.com](https://console.aws.amazon.com).
2. Passa alla pagina Account Settings (Impostazioni account) all'[indirizzo https://console.aws.amazon.com/billing/home?#/account](https://console.aws.amazon.com/billing/home?#/account).

 **Warning**

I seguenti passaggi eliminano definitivamente tutti i dati archiviati in tutti i AWS servizi in tutte le AWS regioni.

3. Nella sezione Chiudi account, leggi il disclaimer che descrive le conseguenze della chiusura dell'account AWS . Per accettare le condizioni, seleziona la casella di controllo e quindi seleziona Close Account (Chiudi account).
4. Nella finestra di dialogo di conferma, selezionare Close Account (Chiudi account).

# Esempi di codice per Amazon Pinpoint con AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare Amazon Pinpoint con un Software Development Kit (SDK) AWS.

Per un elenco completo delle guide per gli sviluppatori di SDK AWS ed esempi di codice, consulta la sezione [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Esempi di codice

- [Esempi di codice per Amazon Pinpoint con SDK AWS](#)
  - [Azioni per Amazon Pinpoint tramite SDK AWS](#)
    - [Utilizzo CreateApp con un AWS SDK o una CLI](#)
    - [Utilizzo CreateCampaign con un AWS SDK o una CLI](#)
    - [Utilizzo CreateExportJob con un AWS SDK o una CLI](#)
    - [Utilizzo CreateImportJob con un AWS SDK o una CLI](#)
    - [Utilizzo CreateSegment con un AWS SDK o una CLI](#)
    - [Utilizzo DeleteApp con un AWS SDK o una CLI](#)
    - [Utilizzo DeleteEndpoint con un AWS SDK o una CLI](#)
    - [Utilizzo GetEndpoint con un AWS SDK o una CLI](#)
    - [Utilizzo GetSegments con un AWS SDK o una CLI](#)
    - [Utilizzo GetSmsChannel con un AWS SDK o una CLI](#)
    - [Utilizzo GetUserEndpoints con un AWS SDK o una CLI](#)
    - [Utilizzo SendMessages con un AWS SDK o una CLI](#)
    - [Utilizzo UpdateEndpoint con un AWS SDK o una CLI](#)
- [Esempi di codice per l'API Amazon Pinpoint SMS and Voice utilizzando SDK AWS](#)
  - [Azioni per l'API SMS e Voice di Amazon Pinpoint tramite SDK AWS](#)
    - [Utilizzo SendVoiceMessage con un AWS SDK o una CLI](#)

## Esempi di codice per Amazon Pinpoint con SDK AWS

I seguenti esempi di codice mostrano come usare Amazon Pinpoint con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. [Utilizzo di Amazon Pinpoint con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Esempi di codice

- [Azioni per Amazon Pinpoint tramite SDK AWS](#)
  - [Utilizzo CreateApp con un AWS SDK o una CLI](#)
  - [Utilizzo CreateCampaign con un AWS SDK o una CLI](#)
  - [Utilizzo CreateExportJob con un AWS SDK o una CLI](#)
  - [Utilizzo CreateImportJob con un AWS SDK o una CLI](#)
  - [Utilizzo CreateSegment con un AWS SDK o una CLI](#)
  - [Utilizzo DeleteApp con un AWS SDK o una CLI](#)
  - [Utilizzo DeleteEndpoint con un AWS SDK o una CLI](#)
  - [Utilizzo GetEndpoint con un AWS SDK o una CLI](#)
  - [Utilizzo GetSegments con un AWS SDK o una CLI](#)
  - [Utilizzo GetSmsChannel con un AWS SDK o una CLI](#)
  - [Utilizzo GetUserEndpoints con un AWS SDK o una CLI](#)
  - [Utilizzo SendMessages con un AWS SDK o una CLI](#)
  - [Utilizzo UpdateEndpoint con un AWS SDK o una CLI](#)

## Azioni per Amazon Pinpoint tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni Amazon Pinpoint con AWS gli SDK. Queste parti di codice chiamano l'API Amazon Pinpoint e provengono da programmi più grandi che devono essere eseguiti nel contesto. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la [documentazione di riferimento dell'API Amazon Pinpoint](#).

## Esempi

- [Utilizzo CreateApp con un AWS SDK o una CLI](#)
- [Utilizzo CreateCampaign con un AWS SDK o una CLI](#)
- [Utilizzo CreateExportJob con un AWS SDK o una CLI](#)
- [Utilizzo CreateImportJob con un AWS SDK o una CLI](#)
- [Utilizzo CreateSegment con un AWS SDK o una CLI](#)
- [Utilizzo DeleteApp con un AWS SDK o una CLI](#)
- [Utilizzo DeleteEndpoint con un AWS SDK o una CLI](#)
- [Utilizzo GetEndpoint con un AWS SDK o una CLI](#)
- [Utilizzo GetSegments con un AWS SDK o una CLI](#)
- [Utilizzo GetSmsChannel con un AWS SDK o una CLI](#)
- [Utilizzo GetUserEndpoints con un AWS SDK o una CLI](#)
- [Utilizzo SendMessages con un AWS SDK o una CLI](#)
- [Utilizzo UpdateEndpoint con un AWS SDK o una CLI](#)

## Utilizzo **CreateApp** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `CreateApp`.

### CLI

#### AWS CLI

Esempio 1: creazione di un'applicazione

L'esempio `create-app` seguente crea una nuova applicazione (progetto).

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp
```

Output:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",
```

```
    "tags": {}  
  }  
}
```

## Esempio 2: creazione di un'applicazione con tag

L'esempio `create-app` seguente crea una nuova applicazione (progetto) e associa un tag (chiave e valore) all'applicazione.

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

## Output:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {  
      "Stack": "Test"  
    }  
  }  
}
```

- Per i dettagli sull'API, consulta [CreateApp AWS CLI Command Reference](#).

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.pinpoint.PinpointClient;  
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = ""

            Usage: <appName>

            Where:
            appName - The name of the application to create.

            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }

    public static String createApplication(PinpointClient pinpoint, String
appName) {
        try {
```

```

        CreateApplicationRequest appRequest =
CreateApplicationRequest.builder()
        .name(appName)
        .build();

        CreateAppRequest request = CreateAppRequest.builder()
        .createApplicationRequest(appRequest)
        .build();

        CreateAppResponse result = pinpoint.createApp(request);
        return result.applicationResponse().id();

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
}

```

- Per i dettagli sull'API, consulta la [CreateApp](#) sezione AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

suspend fun createApplication(applicationName: String?): String? {

    val createApplicationRequestObj = CreateApplicationRequest {
        name = applicationName
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.createApp(
            CreateAppRequest {

```

```
        createApplicationRequest = createApplicationRequestOb  
    }  
    )  
    return result.applicationResponse?.id  
    }  
}
```

- Per i dettagli sull'API, [CreateApp](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **CreateCampaign** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `CreateCampaign`.

### Java

#### SDK per Java 2.x

#### Note

C'è altro su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea una campagna.

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.pinpoint.PinpointClient;  
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;  
import software.amazon.awssdk.services.pinpoint.model.Message;  
import software.amazon.awssdk.services.pinpoint.model.Schedule;  
import software.amazon.awssdk.services.pinpoint.model.Action;  
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;  
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;  
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;  
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;  
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = ""

            Usage:  <appId> <segmentId>

            Where:
                appId - The ID of the application to create the campaign in.
                segmentId - The ID of the segment to create the campaign from.
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId,
String segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }
}
```

```
public static CampaignResponse createCampaign(PinpointClient client, String
appID, String segmentID) {

    try {
        Schedule schedule = Schedule.builder()
            .startTime("IMMEDIATE")
            .build();

        Message defaultMessage = Message.builder()
            .action(Action.OPEN_APP)
            .body("My message body.")
            .title("My message title.")
            .build();

        MessageConfiguration messageConfiguration =
MessageConfiguration.builder()
            .defaultMessage(defaultMessage)
            .build();

        WriteCampaignRequest request = WriteCampaignRequest.builder()
            .description("My description")
            .schedule(schedule)
            .name("MyCampaign")
            .segmentId(segmentID)
            .messageConfiguration(messageConfiguration)
            .build();

        CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
            .applicationId(appID)
            .writeCampaignRequest(request).build());

        System.out.println("Campaign ID: " + result.campaignResponse().id());
        return result.campaignResponse();

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return null;
}
}
```

- Per i dettagli sull'API, [CreateCampaign](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun createPinCampaign(appId: String, segmentIdVal: String) {

    val schedule0b = Schedule {
        startTime = "IMMEDIATE"
    }

    val defaultMessage0b = Message {
        action = Action.OpenApp
        body = "My message body"
        title = "My message title"
    }

    val messageConfiguration0b = MessageConfiguration {
        defaultMessage = defaultMessage0b
    }

    val writeCampaign = WriteCampaignRequest {
        description = "My description"
        schedule = schedule0b
        name = "MyCampaign"
        segmentId = segmentIdVal
        messageConfiguration = messageConfiguration0b
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse = pinpoint.createCampaign(
            CreateCampaignRequest {
                applicationId = appId
            }
        )
    }
}
```

```
        writeCampaignRequest = writeCampaign
    }
)
println("Campaign ID is ${result.campaignResponse?.id}")
}
}
```

- Per i dettagli sull'API, [CreateCampaign](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **CreateExportJob** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare `CreateExportJob`.

### Java

#### SDK per Java 2.x

#### Note

C'è altro su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Esporta un endpoint.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
```

```
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = ""

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and
                prints them.

                Usage: ExportEndpoints <applicationId> <s3BucketName>
                <iamExportRoleArn> <path>

                Where:
```

```
        applicationId - The ID of the Amazon Pinpoint application that
has the endpoint.
        s3BucketName - The name of the Amazon S3 bucket to export the
JSON file to.\s
        iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket. path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
        """;

    if (args.length != 4) {
        System.out.println(usage);
        System.exit(1);
    }

    String applicationId = args[0];
    String s3BucketName = args[1];
    String iamExportRoleArn = args[2];
    String path = args[3];
    System.out.println("Deleting an application with ID: " + applicationId);

    Region region = Region.US_EAST_1;
    PinpointClient pinpoint = PinpointClient.builder()
        .region(region)
        .build();

    S3Client s3Client = S3Client.builder()
        .region(region)
        .build();

    exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
    pinpoint.close();
    s3Client.close();
}

public static void exportAllEndpoints(PinpointClient pinpoint,
    S3Client s3Client,
    String applicationId,
    String s3BucketName,
    String path,
    String iamExportRoleArn) {

    try {
```

```

        List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
s3BucketName, iamExportRoleArn,
            applicationId);
        List<String> endpointFileKeys = objectKeys.stream().filter(o ->
o.endsWith(".gz"))
            .collect(Collectors.toList());
        downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static List<String> exportEndpointsToS3(PinpointClient pinpoint,
S3Client s3Client, String s3BucketName,
    String iamExportRoleArn, String applicationId) {

    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
    String endpointsKeyPrefix = "exports/" + applicationId + "_" +
dateFormat.format(new Date());
    String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
"/";

    List<String> objectKeys = new ArrayList<>();
    String key;

    try {
        // Defines the export job that Amazon Pinpoint runs.
        ExportJobRequest jobRequest = ExportJobRequest.builder()
            .roleArn(iamExportRoleArn)
            .s3UrlPrefix(s3UrlPrefix)
            .build();

        CreateExportJobRequest exportJobRequest =
CreateExportJobRequest.builder()
            .applicationId(applicationId)
            .exportJobRequest(jobRequest)
            .build();

        System.out.format("Exporting endpoints from Amazon Pinpoint
application %s to Amazon S3 " +
            "bucket %s . . .\n", applicationId, s3BucketName);
    }
}

```

```
        CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
        String jobId = exportResult.exportJobResponse().id();
        System.out.println(jobId);
        printExportJobStatus(pinpoint, applicationId, jobId);

        ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                .bucket(s3BucketName)
                .prefix(endpointsKeyPrefix)
                .build();

        // Create a list of object keys.
        ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
        List<S3Object> objects = v2Response.contents();
        for (S3Object object : objects) {
            key = object.key();
            objectKeys.add(key);
        }

        return objectKeys;

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return null;
}

private static void printExportJobStatus(PinpointClient pinpointClient,
        String applicationId,
        String jobId) {

    GetExportJobResponse getExportJobResult;
    String status;

    try {
        // Checks the job status until the job completes or fails.
        GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                .jobId(jobId)
                .applicationId(applicationId)
                .build();

        do {
```

```
        getExportJobResult =
pinpointClient.getExportJob(exportJobRequest);
        status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
        System.out.format("Export job %s . . .\n", status);
        TimeUnit.SECONDS.sleep(3);

    } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

    if (status.equals("COMPLETED")) {
        System.out.println("Finished exporting endpoints.");
    } else {
        System.err.println("Failed to export endpoints.");
        System.exit(1);
    }

} catch (PinpointException | InterruptedException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}

// Download files from an Amazon S3 bucket and write them to the path
location.
public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

    String newPath;
    try {
        for (String key : objectKeys) {
            GetObjectRequest objectRequest = GetObjectRequest.builder()
                .bucket(s3BucketName)
                .key(key)
                .build();

            ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
            byte[] data = objectBytes.asByteArray();

            // Write the data to a local file.
            String fileSuffix = new
SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
            newPath = path + fileSuffix + ".gz";
            File myFile = new File(newPath);
```

```
        OutputStream os = new FileOutputStream(myFile);
        os.write(data);
    }
    System.out.println("Download finished.");

    } catch (S3Exception | NullPointerException | IOException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [CreateExportJob](#) consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **CreateImportJob** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare `CreateImportJob`.

Java

SDK per Java 2.x

### Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Importa un segmento.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = ""

            Usage:  <appId> <bucket> <key> <roleArn>\s

            Where:
                appId - The application ID to create a segment for.
                bucket - The name of the Amazon S3 bucket that contains the
segment definitons.
                key - The key of the S3 object.
                roleArn - ARN of the role that allows Amazon
Pinpoint to access S3. You need to set trust management for this
to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\_policies\_elements\_principal.html
            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
key, roleArn);
    }
}
```

```
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for
import job status.");
        System.out.println("See application " + response.jobStatus() + " for
import job status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
        String appId,
        String bucket,
        String key,
        String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                .defineSegment(true)
                .registerEndpoints(true)
                .roleArn(roleArn)
                .format(Format.JSON)
                .s3Url("s3://" + bucket + "/" + key)
                .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                .importJobRequest(importRequest)
                .applicationId(appId)
                .build();

            CreateImportJobResponse jobResponse =
client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Per i dettagli sull'API, [CreateImportJob](#) consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **CreateSegment** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `CreateSegment`.

### Java

#### SDK per Java 2.x

#### Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 */
```

```

* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class CreateSegment {
    public static void main(String[] args) {
        final String usage = ""

                Usage:  <appId>

                Where:
                    appId - The application ID to create a segment
for.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        SegmentResponse result = createSegment(pinpoint, appId);
        System.out.println("Segment " + result.name() + " created.");
        System.out.println(result.segmentType());
        pinpoint.close();
    }

    public static SegmentResponse createSegment(PinpointClient client, String
appId) {
        try {
            Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
            segmentAttributes.put("Team",
AttributeDimension.builder()
                .attributeType(AttributeType.INCLUSIVE)
                .values("Lakers")
                .build());

            RecencyDimension recencyDimension =
RecencyDimension.builder()

```

```
                .duration("DAY_30")
                .recencyType("ACTIVE")
                .build();

        SegmentBehaviors segmentBehaviors =
SegmentBehaviors.builder()

                .recency(recencyDimension)
                .build();

        SegmentDemographics segmentDemographics =
SegmentDemographics

                .builder()
                .build();

        SegmentLocation segmentLocation = SegmentLocation
                .builder()
                .build();

        SegmentDimensions dimensions = SegmentDimensions
                .builder()
                .attributes(segmentAttributes)
                .behavior(segmentBehaviors)
                .demographic(segmentDemographics)
                .location(segmentLocation)
                .build();

        WriteSegmentRequest writeSegmentRequest =
WriteSegmentRequest.builder()

                .name("MySegment")
                .dimensions(dimensions)
                .build();

        CreateSegmentRequest createSegmentRequest =
CreateSegmentRequest.builder()

                .applicationId(appId)
                .writeSegmentRequest(writeSegmentRequest)
                .build();

        CreateSegmentResponse createSegmentResult =
client.createSegment(createSegmentRequest);
        System.out.println("Segment ID: " +
createSegmentResult.segmentResponse().id());
        System.out.println("Done");
        return createSegmentResult.segmentResponse();
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Per i dettagli sull'API, [CreateSegment](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {

    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts = AttributeDimension {
        attributeType = AttributeType.Inclusive
        values = myList
    }

    segmentAttributes["Team"] = atts
    val recencyDimension = RecencyDimension {
        duration = Duration.fromValue("DAY_30")
        recencyType = RecencyType.fromValue("ACTIVE")
    }

    val segmentBehaviors = SegmentBehaviors {
        recency = recencyDimension
    }
}
```

```

val segmentLocation = SegmentLocation {}
val dimensions0b = SegmentDimensions {
    attributes = segmentAttributes
    behavior = segmentBehaviors
    demographic = SegmentDemographics {}
    location = segmentLocation
}

val writeSegmentRequest0b = WriteSegmentRequest {
    name = "MySegment101"
    dimensions = dimensions0b
}

PinpointClient { region = "us-west-2" }.use { pinpoint ->
    val createSegmentResult: CreateSegmentResponse = pinpoint.createSegment(
        CreateSegmentRequest {
            applicationId = applicationIdVal
            writeSegmentRequest = writeSegmentRequest0b
        }
    )
    println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
    return createSegmentResult.segmentResponse?.id
}
}

```

- Per i dettagli sull'API, [CreateSegment](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **DeleteApp** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `DeleteApp`.

### CLI

#### AWS CLI

#### Eliminazione di un'applicazione

L'esempio delete-app seguente elimina un'applicazione (progetto).

```
aws pinpoint delete-app \  
  --application-id 810c7aab86d42fb2b56c8c966example
```

Output:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {}  
  }  
}
```

- Per i dettagli sull'API, consulta [DeleteApp AWS CLI Command Reference](#).

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Elimina un'applicazione.

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.pinpoint.PinpointClient;  
import software.amazon.awssdk.services.pinpoint.model.DeleteAppRequest;  
import software.amazon.awssdk.services.pinpoint.model.DeleteAppResponse;  
import software.amazon.awssdk.services.pinpoint.model.PinpointException;  
  
/**  
 * Before running this Java V2 code example, set up your development  
 * environment, including your credentials.  
 */
```

```
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class DeleteApp {
    public static void main(String[] args) {
        final String usage = ""

            Usage: <appId>

            Where:
                appId - The ID of the application to delete.

            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        System.out.println("Deleting an application with ID: " + appId);
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        deletePinApp(pinpoint, appId);
        System.out.println("Done");
        pinpoint.close();
    }

    public static void deletePinApp(PinpointClient pinpoint, String appId) {
        try {
            DeleteAppRequest appRequest = DeleteAppRequest.builder()
                .applicationId(appId)
                .build();

            DeleteAppResponse result = pinpoint.deleteApp(appRequest);
            String appName = result.applicationResponse().name();
            System.out.println("Application " + appName + " has been deleted.");
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }
}
```

```
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [DeleteApp](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun deletePinApp(appId: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteApp(
            DeleteAppRequest {
                applicationId = appId
            }
        )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- Per i dettagli sull'API, [DeleteApp](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **DeleteEndpoint** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `DeleteEndpoint`.

### Java

#### SDK per Java 2.x

#### Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

#### Eliminare un endpoint.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = ""

                Usage:  <appName> <endpointId >

                Where:
                appId - The id of the application to delete.
                endpointId - The id of the endpoint to delete.
                """;
    }
}
```

```
    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String appId = args[0];
    String endpointId = args[1];
    System.out.println("Deleting an endpoint with id: " + endpointId);
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    deletePinEndpoint(pinpoint, appId, endpointId);
    pinpoint.close();
}

public static void deletePinEndpoint(PinpointClient pinpoint, String appId,
String endpointId) {
    try {
        DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
            .applicationId(appId)
            .endpointId(endpointId)
            .build();

        DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
        String id = result.endpointResponse().id();
        System.out.println("The deleted endpoint id " + id);

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    System.out.println("Done");
}
}
```

- Per i dettagli sull'API, [DeleteEndpoint](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun deletePinEndpoint(appIdVal: String?, endpointIdVal: String?) {  
  
    val deleteEndpointRequest = DeleteEndpointRequest {  
        applicationId = appIdVal  
        endpointId = endpointIdVal  
    }  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)  
        val id = result.endpointResponse?.id  
        println("The deleted endpoint is $id")  
    }  
}
```

- Per i dettagli sull'API, [DeleteEndpoint](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

### Utilizzo **GetEndpoint** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `GetEndpoint`.

#### CLI

##### AWS CLI

Recupero di informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione

L'esempio `get-endpoint` seguente recupera le informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione.

```
aws pinpoint get-endpoint \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --endpoint-id testendpoint \  
  --region us-east-1
```

Output:

```
{  
  "EndpointResponse": {  
    "Address": "+11234567890",  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "Attributes": {},  
    "ChannelType": "SMS",  
    "CohortId": "63",  
    "CreationDate": "2019-01-28T23:55:11.534Z",  
    "EffectiveDate": "2021-08-06T00:04:51.763Z",  
    "EndpointStatus": "ACTIVE",  
    "Id": "testendpoint",  
    "Location": {  
      "Country": "USA"  
    },  
    "Metrics": {  
      "SmsDelivered": 1.0  
    },  
    "OptOut": "ALL",  
    "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",  
    "User": {  
      "UserAttributes": {  
        "Age": [  
          "24"  
        ]  
      },  
      "UserId": "testuser"  
    }  
  }  
}
```

- Per i dettagli sull'API, consulta [GetEndpoint AWS CLI Command Reference](#).

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = ""

                Usage:  <appId> <endpoint>

                Where:
                    appId - The ID of the application to delete.
                    endpoint - The ID of the endpoint.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
        }
    }
}
```

```
        System.exit(1);
    }

    String appId = args[0];
    String endpoint = args[1];
    System.out.println("Looking up an endpoint point with ID: " + endpoint);
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    lookupPinpointEndpoint(pinpoint, appId, endpoint);
    pinpoint.close();
}

public static void lookupPinpointEndpoint(PinpointClient pinpoint, String
appId, String endpoint) {
    try {
        GetEndpointRequest appRequest = GetEndpointRequest.builder()
            .applicationId(appId)
            .endpointId(endpoint)
            .build();

        GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
        EndpointResponse endResponse = result.endpointResponse();

        // Uses the Google Gson library to pretty print the endpoint JSON.
        Gson gson = new GsonBuilder()
            .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
            .setPrettyPrinting()
            .create();

        String endpointJson = gson.toJson(endResponse);
        System.out.println(endpointJson);

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    System.out.println("Done");
}
}
```

- Per i dettagli sull'API, [GetEndpoint](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun lookupPinpointEndpoint(appId: String?, endpoint: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.getEndpoint(
            GetEndpointRequest {
                applicationId = appId
                endpointId = endpoint
            }
        )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson = GsonBuilder()
            .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
            .setPrettyPrinting()
            .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Per i dettagli sull'API, [GetEndpoint](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **GetSegments** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `GetSegments`.

### Java

#### SDK per Java 2.x

##### Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Elenca i segmenti.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListSegments {
    public static void main(String[] args) {
        final String usage = ""

            Usage:    <appId>

            Where:
                appId - The ID of the application that contains a segment.
    }
}
```

```
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String appId = args[0];
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    listSegs(pinpoint, appId);
    pinpoint.close();
}

public static void listSegs(PinpointClient pinpoint, String appId) {
    try {
        GetSegmentsRequest request = GetSegmentsRequest.builder()
            .applicationId(appId)
            .build();

        GetSegmentsResponse response = pinpoint.getSegments(request);
        List<SegmentResponse> segments = response.segmentsResponse().item();
        for (SegmentResponse segment : segments) {
            System.out
                .println("Segment " + segment.id() + " " +
segment.name() + " " + segment.lastModifiedDate());
        }

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [GetSegments](#) consulta AWS SDK for Java 2.x API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun listSegs(appId: String?) {  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
  
        val response = pinpoint.getSegments(  
            GetSegmentsRequest {  
                applicationId = appId  
            }  
        )  
        response.segmentsResponse?.item?.forEach { segment ->  
            println("Segement id is ${segment.id}")  
        }  
    }  
}
```

- Per i dettagli sull'API, [GetSegments](#) consulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

### Utilizzo **GetSmsChannel** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `GetSmsChannel`.

## CLI

### AWS CLI

Recupero delle informazioni sullo stato e sulle impostazioni del canale SMS per un'applicazione

L'esempio `get-sms-channel` seguente recupera lo stato e le impostazioni del canale di sms per un'applicazione.

```
aws pinpoint get-sms-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Output:

```
{  
  "SMSChannelResponse": {  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "CreationDate": "2019-10-08T18:39:18.511Z",  
    "Enabled": true,  
    "Id": "sms",  
    "IsArchived": false,  
    "LastModifiedDate": "2019-10-08T18:39:18.511Z",  
    "Platform": "SMS",  
    "PromotionalMessagesPerSecond": 20,  
    "TransactionalMessagesPerSecond": 20,  
    "Version": 1  
  }  
}
```

- Per i dettagli sull'API, consulta [GetSmsChannel AWS CLI Command Reference](#).

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class UpdateChannel {
    public static void main(String[] args) {
        final String usage = ""

                Usage: CreateChannel <appId>

                Where:
                appId - The name of the application whose channel is updated.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        SMSChannelResponse getResponse = getSmsChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }
}
```

```
private static SMSChannelResponse getSMSChannel(PinpointClient client, String
appId) {
    try {
        GetSmsChannelRequest request = GetSmsChannelRequest.builder()
            .applicationId(appId)
            .build();

        SMSChannelResponse response =
client.getSmsChannel(request).smsChannelResponse();
        System.out.println("Channel state is " + response.enabled());
        return response;

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return null;
}

private static void toggleSmsChannel(PinpointClient client, String appId,
SMSChannelResponse getResponse) {
    boolean enabled = !getResponse.enabled();
    try {
        SMSChannelRequest request = SMSChannelRequest.builder()
            .enabled(enabled)
            .build();

        UpdateSmsChannelRequest updateRequest =
UpdateSmsChannelRequest.builder()
            .smsChannelRequest(request)
            .applicationId(appId)
            .build();

        UpdateSmsChannelResponse result =
client.updateSmsChannel(updateRequest);
        System.out.println("Channel state: " +
result.smsChannelResponse().enabled());

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
}
```

- Per i dettagli sull'API, [GetSmsChannel](#) consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **GetUserEndpoints** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare `GetUserEndpoints`.

Java

SDK per Java 2.x

### Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListEndpointIds {
```

```
public static void main(String[] args) {
    final String usage = ""

        Usage:    <applicationId> <userId>

        Where:
            applicationId - The ID of the Amazon Pinpoint application that
has the endpoint.
            userId - The user id applicable to the endpoints""";

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String applicationId = args[0];
    String userId = args[1];
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    listAllEndpoints(pinpoint, applicationId, userId);
    pinpoint.close();
}

public static void listAllEndpoints(PinpointClient pinpoint,
    String applicationId,
    String userId) {

    try {
        GetUserEndpointsRequest endpointsRequest =
GetUserEndpointsRequest.builder()
            .userId(userId)
            .applicationId(applicationId)
            .build();

        GetUserEndpointsResponse response =
pinpoint.getUserEndpoints(endpointsRequest);
        List<EndpointResponse> endpoints =
response.endpointsResponse().item();

        // Display the results.
        for (EndpointResponse endpoint : endpoints) {
```

```
        System.out.println("The channel type is: " +
endpoint.channelType());
        System.out.println("The address is " + endpoint.address());
    }

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [GetUserEndpoints](#) consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **SendMessage** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `SendMessage`.

.NET

AWS SDK for .NET

### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Invia un messaggio e-mail.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;
```

```
public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon
        Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
        message.
        // Make sure that the SMS channel is enabled for the project or
        application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
                ex.Message);
        }
    }
}
```

```

public static async Task<MessageResponse> SendEmailMessage(
    string region, string appId, string toAddress, string senderAddress)
{
    var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

    // The subject line of the email.
    string subject = "Amazon Pinpoint Email test";

    // The body of the email for recipients whose email clients don't
    // support HTML content.
    string textBody = @"Amazon Pinpoint Email Test (.NET)"
        + "\n-----"
        + "\nThis email was sent using the Amazon Pinpoint API
using the AWS SDK for .NET.";

    // The body of the email for recipients whose email clients support
    // HTML content.
    string htmlBody = @"<html>"
        + "\n<head></head>"
        + "\n<body>"
        + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK
for .NET)</h1>"
        + "\n  <p>This email was sent using the "
        + "\n    <a href='https://aws.amazon.com/
pinpoint/'>Amazon Pinpoint</a> API "
        + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
        + "\n  </p>"
        + "\n</body>"
        + "\n</html>";

    // The character encoding the you want to use for the subject line and
    // message body of the email.
    string charset = "UTF-8";

    var sendRequest = new SendMessagesRequest
    {
        ApplicationId = appId,
        MessageRequest = new MessageRequest
        {
            Addresses = new Dictionary<string, AddressConfiguration>
            {
                {

```

```
        toAddress,
        new AddressConfiguration
        {
            ChannelType = ChannelType.EMAIL
        }
    },
    MessageConfiguration = new DirectMessageConfiguration
    {
        EmailMessage = new EmailMessage
        {
            FromAddress = senderAddress,
            SimpleEmail = new SimpleEmail
            {
                HtmlPart = new SimpleEmailPart
                {
                    Charset = charset,
                    Data = htmlBody
                },
                TextPart = new SimpleEmailPart
                {
                    Charset = charset,
                    Data = textBody
                },
                Subject = new SimpleEmailPart
                {
                    Charset = charset,
                    Data = subject
                }
            }
        }
    }
};
Console.WriteLine("Sending message...");
SendMessageResponse response = await
client.SendMessagesAsync(sendRequest);
Console.WriteLine("Message sent!");
return response.MessageResponse;
}
}
```

## Invia un messaggio SMS.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone
        // number
        // or short code that you specify has to be associated with your Amazon
        // Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number. For best results, you should specify
        // the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this
        // message.
        // Make sure that the SMS channel is enabled for the project or
        // application
        // that you choose.
        string appId = configuration["AppId"]!;
```

```
// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
MessageType messageType = MessageType.TRANSACTIONAL;

// The registered keyword associated with the originating short code.
string? registeredKeyword = configuration["RegisteredKeyword"];

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
string? senderId = configuration["SenderId"];

try
{
    var response = await SendSmsMessage(region, appId, destinationNumber,
        originationNumber, registeredKeyword, senderId, messageType);
    Console.WriteLine($"Message sent to
{response.MessageResponse.Result.Count} recipient(s).");
    foreach (var messageResultValue in
        response.MessageResponse.Result.Select(r => r.Value))
    {
        Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
    }
}
catch (Exception ex)
{
    Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
}
}

public static async Task<SendMessagesResponse> SendSmsMessage(
    string region, string appId, string destinationNumber, string
originationNumber,
    string? keyword, string? senderId, MessageType messageType)
{

    // The content of the SMS message.
    string message = "This message was sent through Amazon Pinpoint using" +
        " the AWS SDK for .NET. Reply STOP to opt out.";
```

```
var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

SendMessagesRequest sendRequest = new SendMessagesRequest
{
    ApplicationId = appId,
    MessageRequest = new MessageRequest
    {
        Addresses =
            new Dictionary<string, AddressConfiguration>
            {
                {
                    destinationNumber,
                    new AddressConfiguration { ChannelType =
ChannelType.SMS }
                }
            },
        MessageConfiguration = new DirectMessageConfiguration
        {
            SMSMessage = new SMSMessage
            {
                Body = message,
                MessageType = MessageType.TRANSACTIONAL,
                OriginationNumber = originationNumber,
                SenderId = senderId,
                Keyword = keyword
            }
        }
    }
};
SendMessagesResponse response = await
client.SendMessagesAsync(sendRequest);
return response;
}
```

- Per i dettagli sull'API, [SendMessages](#) consulta AWS SDK for .NET API Reference.

## CLI

### AWS CLI

Invio di un messaggio SMS utilizzando l'endpoint di un'applicazione

L'esempio `send-messages` seguente invia un messaggio diretto per un'applicazione con un endpoint.

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-west-2
```

Contenuto di `myfile.json`.

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Endpoints": {  
    "testendpoint": {}  
  }  
}
```

Output:

```
{  
  "MessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "EndpointResult": {  
      "testendpoint": {  
        "Address": "+12345678900",  
        "DeliveryStatus": "SUCCESSFUL",  
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0",  
        "StatusCode": 200,  
        "StatusMessage": "MessageId:  
itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0"  
      }  
    },  
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Canale SMS di Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

- Per i dettagli sull'API, consulta [SendMessages AWS CLI Command Reference](#).

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Invia un messaggio e-mail.

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.pinpoint.PinpointClient;  
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;  
import software.amazon.awssdk.services.pinpoint.model.ChannelType;  
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;  
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;  
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;  
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;  
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;  
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;  
import software.amazon.awssdk.services.pinpoint.model.PinpointException;  
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;  
import software.amazon.awssdk.services.pinpointemail.model.Body;  
import software.amazon.awssdk.services.pinpointemail.model.Content;  
import software.amazon.awssdk.services.pinpointemail.model.Destination;  
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;  
import software.amazon.awssdk.services.pinpointemail.model.Message;  
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;  
  
import java.util.HashMap;  
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessage {

    // The character encoding the you want to use for the subject line and
    // message body of the email.
    public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
    content.
    static final String body = ""
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
    SDK for Java 2.x

    """;

    public static void main(String[] args) {
        final String usage = ""

            Usage:    <subject> <appId> <senderAddress>
<toAddress>

            Where:
                subject - The email subject to use.
                senderAddress - The from address. This address has to be verified
in Amazon Pinpoint in the region you're using to send email\s
                toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }
    }
}
```

```
String subject = args[0];
String senderAddress = args[1];
String toAddress = args[2];
System.out.println("Sending a message");
PinpointEmailClient pinpoint = PinpointEmailClient.builder()
    .region(Region.US_EAST_1)
    .build();

sendEmail(pinpoint, subject, senderAddress, toAddress);
System.out.println("Email was sent");
pinpoint.close();
}

public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
    try {
        Content content = Content.builder()
            .data(body)
            .build();

        Body messageBody = Body.builder()
            .text(content)
            .build();

        Message message = Message.builder()
            .body(messageBody)
            .subject(Content.builder().data(subject).build())
            .build();

        Destination destination = Destination.builder()
            .toAddresses(toAddress)
            .build();

        EmailContent emailContent = EmailContent.builder()
            .simple(message)
            .build();

        SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
            .fromEmailAddress(senderAddress)
            .destination(destination)
            .content(emailContent)
            .build();

        pinpointEmailClient.sendEmail(sendEmailRequest);
    }
}
```

```
        System.out.println("Message Sent");

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

## Invio di un messaggio e-mail con valori CC.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessageCC {

    // The body of the email.
    static final String body = """"
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
        SDK for Java 2.x

        """";

    public static void main(String[] args) {
```

```
final String usage = ""

Usage:    <subject> <senderAddress> <toAddress> <ccAddress>

Where:
  subject - The email subject to use.
  senderAddress - The from address. This address has to be verified
in Amazon Pinpoint in the region you're using to send email\s
  toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
  ccAddress - The CC address.
"";

if (args.length != 4) {
    System.out.println(usage);
    System.exit(1);
}

String subject = args[0];
String senderAddress = args[1];
String toAddress = args[2];
String ccAddress = args[3];

System.out.println("Sending a message");
PinpointEmailClient pinpoint = PinpointEmailClient.builder()
    .region(Region.US_EAST_1)
    .build();

ArrayList<String> ccList = new ArrayList<>();
ccList.add(ccAddress);
sendEmail(pinpoint, subject, senderAddress, toAddress, ccList);
pinpoint.close();
}

public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress, ArrayList<String> ccAddresses)
{
    try {
        Content content = Content.builder()
            .data(body)
            .build();

        Body messageBody = Body.builder()
            .text(content)
```

```
        .build();

        Message message = Message.builder()
            .body(messageBody)
            .subject(Content.builder().data(subject).build())
            .build();

        Destination destination = Destination.builder()
            .toAddresses(toAddress)
            .ccAddresses(ccAddresses)
            .build();

        EmailContent emailContent = EmailContent.builder()
            .simple(message)
            .build();

        SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
            .fromEmailAddress(senderAddress)
            .destination(destination)
            .content(emailContent)
            .build();

        pinpointEmailClient.sendEmail(sendEmailRequest);
        System.out.println("Message Sent");

    } catch (PinpointException e) {
        // Handle exception
        e.printStackTrace();
    }
}
}
```

### Invia un messaggio SMS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
```

```

import software.amazon.awssdk.services.pinpoint.model.SendMessageResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
    public static String senderId = "MySenderId";

    public static void main(String[] args) {
        final String usage = ""

            Usage:  <message> <appId> <originationNumber>
<destinationNumber>\s

            Where:
                message - The body of the message to send.
                appId - The Amazon Pinpoint project/application
ID to use when you send this message.
                originationNumber - The phone number or
short code that you specify has to be associated with your Amazon Pinpoint

```

account. For best results, specify long codes in E.164 format (for example, +1-555-555-5654).

`destinationNumber` - The recipient's phone number. For best results, you should specify the phone number in E.164 format (for example, +1-555-555-5654).\s

```

        """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String message = args[0];
        String appId = args[1];
        String originationNumber = args[2];
        String destinationNumber = args[3];
        System.out.println("Sending a message");
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendSMSMessage(pinpoint, message, appId, originationNumber,
            destinationNumber);
        pinpoint.close();
    }

    public static void sendSMSMessage(PinpointClient pinpoint, String
        message, String appId,
            String originationNumber,
            String destinationNumber) {
        try {
            Map<String, AddressConfiguration> addressMap = new
                HashMap<String, AddressConfiguration>();
            AddressConfiguration addConfig =
                AddressConfiguration.builder()
                    .channelType(ChannelType.SMS)
                    .build();

            addressMap.put(destinationNumber, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                .body(message)
                .messageType(messageType)
                .originationNumber(originationNumber)
                .senderId(senderId)

```

```

        .keyword(registeredKeyword)
        .build();

        // Create a DirectMessageConfiguration object.
        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
        .smsMessage(smsMessage)
        .build();

        MessageRequest msgReq = MessageRequest.builder()
        .addresses(addressMap)
        .messageConfiguration(direct)
        .build();

        // create a SendMessagesRequest object
        SendMessagesRequest request =
SendMessagesRequest.builder()
        .applicationId(appId)
        .messageRequest(msgReq)
        .build();

        SendMessagesResponse response =
pinpoint.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

        // Write out the result of sendMessage.
        map1.forEach((k, v) -> System.out.println((k + ":" +
v)));

    } catch (PinpointException e) {
        System.err.println(e.getAwsErrorDetails().getErrorMessage());
        System.exit(1);
    }
}
}
}

```

Invia messaggi SMS in batch.

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;

```

```
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessageBatch {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
    public static String senderId = "MySenderId";

    public static void main(String[] args) {
        final String usage = ""

            Usage:  <message> <appId> <originationNumber>
<destinationNumber> <destinationNumber1>\s

            Where:
                message - The body of the message to send.
```

```

        appId - The Amazon Pinpoint project/application
        ID to use when you send this message.
        originationNumber - The phone number or
        short code that you specify has to be associated with your Amazon Pinpoint
        account. For best results, specify long codes in E.164 format (for example,
        +1-555-555-5654).
        destinationNumber - The recipient's phone
        number. For best results, you should specify the phone number in E.164 format
        (for example, +1-555-555-5654).
        destinationNumber1 - The second recipient's
        phone number. For best results, you should specify the phone number in E.164
        format (for example, +1-555-555-5654).\s
        """;

    if (args.length != 5) {
        System.out.println(usage);
        System.exit(1);
    }

    String message = args[0];
    String appId = args[1];
    String originationNumber = args[2];
    String destinationNumber = args[3];
    String destinationNumber1 = args[4];
    System.out.println("Sending a message");
    PinpointClient pinpoint = PinpointClient.builder()
        .region(Region.US_EAST_1)
        .build();

    sendSMSMessage(pinpoint, message, appId, originationNumber,
        destinationNumber, destinationNumber1);
    pinpoint.close();
}

public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
    String originationNumber,
    String destinationNumber, String destinationNumber1) {
    try {
        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
        AddressConfiguration addConfig =
AddressConfiguration.builder()
            .channelType(ChannelType.SMS)

```

```
        .build());

        // Add an entry to the Map object for each number to whom
you want to send a
        // message.
        addressMap.put(destinationNumber, addConfig);
        addressMap.put(destinationNumber1, addConfig);
        SMSMessage smsMessage = SMSMessage.builder()
            .body(message)
            .messageType(messageType)
            .originationNumber(originationNumber)
            .senderId(senderId)
            .keyword(registeredKeyword)
            .build();

        // Create a DirectMessageConfiguration object.
        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
            .smsMessage(smsMessage)
            .build();

        MessageRequest msgReq = MessageRequest.builder()
            .addresses(addressMap)
            .messageConfiguration(direct)
            .build();

        // Create a SendMessagesRequest object.
        SendMessagesRequest request =
SendMessagesRequest.builder()
            .applicationId(appId)
            .messageRequest(msgReq)
            .build();

        SendMessagesResponse response =
pinpoint.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

        // Write out the result of sendMessage.
        map1.forEach((k, v) -> System.out.println((k + ":" +
v)));

    } catch (PinpointException e) {
        System.err.println(e.getAwsErrorDetails().getErrorMessage());
    }
}
```

```
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [SendMessages](#) consulta AWS SDK for Java 2.x API Reference.

## JavaScript

### SDK per JavaScript (v3)

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Creare il client in un modulo separato ed esportarlo.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

Invia un messaggio e-mail.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
var subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
```

```
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
    using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the subject line and message body of the email.
var charset = "UTF-8";

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
```

```
        Data: body_text,
      },
    },
  },
},
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    } else {
      console.log(recipientResult.MessageId);
    }
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Invia un messaggio SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "../libs/pinClient.js";
```

```
/* The phone number or short code to send the message from. The phone number
   or short code that you specify has to be associated with your Amazon Pinpoint
   account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number. For best results, you should specify the phone
   number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX

// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
   Make sure that the SMS channel is enabled for the project or application
   that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
   time-sensitive content, specify TRANSACTIONAL. If you plan to send
   marketing-related content, specify PROMOTIONAL.*/
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
   // varies by country or region. For more information, see
   https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
   countries.html.*/

var senderId = "MySenderId";

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
  },
}
```

```

    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinpointClient.send(new SendMessagesCommand(params));
    console.log(
      "Message sent! " +
      data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"],
    );
  } catch (err) {
    console.log(err);
  }
};
run();

```

- Per i dettagli sull'API, [SendMessages](#) consulta AWS SDK for JavaScript API Reference.

## SDK per JavaScript (v2)

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

## Invia un messaggio e-mail.

```

"use strict";

const AWS = require("aws-sdk");

```

```
// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using
    the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
```

```
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
```

```
pinpoint.sendMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"]["toAddress"]["MessageId"]
    );
  }
});
```

Invia un messaggio SMS.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number. For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
```

```
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
// countries.html
var senderId = "MySenderId";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
  },
  MessageConfiguration: {
    SMSMessage: {
      Body: message,
      Keyword: registeredKeyword,
      MessageType: messageType,
      OriginationNumber: originationNumber,
      SenderId: senderId,
```

```
    },
  },
};

//Try to send the message.
pinpoint.sendMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
      data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

- Per i dettagli sull'API, [SendMessages](#) consulta AWS SDK for JavaScript API Reference.

## Kotlin

### SDK per Kotlin

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
*/

val body: String = ""
```

## Amazon Pinpoint test (AWS SDK for Kotlin)

This email was sent through the Amazon Pinpoint Email API using the AWS SDK for Kotlin.

```
""".trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
Usage:
    <subject> <appId> <senderAddress> <toAddress>

Where:
    subject - The email subject to use.
    senderAddress - The from address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email
    toAddress - The to address. This address has to be verified in Amazon
Pinpoint in the region you're using to send email
    """

    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(subjectVal: String?, senderAddress: String, toAddressVal:
String) {
    var content = Content {
        data = body
    }

    val messageBody = Body {
        text = content
    }

    val subContent = Content {
        data = subjectVal
    }
}
```

```
val message = Message {
    body = messageBody
    subject = subContent
}

val destination0b = Destination {
    toAddresses = listOf(toAddressVal)
}

val emailContent = EmailContent {
    simple = message
}

val sendEmailRequest = SendEmailRequest {
    fromEmailAddress = senderAddress
    destination = destination0b
    this.content = emailContent
}

PinpointEmailClient { region = "us-east-1" }.use { pinpointemail ->
    pinpointemail.sendEmail(sendEmailRequest)
    println("Message Sent")
}
}
```

- Per i dettagli sull'API, [SendMessages](#) consulta AWS SDK for Kotlin API reference.

## Python

### SDK per Python (Boto3)

#### Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Invia un messaggio e-mail.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this
    message.
    :param sender: The "From" address. This address must be verified in
    Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
    account
        is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and
    message
        body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
    can
        display HTML content.
    :param text_message: The body of the email for recipients whose email clients
        don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
```

```

        to_address: {"ChannelType": "EMAIL"} for to_address in
to_addresses
    },
    "MessageConfiguration": {
        "EmailMessage": {
            "FromAddress": sender,
            "SimpleEmail": {
                "Subject": {"Charset": char_set, "Data": subject},
                "HtmlPart": {"Charset": char_set, "Data":
html_message},
                "TextPart": {"Charset": char_set, "Data":
text_message},
            },
        },
    },
)
except ClientError:
    logger.exception("Couldn't send email.")
    raise
else:
    return {
        to_address: message["MessageId"]
        for to_address, message in response["MessageResponse"]
["Result"].items()
    }

def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
-----
This email was sent with Amazon Pinpoint using the AWS SDK for Python
(Boto3).
For more information, see https://aws.amazon.com/sdk-for-python/
"""
    html_message = """<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for Python (Boto3))</h1>

```

```
<p>This email was sent with
  <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
  <a href='https://aws.amazon.com/sdk-for-python/'>
    AWS SDK for Python (Boto3)</a>.</p>
</body>
</html>

"""

print("Sending email.")
message_ids = send_email_message(
    boto3.client("pinpoint"),
    app_id,
    sender,
    [to_address],
    char_set,
    subject,
    html_message,
    text_message,
)
print(f"Message sent! Message IDs: {message_ids}")

if __name__ == "__main__":
    main()
```

Invia un messaggio SMS.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
```

```

):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you
    send
        this message. The SMS channel must be enabled for the project
    or
        application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This
    phone
        number must be associated with your Amazon
    Pinpoint
        account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you
    send
        time-sensitive content, specify TRANSACTIONAL. If you
    send
        marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]

```

```
def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK
for "
        "Python (Boto 3).")
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")

if __name__ == "__main__":
    main()
```

Invia un messaggio e-mail con un modello di e-mail esistente.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
    template_version
):
    """
```

Sends an email message with HTML and plain text versions.

:param pinpoint\_client: A Boto3 Pinpoint client.

:param project\_id: The Amazon Pinpoint project ID to use when you send this message.

:param sender: The "From" address. This address must be verified in Amazon Pinpoint in the AWS Region you're using to send email.

:param to\_addresses: The addresses on the "To" line. If your Amazon Pinpoint account is in the sandbox, these addresses must be verified.

:param template\_name: The name of the email template to use when sending the message.

:param template\_version: The version number of the message template.

:return: A dict of to\_addresses and their message IDs.

"""

try:

```

    response = pinpoint_client.send_messages(
        ApplicationId=project_id,
        MessageRequest={
            "Addresses": {
                to_address: {"ChannelType": "EMAIL"} for to_address in
to_addresses
            },
            "MessageConfiguration": {"EmailMessage": {"FromAddress":
sender}},
            "TemplateConfiguration": {
                "EmailTemplate": {
                    "Name": template_name,
                    "Version": template_version,
                }
            },
        },
    )
except ClientError:
    logger.exception("Couldn't send email.")
    raise
else:
    return {
        to_address: message["MessageId"]
        for to_address, message in response["MessageResponse"]
["Result"].items()
    }

```

```
def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")

if __name__ == "__main__":
    main()
```

Invia un messaggio di testo con un modello SMS esistente.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
```

Sends an SMS message to a specific phone number using a pre-defined template.

```

:param pinpoint_client: A Boto3 Pinpoint client.
:param project_id: An Amazon Pinpoint project (application) ID.
:param destination_number: The phone number to send the message to.
:param message_type: The type of SMS message (promotional or transactional).
:param origination_number: The phone number that the message is sent from.
:param template_name: The name of the SMS template to use when sending the
message.
:param template_version: The version number of the message template.

:return The ID of the message.
"""
try:
    response = pinpoint_client.send_messages(
        ApplicationId=project_id,
        MessageRequest={
            "Addresses": {destination_number: {"ChannelType": "SMS"}},
            "MessageConfiguration": {
                "SMSMessage": {
                    "MessageType": message_type,
                    "OriginationNumber": origination_number,
                }
            },
            "TemplateConfiguration": {
                "SMSTemplate": {"Name": template_name, "Version":
template_version}
            },
        },
    )

except ClientError:
    logger.exception("Couldn't send message.")
    raise
else:
    return response["MessageResponse"]["Result"][destination_number]
["MessageId"]

def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"

```

```
message_type = "TRANSACTIONAL"
template_name = "My_SMS_Template"
template_version = "1"
message_id = send_templated_sms_message(
    boto3.client("pinpoint", region_name=region),
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
)
print(f"Message sent! Message ID: {message_id}.")

if __name__ == "__main__":
    main()
```

- Per i dettagli sull'API, consulta [SendMessages AWSSDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **UpdateEndpoint** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare `UpdateEndpoint`.

Java

SDK per Java 2.x

### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
```

```
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Date;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class UpdateEndpoint {
    public static void main(String[] args) {
        final String usage = ""

            Usage: <appId>

            Where:
                appId - The ID of the application to create an endpoint for.

            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
    }
}
```

```
String appId = args[0];
PinpointClient pinpoint = PinpointClient.builder()
    .region(Region.US_EAST_1)
    .build();

EndpointResponse response = createEndpoint(pinpoint, appId);
System.out.println("Got Endpoint: " + response.id());
pinpoint.close();
}

public static EndpointResponse createEndpoint(PinpointClient client, String
appId) {
    String endpointId = UUID.randomUUID().toString();
    System.out.println("Endpoint ID: " + endpointId);

    try {
        EndpointRequest endpointRequest = createEndpointRequestData();
        UpdateEndpointRequest updateEndpointRequest =
UpdateEndpointRequest.builder()
            .applicationId(appId)
            .endpointId(endpointId)
            .endpointRequest(endpointRequest)
            .build();

        UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
        System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

        GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
            .applicationId(appId)
            .endpointId(endpointId)
            .build();

        GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
        System.out.println(getEndpointResponse.endpointResponse().address());

        System.out.println(getEndpointResponse.endpointResponse().channelType());

        System.out.println(getEndpointResponse.endpointResponse().applicationId());

        System.out.println(getEndpointResponse.endpointResponse().endpointStatus());
    }
}
```

```
System.out.println(getEndpointResponse.endpointResponse().requestId());
    System.out.println(getEndpointResponse.endpointResponse().user());

    return getEndpointResponse.endpointResponse();

} catch (PinpointException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return null;
}

private static EndpointRequest createEndpointRequestData() {
    try {
        List<String> favoriteTeams = new ArrayList<>();
        favoriteTeams.add("Lakers");
        favoriteTeams.add("Warriors");
        HashMap<String, List<String>> customAttributes = new HashMap<>();
        customAttributes.put("team", favoriteTeams);

        EndpointDemographic demographic = EndpointDemographic.builder()
            .appVersion("1.0")
            .make("apple")
            .model("iPhone")
            .modelVersion("7")
            .platform("ios")
            .platformVersion("10.1.1")
            .timezone("America/Los_Angeles")
            .build();

        EndpointLocation location = EndpointLocation.builder()
            .city("Los Angeles")
            .country("US")
            .latitude(34.0)
            .longitude(-118.2)
            .postalCode("90068")
            .region("CA")
            .build();

        Map<String, Double> metrics = new HashMap<>();
        metrics.put("health", 100.00);
        metrics.put("luck", 75.00);
    }
}
```

```

        EndpointUser user = EndpointUser.builder()
            .userId(UUID.randomUUID().toString())
            .build();

        DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
        Quoted "Z" to indicate UTC, no timezone                                     //
        offset

        String nowAsISO = df.format(new Date());

        return EndpointRequest.builder()
            .address(UUID.randomUUID().toString())
            .attributes(customAttributes)
            .channelType("APNS")
            .demographic(demographic)
            .effectiveDate(nowAsISO)
            .location(location)
            .metrics(metrics)
            .optOut("NONE")
            .requestId(UUID.randomUUID().toString())
            .user(user)
            .build();

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return null;
}
}

```

- Per i dettagli sull'API, [UpdateEndpoint](#) consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

# Esempi di codice per l'API Amazon Pinpoint SMS and Voice utilizzando SDK AWS

I seguenti esempi di codice mostrano come utilizzare l'API Amazon Pinpoint SMS and Voice con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. [Utilizzo di Amazon Pinpoint con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- [Azioni per l'API SMS e Voice di Amazon Pinpoint tramite SDK AWS](#)
- [Utilizzo SendVoiceMessage con un AWS SDK o una CLI](#)

## Azioni per l'API SMS e Voice di Amazon Pinpoint tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni API Amazon Pinpoint SMS e Voice con AWS gli SDK. Queste parti di codice chiamano l'API SMS e Voce di Amazon Pinpoint e provengono da programmi più grandi che devono essere eseguiti nel contesto. Ogni esempio include un link a GitHub, dove puoi trovare le istruzioni per configurare ed eseguire il codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la [documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint](#).

Esempi

- [Utilizzo SendVoiceMessage con un AWS SDK o una CLI](#)

## Utilizzo **SendVoiceMessage** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare `SendVoiceMessage`.

## Java

### SDK per Java 2.x

#### Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
    list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/
    voicelist.html
    static final String voiceName = "Matthew";
```

```

// The language to use when sending the message. For a list of supported
// languages, see
// https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
static final String languageCode = "en-US";

// The content of the message. This example uses SSML to customize and
control
// certain aspects of the message, such as by adding pauses and changing
// phonation. The message can't contain any line breaks.
static final String ssmlMessage = "<speak>This is a test message sent
from "
    + "<emphasis>Amazon Pinpoint</emphasis> "
    + "using the <break strength='weak'>AWS "
    + "SDK for Java. "
    + "<amazon:effect phonation='soft'>Thank "
    + "you for listening.</amazon:effect></speak>";

public static void main(String[] args) {

    final String usage = ""

        Usage:  <originationNumber> <destinationNumber>

\s

        Where:
            originationNumber - The phone number or
short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format (for example,
+1-555-555-5654).

            destinationNumber - The recipient's phone
number. For best results, you should specify the phone number in E.164 format
(for example, +1-555-555-5654).\s
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String originationNumber = args[0];
    String destinationNumber = args[1];
    System.out.println("Sending a voice message");

    // Set the content type to application/json.

```

```
List<String> listVal = new ArrayList<>();
listVal.add("application/json");
Map<String, List<String>> values = new HashMap<>();
values.put("Content-Type", listVal);

ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
    .headers(values)
    .build();

PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
    .overrideConfiguration(config2)
    .region(Region.US_EAST_1)
    .build();

sendVoiceMsg(client, originationNumber, destinationNumber);
client.close();
}

public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
    String destinationNumber) {
    try {
        SSMLMessageType ssmlMessageType =
SSMLMessageType.builder()
            .languageCode(languageCode)
            .text(ssmlMessage)
            .voiceId(voiceName)
            .build();

        VoiceMessageContent content =
VoiceMessageContent.builder()
            .ssmlMessage(ssmlMessageType)
            .build();

        SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()

            .destinationPhoneNumber(destinationNumber)

            .originationPhoneNumber(originationNumber)
                .content(content)
                .build();
```

```
        client.sendVoiceMessage(voiceMessageRequest);
        System.out.println("The message was sent successfully.");

    } catch (PinpointSmsVoiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Per i dettagli sull'API, [SendVoiceMessage](#) consulta AWS SDK for Java 2.x API Reference.

## JavaScript

### SDK per JavaScript (v2)

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
// results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";
```

```
// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/
SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
    "<speak>" +
    "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
    "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
    "<amazon:effect phonation='soft'>Thank you for listening." +
    "</amazon:effect>" +
    "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
// account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointSMSVoice = new AWS.PinpointSMSVoice();

var params = {
    CallerId: callerId,
    ConfigurationSetName: configurationSet,
    Content: {
```

```
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointSMSVoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

- Per i dettagli sull'API, [SendVoiceMessage](#) consulta AWS SDK for JavaScript API Reference.

## Python

### SDK per Python (Boto3)

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
```

```

def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
        The phone number must be associated with your
    Amazon
        Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
        device. The phone number must be associated with your
    Amazon
        Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
        number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
    message.
    :param ssml_message: The content of the message. This example uses SSML to
    control
        certain aspects of the message, such as the volume and
    the
        speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            }
        )

```

```
        }
    },
)
except ClientError:
    logger.exception(
        "Couldn't send message from %s to %s.",
        origination_number,
        destination_number,
    )
    raise
else:
    return response["MessageId"]

def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to "
          f"{destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")

if __name__ == "__main__":
    main()
```

- Per i dettagli sull'API, consulta [SendVoiceMessage AWSSDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Pinpoint con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

# Sicurezza in Amazon Pinpoint

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili ad Amazon Pinpoint, consulta [AWS Services in Scope by Compliance Program](#) Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

La presente documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon Pinpoint. Gli argomenti seguenti descrivono come configurare Amazon Pinpoint per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Pinpoint.

Per ulteriori informazioni sulle architetture di riferimento, consulta il manuale relativo all'[architettura resiliente di Amazon Pinpoint](#).

## Argomenti

- [Protezione dei dati in Amazon Pinpoint](#)
- [Gestione di identità e accessi per Amazon Pinpoint](#)
- [Registrazione di log e monitoraggio in Amazon Pinpoint](#)
- [Convalida della conformità per Amazon Pinpoint](#)
- [Resilienza in Amazon Pinpoint](#)
- [Sicurezza dell'infrastruttura in Amazon Pinpoint](#)
- [Analisi della configurazione e delle vulnerabilità in Amazon Pinpoint](#)

- [Best practice di sicurezza per Amazon Pinpoint](#)

## Protezione dei dati in Amazon Pinpoint

Il [modello di responsabilità AWS condivisa](#) si applica alla protezione dei dati in Amazon Pinpoint. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon Pinpoint o altri utenti Servizi AWS utilizzando la console, l'API o AWS gli AWS CLI SDK. I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

A seconda di come configuri e utilizzi il servizio, Amazon Pinpoint può archiviare i seguenti tipi di dati personali tuoi o dei tuoi clienti.

### Dati di configurazione

Sono inclusi i dati di configurazione del progetto, ad esempio le credenziali e le impostazioni che definiscono come e quando Amazon Pinpoint invia i messaggi attraverso i canali supportati e i segmenti utente a cui vengono inviati i messaggi. Per inviare i messaggi, questi dati possono includere indirizzi IP dedicati per i messaggi e-mail, codici brevi e ID mittente per i messaggi SMS e credenziali per la comunicazione con i servizi di notifica push quali il servizio di notifiche push di Apple (APN) e il servizio Firebase Cloud Messaging (FCM).

### Dati relativi a utenti ed endpoint

Sono inclusi gli attributi standard e personalizzati utilizzati per archiviare e gestire i dati relativi agli utenti e agli endpoint per un progetto Amazon Pinpoint. Un attributo può memorizzare informazioni su un utente specifico (ad esempio il nome di un utente) o su un endpoint specifico per un utente (ad esempio l'indirizzo e-mail di un utente, il numero di cellulare o il token del dispositivo mobile). Questi dati possono anche includere gli ID utente esterni che correlano gli utenti di un progetto Amazon Pinpoint con gli utenti di un sistema esterno, ad esempio un sistema di gestione delle relazioni con i clienti. Per ulteriori informazioni su ciò che questi dati possono includere, consulta gli schemi [Utente](#) ed [Endpoint](#) nella documentazione di riferimento dell'API Amazon Pinpoint.

### Dati analitici

Sono inclusi i dati per le metriche, chiamati anche indicatori chiave delle prestazioni (KPI), che forniscono informazioni dettagliate sulle prestazioni di un progetto Amazon Pinpoint per aree quali il coinvolgimento degli utenti e l'attività di acquisto. Sono inclusi anche i dati per i parametri che forniscono informazioni dettagliate sui dati demografici degli utenti per un progetto. I dati possono derivare da attributi standard e personalizzati per utenti ed endpoint, ad esempio la città in cui risiede un utente. Possono derivare anche da eventi, ad esempio eventi di apertura e clic per i messaggi e-mail inviati per un progetto.

### Dati importati

Sono inclusi i dati di qualsiasi utente, segmentazione e analisi aggiunti o importati da origini esterne e utilizzati in Amazon Pinpoint. Un esempio è un file JSON importato in Amazon Pinpoint (direttamente tramite la console o da un bucket Amazon S3) per creare un segmento statico. Altri esempi sono i dati degli endpoint aggiunti a livello di programmazione per creare un segmento dinamico, gli indirizzi degli endpoint a cui si inviano i messaggi diretti e gli eventi per cui configuri un'app per segnalarli ad Amazon Pinpoint.

## Argomenti

- [Crittografia dei dati](#)
- [Riservatezza del traffico Internet](#)
- [Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint](#)

## Crittografia dei dati

I dati Amazon Pinpoint in transito e quelli inattivi sono crittografati. I dati inviati ad Amazon Pinpoint vengono crittografati mentre vengono ricevuti e archiviati. I dati recuperati da Amazon Pinpoint vengono trasmessi mediante i protocolli di sicurezza correnti.

### Crittografia a riposo

Amazon Pinpoint crittografa tutti i dati archiviati. Sono inclusi i dati di configurazione, i dati dell'utente e degli endpoint, i dati analitici e tutti i dati aggiunti o importati in Amazon Pinpoint. Per crittografare i tuoi dati, Amazon Pinpoint utilizza chiavi AWS Key Management Service interne AWS KMS() che il servizio possiede e gestisce per tuo conto. Queste chiavi vengono ruotate su base regolare. Per informazioni in merito AWS KMS, consulta la [AWS Key Management Service Developer Guide](#).

### Crittografia in transito

Amazon Pinpoint utilizza HTTPS e Transport Layer Security (TLS) 1.2 o versioni successive per comunicare con i client e le applicazioni. Per comunicare con altri AWS servizi, Amazon Pinpoint utilizza HTTPS e TLS 1.2. Inoltre, quando crei e gestisci risorse Amazon Pinpoint utilizzando la console, un AWS SDK o il AWS Command Line Interface, tutte le comunicazioni sono protette tramite HTTPS e TLS 1.2.

### Gestione delle chiavi

Per crittografare i dati di Amazon Pinpoint, Amazon Pinpoint utilizza chiavi AWS KMS interne che il servizio possiede e gestisce per tuo conto. Queste chiavi vengono ruotate su base regolare. Non puoi fornire e utilizzare chiavi tue AWS KMS o di altre chiavi per crittografare i dati archiviati in Amazon Pinpoint.

## Riservatezza del traffico Internet

La privacy del traffico interrete si riferisce alla protezione delle connessioni e del traffico tra Amazon Pinpoint e i client e le applicazioni locali e tra Amazon Pinpoint e AWS altre risorse nella stessa

regione. AWS Le seguenti funzionalità e procedure ti consentono di garantire la riservatezza del traffico Internet per Amazon Pinpoint.

## Traffico tra Amazon Pinpoint e applicazioni e client on-premise

Per stabilire una connessione privata tra Amazon Pinpoint e i client e le applicazioni nella rete on-premise, puoi utilizzare AWS Direct Connect. Consente di collegare la rete a una posizione AWS Direct Connect utilizzando un cavo Ethernet standard in fibra ottica. Un'estremità del cavo è collegata al router. L'altra estremità è collegata a un router. AWS Direct Connect Per ulteriori informazioni, consulta [Che cos'è AWS Direct Connect?](#) nella Guida per l'utente di AWS Direct Connect .

Per garantire l'accesso sicuro ad Amazon Pinpoint tramite API pubblicate, è consigliabile rispettare i requisiti di Amazon Pinpoint per le chiamate API. Amazon Pinpoint richiede ai client l'utilizzo di Transport Layer Security (TLS) 1.2 o versione successiva. I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID di chiave di accesso e una chiave di accesso segreta associata a un principale AWS Identity and Access Management (IAM) per l' AWS account. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

## Traffico tra Amazon Pinpoint e altre risorse AWS

Per proteggere le comunicazioni tra Amazon Pinpoint e altre AWS risorse nella stessa AWS regione, Amazon Pinpoint utilizza HTTPS e TLS 1.2 per impostazione predefinita.

## Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint

È possibile stabilire una connessione privata tra il cloud privato virtuale (VPC) e un endpoint in Amazon Pinpoint creando un endpoint VPC di interfaccia.

Gli endpoint di interfaccia sono alimentati da [AWS PrivateLink](#), una tecnologia che consente di accedere in modo privato alle API di Amazon Pinpoint senza un gateway Internet, un dispositivo NAT, una connessione VPN o. AWS Direct Connect Le istanze presenti nel VPC non richiedono indirizzi IP pubblici per comunicare con le API Amazon Pinpoint integrate con AWS PrivateLink.

Per ulteriori informazioni, consulta la [Guida per AWS PrivateLink](#).

## Creazione di un endpoint VPC di interfaccia

Puoi creare un endpoint di interfaccia utilizzando la console Amazon VPC o AWS Command Line Interface (CLI). Per ulteriori informazioni, consulta [Creare un endpoint di interfaccia nella Guida AWS PrivateLink](#).

Amazon Pinpoint supporta i seguenti nomi di servizio:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Se attivi il DNS privato per un endpoint di interfaccia, puoi effettuare richieste API ad Amazon Pinpoint utilizzando il nome DNS predefinito per Regione AWS, ad esempio, `com.amazonaws.us-east-1.pinpoint`. Per ulteriori informazioni, consulta [Hostname DNS](#) nella Guida per l'utente di AWS PrivateLink.

Per l'elenco di tutte le regioni e di tutti gli endpoint nei quali Amazon Pinpoint è disponibile, consulta la sezione relativa agli [endpoint del servizio AWS](#) nella Riferimenti generali di Amazon Web Services.

## Creazione di una policy degli endpoint VPC

È possibile collegare una policy degli endpoint all'endpoint VPC che controlla l'accesso. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Per ulteriori informazioni, consulta la sezione [Controllo dell'accesso ai servizi con policy di endpoint](#) nella Guida di AWS PrivateLink.

## Esempio di policy di endpoint VPC

La seguente policy degli endpoint VPC concede l'accesso alle azioni Amazon Pinpoint elencate per tutti i principali su tutte le risorse.

```
{
```

```
"Statement": [  
  {  
    "Principal": "*",  
    "Action": [  
      "mobiletargeting:CreateCampaign",  
      "mobiletargeting:CreateApp",  
      "mobiletargeting>DeleteApp",  
    ],  
    "Effect": "Allow",  
    "Resource": "*"   
  }  
]
```

## Gestione di identità e accessi per Amazon Pinpoint

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (chi può effettuare l'accesso) e autorizzato (chi dispone delle autorizzazioni) per utilizzare le risorse Amazon Pinpoint. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

### Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come Amazon Pinpoint funziona con IAM](#)
- [Azioni di Amazon Pinpoint per le policy IAM](#)
- [Esempi di policy basate su identità Amazon Pinpoint](#)
- [Ruoli IAM per attività comuni di Amazon Pinpoint](#)
- [Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint](#)

### Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon Pinpoint.

Utente del servizio: se utilizzi il servizio Amazon Pinpoint per eseguire il processo, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità Amazon Pinpoint utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon Pinpoint, consulta [Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint](#).

Amministratore del servizio: se sei il responsabile delle risorse Amazon Pinpoint presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon Pinpoint. Sarà tua responsabilità determinare le funzionalità e le risorse di Amazon Pinpoint a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon Pinpoint, consulta [Come Amazon Pinpoint funziona con IAM](#).

Amministratore IAM: se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad Amazon Pinpoint. Per visualizzare esempi di policy basate su identità di Amazon Pinpoint che possono essere utilizzate in IAM, consulta [Esempi di policy basate su identità Amazon Pinpoint](#).

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se

non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente di IAM.

## Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente di IAM.

## Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente di IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli

utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- **Sessioni di accesso inoltrato (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Cloud è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

## Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La

maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'AWS CLI o dall' AWS API.

## Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruoli IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente di IAM.

Amazon Pinpoint supporta l'utilizzo di policy basate su identità per controllare l'accesso alle risorse Amazon Pinpoint.

## Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a

una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Amazon Pinpoint supporta l'utilizzo di policy basate su risorse per controllare l'accesso alle risorse Amazon Pinpoint.

## Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Amazon Pinpoint non supporta l'uso delle liste di controllo degli accessi (ACL) per controllare l'accesso alle risorse.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations  
AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account

AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .

- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente di IAM.

Amazon Pinpoint supporta l'utilizzo di questi tipi di policy per controllare l'accesso alle risorse Amazon Pinpoint.

## Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

## Come Amazon Pinpoint funziona con IAM

Per utilizzare Amazon Pinpoint, gli utenti del tuo AWS account richiedono autorizzazioni che consentano loro di visualizzare dati di analisi, creare progetti, definire segmenti di utenti, distribuire campagne e altro ancora. Se integri un'app per dispositivi mobili o Web con Amazon Pinpoint, anche gli utenti dell'app richiedono l'accesso ad Amazon Pinpoint. Questo accesso consente all'app di registrare gli endpoint e di segnalare i dati di utilizzo ad Amazon Pinpoint. Per concedere l'accesso alle funzionalità di Amazon Pinpoint, crea policy AWS Identity and Access Management (IAM) che consentano azioni Amazon Pinpoint per identità IAM o risorse Amazon Pinpoint.

IAM è un servizio che aiuta gli amministratori a controllare in modo sicuro l'accesso alle risorse. AWS Le policy IAM includono istruzioni che consentono o negano azioni specifiche per risorse o utenti specifici. In Amazon Pinpoint è disponibile una [serie di azioni](#) che è possibile utilizzare nelle policy IAM per specificare autorizzazioni granulari per utenti e risorse Amazon Pinpoint. Questo significa che puoi concedere il livello appropriato di accesso ad Amazon Pinpoint senza creare policy troppo permissive che potrebbero esporre dati importanti o compromettere le tue risorse. Ad esempio, puoi

concedere l'accesso illimitato a un amministratore Amazon Pinpoint e in sola lettura a coloro che devono accedere solo a uno specifico progetto.

Prima di utilizzare IAM per gestire l'accesso ad Amazon Pinpoint, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con Amazon Pinpoint. Per avere una visione di alto livello di come Amazon Pinpoint e AWS altri servizi funzionano con IAM, [AWS consulta i servizi che funzionano con IAM](#) nella IAM User Guide.

## Argomenti

- [Policy basate su identità Amazon Pinpoint](#)
- [Policy di autorizzazione basate su risorse di Amazon Pinpoint](#)
- [Autorizzazione basata sui tag Amazon Pinpoint](#)
- [Ruoli IAM Amazon Pinpoint](#)

## Policy basate su identità Amazon Pinpoint

Con le policy basate su identità di IAM, è possibile specificare quali azioni e risorse sono consentite o rifiutate, nonché le condizioni in base alle quali le azioni sono consentite o rifiutate. Amazon Pinpoint supporta specifiche azioni, risorse e chiavi di condizione. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

## Azioni

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Ciò significa che le azioni delle policy controllano ciò che gli utenti possono eseguire sulla console Amazon Pinpoint. Controllano anche ciò che gli utenti possono fare a livello di codice utilizzando direttamente gli AWS SDK, AWS Command Line Interface (AWS CLI) o le API Amazon Pinpoint.

Le azioni di policy in Amazon Pinpoint utilizzano i seguenti prefissi:

- **mobiletargeting**: per le azioni che derivano dall'API Amazon Pinpoint, che è l'API primaria per Amazon Pinpoint.
- **sms-voice**: per le azioni che derivano dalla API SMS e Voce di Amazon Pinpoint, che è un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint.

Ad esempio, per concedere l'autorizzazione per visualizzare le informazioni su tutti i segmenti di un progetto, ovvero un'operazione che corrisponde all'operazione `GetSegments` nell'API Amazon Pinpoint, includi l'operazione `mobiletargeting:GetSegments` nella policy. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon Pinpoint definisce un proprio set di azioni che descrivono le attività che puoi eseguire.

Per specificare più operazioni in una sola istruzione, separarle con la virgola:

```
"Action": [  
    "mobiletargeting:action1",  
    "mobiletargeting:action2"
```

Puoi anche specificare più operazioni utilizzando i caratteri jolly (\*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Get`, includi la seguente azione:

```
"Action": "mobiletargeting:Get*"
```

Tuttavia, è consigliabile definire policy in grado di seguire il principio del privilegio minimo. In altre parole, è necessario creare policy che includano solo le autorizzazioni necessarie per eseguire un'operazione specifica.

Per un elenco delle azioni Amazon Pinpoint che puoi utilizzare nelle policy IAM, consulta [Azioni di Amazon Pinpoint per le policy IAM](#).

## Risorse

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best

practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Ad esempio, l'azione `mobiletargeting:GetSegments` recupera le informazioni su tutti i segmenti associati a un progetto Amazon Pinpoint specifico. Identifica un progetto con un ARN nel seguente formato:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Per ulteriori informazioni sul formato degli ARN, consulta [Amazon Resource Names \(ARN\)](#) nella Riferimenti generali di AWS.

Nelle policy IAM, puoi specificare gli ARN per i seguenti tipi di risorse Amazon Pinpoint:

- Campagne
- Percorsi
- Modelli di messaggio (denominati modelli in alcuni contesti)
- Progetti (denominati app o applicazioni in alcuni contesti)
- Modelli di raccomandazioni (indicati come raccomandatori in alcuni contesti)
- Segmenti

Ad esempio, per creare un'istruzione di policy per il progetto con ID progetto `810c7aab86d42fb2b56c8c966example`, utilizza il seguente ARN:

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Per specificare tutti i progetti che appartengono ad un account specifico, utilizza il carattere jolly (\*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Alcune azioni Amazon Pinpoint, ad esempio la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (\*).

```
"Resource": "*"
```

Nelle policy IAM, puoi anche specificare gli ARN per i seguenti tipi di risorse SMS e Voce di Amazon Pinpoint:

- Set di configurazione
- Elenco di esclusione
- Numero di telefono
- Pool
- ID mittente

Ad esempio, in una policy per creare un'istruzione per un numero di telefono associato l'ID phone-12345678901234567890123456789012, utilizza il seguente ARN:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/  
phone-12345678901234567890123456789012"
```

Per specificare tutti i numeri di telefono appartenenti a un account specifico, utilizza un carattere jolly (\*) al posto dell'ID del numero di telefono:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Alcune azioni di tipo SMS e Voce di Amazon Pinpoint non vengono eseguite su una risorsa specifica, ad esempio le azioni di gestione delle impostazioni a livello di account come i limiti di spesa. In questi casi, è necessario utilizzare il carattere jolly (\*).

```
"Resource": "*"
```

Molte azioni API Amazon Pinpoint interessano più risorse. Ad esempio, l'operazione `TagResource` può aggiungere un tag a più progetti. Per specificare più risorse in una singola istruzione, separa gli ARN con la virgola:

```
"Resource": [
```

```
"resource1",  
"resource2"
```

Per visualizzare l'elenco dei tipi di risorse Amazon Pinpoint e dei relativi ARN, consulta [Risorse definite da Amazon Pinpoint](#) nella Guida per l'utente di IAM. Per informazioni sulle azioni che è possibile specificare con l'ARN di ogni tipo di risorsa, consulta [Operazioni definite da Amazon Pinpoint](#) nella Guida per l'utente di IAM.

## Chiavi di condizione

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Amazon Pinpoint definisce il proprio set di chiavi di condizione, nonché supporta alcune chiavi di condizione globali. Per visualizzare un elenco di tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali](#) nella Guida per l'utente IAM. Per visualizzare l'elenco delle chiavi condizione di Amazon Pinpoint, consulta [Chiavi di condizione per Amazon Pinpoint](#) nella Guida dell'utente di IAM. Per informazioni su azioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta [Operazioni definite da Amazon Pinpoint](#) nella Guida dell'utente di IAM.

## Esempi

Per visualizzare esempi di policy basate su identità Amazon Pinpoint, consulta [Esempi di policy basate su identità Amazon Pinpoint](#).

## Policy di autorizzazione basate su risorse di Amazon Pinpoint

Le policy di autorizzazione basate su risorse sono documenti di policy JSON che specificano le azioni che possono essere eseguite da un'entità principale specificata su una risorsa Amazon Pinpoint e in base a quali condizioni. Amazon Pinpoint supporta policy di autorizzazione basate su risorse per campagne, percorsi, modelli di messaggi (modelli), modelli di raccomandazione, progetti (app) e segmenti.

## Esempi

Per visualizzare esempi di policy basate su risorse Amazon Pinpoint, consulta [the section called "Esempi di policy basate su identità"](#).

## Autorizzazione basata sui tag Amazon Pinpoint

Puoi associare i tag a determinati tipi di risorse Amazon Pinpoint o passare i tag in una richiesta ad Amazon Pinpoint. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}` o `aws:TagKeys`.

Per informazioni sull'assegnazione di tag alle risorse Amazon Pinpoint, inclusa una policy IAM di esempio, consulta [Assegnazione di tag alle risorse Amazon Pinpoint](#).

## Ruoli IAM Amazon Pinpoint

Un [ruolo IAM](#) è un'entità all'interno dell'account AWS che dispone di autorizzazioni specifiche.

## Utilizzo di credenziali temporanee con Amazon Pinpoint

Puoi utilizzare le credenziali temporanee per effettuare l'accesso utilizzando la federazione, assumere un ruolo IAM o assumere un ruolo tra più account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni API AWS Security Token Service (AWS STS) come [AssumeRole](#) o [GetFederationToken](#).

Amazon Pinpoint supporta l'uso di credenziali temporanee.

## Ruoli collegati ai servizi

[I ruoli collegati ai](#) AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

Amazon Pinpoint non utilizza ruoli collegati ai servizi.

## Ruoli dei servizi

Questa caratteristica consente a un servizio di assumere un [ruolo di servizio](#) per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

Amazon Pinpoint supporta l'utilizzo dei ruoli di servizio.

## Azioni di Amazon Pinpoint per le policy IAM

Per gestire l'accesso alle risorse Amazon Pinpoint nel tuo AWS account, puoi aggiungere azioni Amazon Pinpoint AWS Identity and Access Management alle policy (IAM). Con le azioni aggiunte alle policy, puoi controllare ciò che possono fare gli utenti mediante la console Amazon Pinpoint. Puoi anche controllare cosa possono fare gli utenti a livello di codice utilizzando direttamente gli AWS SDK, AWS Command Line Interface (AWS CLI) o le API Amazon Pinpoint.

In una policy si specificano le singole azioni con lo spazio dei nomi Amazon Pinpoint appropriato seguito da due punti (:) e dal nome dell'operazione, ad esempio `GetSegments`. La maggior parte delle azioni corrisponde a una richiesta all'API Amazon Pinpoint utilizzando un URI e un metodo HTTP specifici. Ad esempio, se consenti l'operazione `mobiletargeting:GetSegments` nella policy di un utente, l'utente può recuperare le informazioni su tutti i segmenti per un progetto inviando una richiesta HTTP GET all'URI [/apps/projectId/segments](#). Questa policy consente inoltre all'utente di visualizzare tali informazioni sulla console e recuperarle utilizzando un SDK o il. AWS CLI

Ogni azione viene eseguita su una specifica risorsa Amazon Pinpoint, identificata in un'istruzione della policy mediante il relativo nome della risorsa Amazon (ARN). Ad esempio, l'operazione

`mobiletargeting:GetSegments` viene eseguita su un progetto specifico, identificato con l'ARN `arn:aws:mobiletargeting:region:accountId:apps/projectId`.

In questo argomento vengono descritte le azioni Amazon Pinpoint che puoi aggiungere alle policy IAM per l'account AWS . Per esempi che illustrano come utilizzare le azioni nelle policy per gestire l'accesso alle risorse Amazon Pinpoint, consulta [Esempi di policy basate su identità Amazon Pinpoint](#).

## Argomenti

- [Azioni dell'API Amazon Pinpoint](#)
- [Azioni dell'API SMS e Voce versione 1 di Amazon Pinpoint](#)

## Azioni dell'API Amazon Pinpoint

In questa sezione vengono descritte le azioni associate alle funzionalità disponibili nell'API Amazon Pinpoint, che rappresenta l'API primaria di Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la [documentazione di riferimento dell'API Amazon Pinpoint](#).

### Categorie:

- [Analisi e parametri](#)
- [Campagne](#)
- [Canali](#)
- [Endpoints](#)
- [Flussi di eventi](#)
- [Eventi](#)
- [Processi di esportazione](#)
- [Processi di importazione](#)
- [Percorsi](#)
- [Modelli dei messaggi](#)
- [Messaggi](#)
- [Codici OTP \(One-Time Password\)](#)
- [Convalida del numero di telefono](#)
- [Progetti](#)
- [Modelli della funzione di suggerimento](#)

- [Segmenti](#)
- [Tag](#)
- [Utenti](#)

## Analisi e parametri

Le seguenti autorizzazioni fanno riferimento alla visualizzazione dei dati di analisi nella console Amazon Pinpoint. Sono anche correlate al recupero (query) dei dati aggregati per i parametri standard, denominati anche indicatori di prestazioni chiave (KPI), che si applicano a progetti, campagne e viaggi.

### **mobiletargeting:GetReports**

Visualizza i dati di analisi nella console Amazon Pinpoint. Questa autorizzazione è necessaria anche per creare segmenti contenenti attributi personalizzati mediante la console Amazon Pinpoint. È inoltre necessario ottenere una stima delle dimensioni di un segmento nella console Amazon Pinpoint.

- URI: non applicabile
- Metodo: non applicabile
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:*`

### **mobiletargeting:GetApplicationDateRangeKpi**

Recupera (esegue query) i dati aggregati per un parametro dell'applicazione standard. Si tratta di un parametro che si applica a tutte le campagne o ai messaggi transazionali associati a un progetto.

- URI: [`/apps/projectId/kpis/daterange/kpi-name`](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/daterange/kpi-name`

### **mobiletargeting:GetCampaignDateRangeKpi**

Recupera (esegue query) i dati aggregati per un parametro di campagna standard. Si tratta di un parametro che si applica a una singola campagna.

- URI: [`/apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`](#)
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`

### **mobiletargeting:GetJourneyDateRangeKpi**

Recupera (esegue query) i dati aggregati per un parametro di coinvolgimento viaggi standard. Si tratta di una metrica di coinvolgimento che si applica a un singolo percorso, ad esempio il numero di messaggi aperti dai partecipanti per tutte le attività di un percorso.

- URI: [/apps/projectId/journeys/journeyId/kpis/daterange/kpi-name](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`

### **mobiletargeting:GetJourneyExecutionMetrics**

Recupera (mediante query) i dati aggregati per le metriche di esecuzione standard applicabili a un singolo percorso, ad esempio il numero di partecipanti coinvolti attivamente in tutte le attività di un percorso.

- URI: [/apps/projectId/journeys/journeyId/execution-metrics](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/execution-metrics`

### **mobiletargeting:GetJourneyExecutionActivityMetrics**

Recupera (mediante query) i dati aggregati per le metriche di esecuzione standard applicabili a una singola attività in un percorso, ad esempio il numero di partecipanti che hanno iniziato o completato un'attività.

- URI: [/apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`

## Campagne

Le autorizzazioni seguenti fanno riferimento alla gestione delle campagne nell'account Amazon Pinpoint.

## **mobiletargeting:CreateCampaign**

Creare una campagna per un progetto.

- URI: [/apps/\*projectId\*/campaigns](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns`

## **mobiletargeting>DeleteCampaign**

Eliminare una determinata campagna.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## **mobiletargeting:GetCampaign**

Recuperare informazioni su una determinata partizione.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## **mobiletargeting:GetCampaignActivities**

Recuperare informazioni sulle attività eseguite da una campagna.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*/activities](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## **mobiletargeting:GetCampaigns**

Recuperare informazioni su tutte le campagne per un progetto.

- URI: [/apps/\*projectId\*/campaigns](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

## **mobiletargeting:GetCampaignVersion**

Recuperare informazioni su una determinata versione di una campagna.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*/versions/\*versionId\*](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## **mobiletargeting:GetCampaignVersions**

Recuperare informazioni sulla versione corrente e sulle versioni precedenti di una campagna.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*/versions](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## **mobiletargeting:UpdateCampaign**

Aggiornare una determinata campagna.

- URI: [/apps/\*projectId\*/campaigns/\*campaignId\*](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

## Canali

Le autorizzazioni seguenti fanno riferimento alla gestione dei canali nell'account Amazon Pinpoint. In Amazon Pinpoint, per canali si intendono i metodi utilizzati per contattare i clienti, come l'invio di messaggi e-mail, SMS o notifiche push.

## **mobiletargeting>DeleteAdmChannel**

Disabilitare il canale Amazon Device Messaging (ADM) per un progetto.

- URI: [/apps/\*projectId\*/channels/adm](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

## **mobiletargeting:GetAdmChannel**

Recuperare informazioni sul canale ADM per un progetto.

- URI: [/apps/\*projectId\*/channels/adm](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

## **mobiletargeting:UpdateAdmChannel**

Abilitare o aggiornare il canale ADM per un progetto.

- URI: [/apps/\*projectId\*/channels/adm](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

## **mobiletargeting>DeleteApnsChannel**

Disabilitare il canale del servizio di notifiche push Apple (APNs) per un progetto.

- URI: [/apps/\*projectId\*/channels/apns](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

## **mobiletargeting:GetApnsChannel**

Recuperare informazioni sul canale APNs per un progetto.

- URI: [/apps/\*projectId\*/channels/apns](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

## **mobiletargeting:UpdateApnsChannel**

Attiva o aggiorna il canale APN per un progetto.

- URI: [/apps/\*projectId\*/channels/apns](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

## **mobiletargeting:DeleteApnsSandboxChannel**

Disabilitare il canale sandbox APNs per un progetto.

- URI: [/apps/\*projectId\*/channels/apns\\_sandbox](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

## **mobiletargeting:GetApnsSandboxChannel**

Recuperare informazioni sul canale sandbox APNs per un progetto.

- URI: [/apps/\*projectId\*/channels/apns\\_sandbox](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

## **mobiletargeting:UpdateApnsSandboxChannel**

Abilita o aggiorna il canale dell'ambiente di sperimentazione (sandbox) per il servizio APN per un progetto.

- URI: [/apps/\*projectId\*/channels/apns\\_sandbox](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

## **mobiletargeting:DeleteApnsVoipChannel**

Disabilitare il canale VoIP APNs per un progetto.

- URI: [/apps/\*projectId\*/channels/apns\\_voip](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

## **mobiletargeting:GetApnsVoipChannel**

Recuperare informazioni sul canale VoIP APNs per un progetto.

- URI: [/apps/\*projectId\*/channels/apns\\_voip](#)
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

### **mobiletargeting:UpdateApnsVoipChannel**

Abilita o aggiorna il canale VoIP del servizio APN per un progetto.

- URI: [/apps/projectId/channels/apns\\_voip](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

### **mobiletargeting>DeleteApnsVoipSandboxChannel**

Disabilitare il canale sandbox VoIP APNs per un progetto.

- URI: [/apps/projectId/channels/apns\\_voip\\_sandbox](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

### **mobiletargeting:GetApnsVoipSandboxChannel**

Recuperare informazioni sul canale sandbox VoIP APNs per un progetto.

- URI: [/apps/projectId/channels/apns\\_voip\\_sandbox](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

### **mobiletargeting:UpdateApnsVoipSandboxChannel**

Attiva o aggiorna il canale dell'ambiente di sperimentazione (sandbox) per VoIP del servizio APN per un progetto.

- URI: [/apps/projectId/channels/apns\\_voip\\_sandbox](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

### **mobiletargeting>DeleteBaiduChannel**

Disabilitare il canale Baidu Cloud Push per un progetto.

- URI: [/apps/projectId/channels/baidu](#)

- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/baidu`

### **mobiletargeting:GetBaiduChannel**

Recuperare informazioni sul canale Baidu Cloud Push per un progetto.

- URI: [/apps/projectId/channels/baidu](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/baidu`

### **mobiletargeting:UpdateBaiduChannel**

Abilitare o aggiornare il canale Baidu Cloud Push per un progetto.

- URI: [/apps/projectId/channels/baidu](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/baidu`

### **mobiletargeting>DeleteEmailChannel**

Disabilitare il canale e-mail per un progetto.

- URI: [/apps/projectId/channels/email](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

### **mobiletargeting:GetEmailChannel**

Recuperare informazioni sul canale e-mail per un progetto.

- URI: [/apps/projectId/channels/email](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

### **mobiletargeting:UpdateEmailChannel**

Abilitare o aggiornare il canale e-mail per un progetto.

- URI: [/apps/projectId/channels/email](#)

- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

### **mobiletargeting:DeleteGcmChannel**

Disabilita il canale Firebase Cloud Messaging (FCM) per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: [/apps/projectId/channels/gcm](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

### **mobiletargeting:GetGcmChannel**

Recupera le informazioni sul canale FCM per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: [/apps/projectId/channels/gcm](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

### **mobiletargeting:UpdateGcmChannel**

Attiva o aggiorna il canale FCM per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: [/apps/projectId/channels/gcm](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

### **mobiletargeting>DeleteSmsChannel**

Disabilitare il canale SMS per un progetto.

- URI: [/apps/projectId/channels/sms](#)
- Metodo: DELETE

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

### **mobiletargeting:GetSmsChannel**

Recuperare informazioni sul canale SMS per un progetto.

- URI: [/apps/projectId/channels/sms](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

### **mobiletargeting:UpdateSmsChannel**

Abilitare o aggiornare il canale SMS per un progetto.

- URI: [/apps/projectId/channels/sms](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

### **mobiletargeting:GetChannels**

Recupera le informazioni sulla cronologia e lo stato di ciascun canale per un'applicazione.

- URI: [/apps/application-id/channels](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels`

### **mobiletargeting>DeleteVoiceChannel**

Disattiva il canale vocale per un'applicazione ed elimina tutte le impostazioni esistenti per il canale.

- URI: [/apps/application-id/channels/voice](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

### **mobiletargeting:GetVoiceChannel**

Recupera le informazioni sullo stato e sulle impostazioni del canale vocale per un'applicazione.

- URI: [/apps/application-id/channels/voice](#)

- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

### **mobiletargeting:UpdateVoiceChannel**

Abilita il canale vocale per un'applicazione o aggiorna lo stato e le impostazioni del canale vocale per un'applicazione.

- URI: [/apps/application-id/channels/voice](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

### Endpoints

Le autorizzazioni seguenti fanno riferimento alla gestione degli endpoint nell'account Amazon Pinpoint. In Amazon Pinpoint, per endpoint si intende una singola destinazione per i messaggi. Ad esempio, un endpoint può essere l'indirizzo e-mail di un cliente, un numero di telefono o un token di dispositivo mobile.

### **mobiletargeting>DeleteEndpoint**

Eliminare un endpoint.

- URI: [/apps/projectId/endpoints/endpointId](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

### **mobiletargeting:GetEndpoint**

Recuperare informazioni su un determinato endpoint.

- URI: [/apps/projectId/endpoints/endpointId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

### **mobiletargeting:RemoveAttributes**

Rimuove uno o più attributi dello stesso tipo da tutti gli endpoint associati a un'applicazione.

- URI: [apps/application-id/attributes/attribute-type](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/attributes/attribute-type`

### **mobiletargeting:UpdateEndpoint**

Creare un endpoint o aggiornare le informazioni per un endpoint.

- URI: [/apps/projectId/endpoints/endpointId](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

### **mobiletargeting:UpdateEndpointsBatch**

Creare o aggiornare endpoint con un'operazione batch.

- URI: [/apps/projectId/endpoints](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

## Flussi di eventi

Le autorizzazioni seguenti fanno riferimento alla gestione dei flussi di eventi nell'account Amazon Pinpoint.

### **mobiletargeting>DeleteEventStream**

Eliminare il flusso di eventi per un progetto.

- URI: [/apps/projectId/eventstream/](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

### **mobiletargeting:GetEventStream**

Recuperare informazioni sul flusso di eventi per un progetto.

- URI: [/apps/projectId/eventstream/](#)

- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

### **mobiletargeting:PutEventStream**

Creare o aggiornare un flusso di eventi per un progetto.

- URI: [/apps/projectId/eventstream/](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

## Eventi

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi relativi agli eventi nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di importazione per creare segmenti in base alle definizioni di endpoint archiviate in un bucket Amazon S3.

### **mobiletargeting:PutEvents**

Crea un nuovo evento da registrare per gli endpoint oppure crea o aggiorna i dati degli endpoint a cui sono associati gli eventi esistenti.

- URI: [/apps/application-id/events](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

## Processi di esportazione

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi di esportazione nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di esportazione per inviare informazioni sugli endpoint a un bucket Amazon S3 per l'archiviazione o per l'analisi.

### **mobiletargeting>CreateExportJob**

Crea un processo di esportazione per esportare le definizioni di endpoint in Amazon S3.

- URI: [/apps/projectId/jobs/export](#)

- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

### **mobiletargeting:GetExportJob**

Recuperare informazioni su un determinato processo di esportazione per un progetto.

- URI: [/apps/projectId/jobs/export/jobId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

### **mobiletargeting:GetExportJobs**

Recuperare un elenco di tutti i processi di esportazione per un progetto.

- URI: [/apps/projectId/jobs/export](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

## Processi di importazione

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi di importazione nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di importazione per creare segmenti in base alle definizioni di endpoint archiviate in un bucket Amazon S3.

### **mobiletargeting:CreateImportJob**

Importa le definizioni di endpoint da Amazon S3 per creare un segmento.

- URI: [/apps/projectId/jobs/import](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### **mobiletargeting:GetImportJob**

Recuperare informazioni su un determinato processo di importazione per un progetto.

- URI: [/apps/projectId/jobs/import/jobId](#)
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

### **mobiletargeting:GetImportJobs**

Recuperare informazioni su tutti i processi di importazione per un progetto.

- URI: [/apps/projectId/jobs/import](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### Percorsi

Le autorizzazioni seguenti fanno riferimento alla gestione dei percorsi nell'account Amazon Pinpoint.

### **mobiletargeting:CreateJourney**

Creare un viaggio per un progetto.

- URI: [/apps/projectId/journeys](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

### **mobiletargeting:GetJourney**

Recuperare informazioni su un determinato viaggio.

- URI: [/apps/projectId/journeys/journeyId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

### **mobiletargeting:ListJourneys**

Recuperare informazioni sui viaggi per un progetto.

- URI: [/apps/projectId/journeys](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

## **mobiletargeting:UpdateJourney**

Aggiorna la configurazione e altre impostazioni per un viaggio specifico.

- URI: [/apps/\*projectId\*/journeys/\*journeyId\*](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

## **mobiletargeting:UpdateJourneyState**

Annullare un viaggio attivo.

- URI: [/apps/\*projectId\*/journeys/\*journeyId\*/state](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/state`

## **mobiletargeting>DeleteJourney**

Eliminare un viaggio specifico.

- URI: [/apps/\*projectId\*/journeys/\*journeyId\*](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

## Modelli dei messaggi

Le seguenti autorizzazioni fanno riferimento alla creazione e alla gestione dei modelli di messaggio per l'account Amazon Pinpoint. Un modello di messaggio è un insieme di contenuti e impostazioni che puoi definire, salvare e riutilizzare nei messaggi inviati per ognuno dei tuoi progetti Amazon Pinpoint.

## **mobiletargeting>ListTemplates**

Recupera le informazioni su tutti i modelli di messaggio associati all'account Amazon Pinpoint.

- URI: [/templates](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates`

## **mobiletargeting:ListTemplateVersions**

Recuperare informazioni su tutte le versioni di un modello di messaggio specifico.

- URI: [/templates/\*template-name\*/\*template-type\*/versions](#)
- Metodo – GET
- ARN risorsa: non applicabile

## **mobiletargeting:UpdateTemplateActiveVersion**

Designare una versione specifica di un modello di messaggio come versione attiva del modello.

- URI: [/templates/\*template-name\*/\*template-type\*/active-version](#)
- Metodo – GET
- ARN risorsa: non applicabile

## **mobiletargeting:GetEmailTemplate**

Recuperare informazioni su un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: [/templates/\*template-name\*/email](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

## **mobiletargeting:CreateEmailTemplate**

Creare un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: [/templates/\*template-name\*/email](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

## **mobiletargeting:UpdateEmailTemplate**

Aggiornare un modello di messaggio esistente per i messaggi inviati attraverso il canale di posta elettronica.

- URI: [/templates/\*template-name\*/email](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

### **mobiletargeting:DeleteEmailTemplate**

Eliminare un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: [/templates/\*template-name\*/email](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

### **mobiletargeting:GetPushTemplate**

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: [/templates/\*template-name\*/push](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

### **mobiletargeting>CreatePushTemplate**

Creare un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: [/templates/\*template-name\*/push](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

### **mobiletargeting:UpdatePushTemplate**

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite un canale di notifica push.

- URI: [/templates/\*template-name\*/push](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

### **mobiletargeting:DeletePushTemplate**

Eliminare un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: [/templates/\*template-name\*/push](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

### **mobiletargeting:GetSmsTemplate**

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite il canale SMS.

- URI: [/templates/\*template-name\*/sms](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

### **mobiletargeting:CreateSmsTemplate**

Creare un modello di messaggio per i messaggi inviati attraverso il canale SMS.

- URI: [/templates/\*template-name\*/sms](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

### **mobiletargeting:UpdateSmsTemplate**

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite il canale SMS.

- URI: [/templates/\*template-name\*/sms](#)

- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

### **mobiletargeting:DeleteSmsTemplate**

Eliminare un modello di messaggio per i messaggi inviati tramite il canale SMS.

- URI: [/templates/template-name/sms](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

### **mobiletargeting:GetVoiceTemplate**

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite il canale vocale.

- URI: [/templates/template-name/voice](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

### **mobiletargeting:CreateVoiceTemplate**

Creare un modello di messaggio per i messaggi inviati attraverso il canale vocale.

- URI: [/templates/template-name/voice](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

### **mobiletargeting:UpdateVoiceTemplate**

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite il canale vocale.

- URI: [/templates/template-name/voice](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

## **mobiletargeting:DeleteVoiceTemplate**

Eliminare un modello di messaggio per i messaggi inviati tramite il canale vocale.

- URI: [/templates/\*template-name\*/voice](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

## Messaggi

Le autorizzazioni seguenti fanno riferimento all'invio di messaggi e notifiche push dall'account Amazon Pinpoint. È possibile utilizzare le operazioni `SendMessage` e `SendUsersMessages` per inviare messaggi a specifici endpoint senza prima creare segmenti e campagne.

## **mobiletargeting:SendMessage**

Inviare un messaggio o una notifica push a determinati endpoint.

- URI: [/apps/\*projectId\*/messages](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

## **mobiletargeting:SendUsersMessages**

Inviare un messaggio o una notifica push a tutti gli endpoint associati a un determinato ID utente.

- URI: [/apps/\*projectId\*/users-messages](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

## Codici OTP (One-Time Password)

Le seguenti autorizzazioni fanno riferimento all'invio e alla convalida di codici OTP (One-Time Password) in Amazon Pinpoint.

## **mobiletargeting:SendOTPMessage**

Inviare un messaggio di testo contenente un codice OTP (One-Time Password).

- URI: [/apps/projectId/otp](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/otp`

### **mobiletargeting:VerifyOTPMessage**

Verifica la validità di una password un codice OTP (One-Time Password) generato mediante l'operazione SendOTPMessage.

- URI: [/apps/projectId/verify-otp](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/verify-otp`

### Convalida del numero di telefono

Le autorizzazioni seguenti fanno riferimento all'utilizzo del servizio di convalida dei numeri di telefono in Amazon Pinpoint.

### **mobiletargeting:PhoneNumberValidate**

Recuperare informazioni su un numero di telefono.

- URI: [/phone/number/validate](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:phone/number/validate`

### Progetti

Le autorizzazioni seguenti fanno riferimento alla gestione dei progetti nell'account Amazon Pinpoint. In origine, i progetti erano denominati applicazioni. Nell'ambito di queste operazioni, un'applicazione Amazon Pinpoint coincide con un progetto Amazon Pinpoint.

### **mobiletargeting>CreateApp**

Creare un progetto Amazon Pinpoint

- URI: [/apps](#)

- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps`

### **mobiletargeting:DeleteApp**

Elimina un progetto Amazon Pinpoint.

- URI: [/apps/projectId](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### **mobiletargeting:GetApp**

Recupera le informazioni su un progetto Amazon Pinpoint.

- URI: [/apps/projectId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### **mobiletargeting:GetApps**

Recupera le informazioni su tutti i progetti associati all'account Amazon Pinpoint.

- URI: [/apps](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps`

### **mobiletargeting:GetApplicationSettings**

Recupera le impostazioni predefinite per un progetto Amazon Pinpoint.

- URI: [/apps/projectId/settings](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### **mobiletargeting:UpdateApplicationSettings**

Aggiorna le impostazioni predefinite per un progetto Amazon Pinpoint.

- URI: [/apps/projectId/settings](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

## Modelli della funzione di suggerimento

Le seguenti autorizzazioni fanno riferimento alla gestione delle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati delle raccomandazioni derivanti dai modelli di raccomandazione. Un modello della funzione di suggerimento è un tipo di modello di machine learning che prevede e genera suggerimenti personalizzati individuando pattern nei dati.

### **mobiletargeting:CreateRecommenderConfiguration**

Crea una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: [/recommenders](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders`

### **mobiletargeting:GetRecommenderConfigurations**

Recupera le informazioni su tutte le configurazioni del modello di raccomandazione associate all'account Amazon Pinpoint.

- URI: [/recommenders](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders`

### **mobiletargeting:GetRecommenderConfiguration**

Recupera le informazioni su una configurazione di Amazon Pinpoint per un modello di raccomandazione.

- URI: [/recommenders/recommenderId](#)
- Metodo – GET
- ARN risorsa:  
`arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

### **mobiletargeting:UpdateRecommenderConfiguration**

Aggiorna una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: [/recommenders/recommenderId](#)
- Method – PUT
- ARN risorsa:  
`arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

## **mobiletargeting:DeleteRecommenderConfiguration**

Elimina una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: [/recommenders/recommenderId](#)
- Metodo: DELETE
- ARN risorsa:  
arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId

## Segmenti

Le autorizzazioni seguenti fanno riferimento alla gestione dei segmenti nell'account Amazon Pinpoint. In Amazon Pinpoint, per segmenti si intendono gruppi di destinatari per le campagne che condividono determinati attributi specificati in precedenza.

## **mobiletargeting:CreateSegment**

Creare un segmento. Per permettere a un utente di creare un segmento importando i dati degli endpoint dall'esterno di Amazon Pinpoint, è necessario consentire l'azione `mobiletargeting:CreateImportJob`.

- URI: [/apps/projectId/segments](#)
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:region:accountId:apps/projectId

## **mobiletargeting>DeleteSegment**

Eliminare un segmento.

- URI: [/apps/projectId/segments/segmentId](#)
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId

## **mobiletargeting:GetSegment**

Recuperare informazioni su un determinato segmento.

- URI: [/apps/projectId/segments/segmentId](#)
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

### **mobiletargeting:GetSegmentExportJobs**

Recuperare informazioni sui processi che esportano definizioni di endpoint per un segmento.

- URI: [/apps/projectId/segments/segmentId/jobs/export](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId/jobs/export`

### **mobiletargeting:GetSegments**

Recuperare informazioni sui segmenti per un progetto.

- URI: [/apps/projectId/segments](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

### **mobiletargeting:GetSegmentImportJobs**

Recupera le informazioni sui processi che creano segmenti mediante l'importazione delle definizioni di endpoint da Amazon S3.

- URI: [/apps/projectId/segments/segmentId/jobs/import](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

### **mobiletargeting:GetSegmentVersion**

Recuperare informazioni su una determinata versione di un segmento.

- URI: [/apps/projectId/segments/segmentId/versions/versionId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

### **mobiletargeting:GetSegmentVersions**

Recuperare informazioni sulla versione corrente e sulle versioni precedenti di un segmento.

- URI: [/apps/projectId/segments/segmentId/versions](#)

- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

### **mobiletargeting:UpdateSegment**

Aggiornare un determinato segmento.

- URI: [/apps/projectId/segments/segmentId](#)
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

### Tag

Le autorizzazioni seguenti fanno riferimento alla visualizzazione e alla gestione dei tag per le risorse Amazon Pinpoint.

### **mobiletargeting:ListTagsForResource**

Recuperare informazioni sui tag associati a un progetto, una campagna, un modello di messaggio o un segmento.

- URI: [/tags/resource-arn](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:*`

### **mobiletargeting:TagResource**

Aggiungere uno o più tag a un progetto, una campagna, un modello di messaggio o un segmento.

- URI: [/tags/resource-arn](#)
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:*`

### **mobiletargeting:UntagResource**

Rimuovere uno o più tag da un progetto, una campagna, un modello di messaggio o un segmento.

- URI: [/tags/resource-arn](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:*`

## Utenti

Le autorizzazioni seguenti sono correlate alla gestione degli utenti. In Amazon Pinpoint, per utenti si intendono le persone che ricevono messaggi. Un singolo utente può essere associato a più endpoint.

### **mobiletargeting:DeleteUserEndpoints**

Eliminare tutti gli endpoint associati a un ID utente.

- URI: [/apps/projectId/users/userId](#)
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

### **mobiletargeting:GetUserEndpoints**

Recuperare informazioni su tutti gli endpoint associati a un ID utente.

- URI: [/apps/projectId/users/userId](#)
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

## Azioni dell'API SMS e Voce versione 1 di Amazon Pinpoint

In questa sezione vengono identificate le funzionalità disponibili nell'API SMS e Voce di Amazon Pinpoint. Si tratta di un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la [documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint](#).

### **sms-voice:CreateConfigurationSet**

Creare un set di configurazione per l'invio di messaggi vocali.

- URI: /sms-voice/configuration-sets
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza \*.

### **sms-voice:DeleteConfigurationSet**

Eliminare un set di configurazione per l'invio di messaggi vocali.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*
- Metodo: DELETE
- ARN risorsa: non disponibile. Utilizza \*.

### **sms-voice:GetConfigurationSetEventDestinations**

Richiamare informazioni su un set di configurazione e sulle destinazioni degli eventi che contiene.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*/event-destinazioni-
- Metodo – GET
- ARN risorsa: non disponibile. Utilizza \*.

### **sms-voice:CreateConfigurationSetEventDestination**

Creare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*/event-destinazioni-
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza \*.

### **sms-voice:UpdateConfigurationSetEventDestination**

Aggiornare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*/event-destinazioni/  
*EventDestinationName*
- Method – PUT
- ARN risorsa: non disponibile. Utilizza \*.

## sms-voice:DeleteConfigurationSetEventDestination

Eliminare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*/event-destinazioni/*EventDestinationName*
- Metodo: DELETE
- ARN risorsa: non disponibile. Utilizza \*.

## sms-voice:SendVoiceMessage

Creare e inviare messaggi vocali.

- URI: /sms-voice/voice/message
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza \*.

## Esempi di policy basate su identità Amazon Pinpoint

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon Pinpoint. Inoltre, non possono eseguire attività utilizzando, o un'API. AWS Management Console AWS CLI AWS Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse di cui hanno bisogno. L'amministratore deve quindi collegare queste policy a utenti o gruppi che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare [Creazione di policy nella scheda JSON](#) nella Guida per l'utente di IAM.

### Argomenti

- [Best practice delle policy](#)
- [Utilizzo della console Amazon Pinpoint](#)
- [Esempio: accesso a un singolo progetto Amazon Pinpoint](#)
- [Esempio: visualizzazione di risorse Amazon Pinpoint in base ai tag](#)
- [Esempio: consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Esempi: autorizzazione dell'accesso alle azioni dell'API Amazon Pinpoint](#)

- [Esempi: autorizzazione dell'accesso alle azioni dell'API SMS e Voce di Amazon Pinpoint](#)
- [Esempio: limitazione dell'accesso del progetto Amazon Pinpoint a indirizzi IP specifici](#)
- [Esempio: limitazione dell'accesso di Amazon Pinpoint in base ai tag](#)
- [Esempio: autorizzazione all'invio di e-mail da parte di Amazon Pinpoint mediante identità verificate in Amazon SES](#)

## Best practice delle policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Pinpoint nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente di IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

## Utilizzo della console Amazon Pinpoint

Per accedere alla console Amazon Pinpoint, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Pinpoint nel AWS tuo account. Se crei una policy basata su identità che applica autorizzazioni più restrittive rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli) associate a tale policy. Per garantire che tali entità possano utilizzare la console Amazon Pinpoint, collega una policy alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

La seguente policy di esempio fornisce l'accesso in sola lettura alla console Amazon Pinpoint in una regione specifica. AWS Include l'accesso in sola lettura ad altri servizi da cui dipende la console Amazon Pinpoint, come Amazon Simple Email Service (Amazon SES), IAM e Amazon Kinesis.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseConsole",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "firehose:ListDeliveryStreams",
        "iam:ListRoles",

```

```

        "kinesis:ListStreams",
        "s3:List*",
        "ses:Describe*",
        "ses:Get*",
        "ses:List*",
        "sns:ListTopics"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        }
    }
}
]
}

```

Nell'esempio di policy precedente, sostituisci *region* con il nome di una AWS Region e sostituisci *AccountID con l'ID* del AWS tuo account.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

## Esempio: accesso a un singolo progetto Amazon Pinpoint

È anche possibile creare policy di sola lettura che forniscono l'accesso solo a progetti specifici. La policy di esempio seguente consente agli utenti di effettuare l'accesso alla console e visualizzare una lista di progetti. Consente inoltre agli utenti di visualizzare informazioni sulle risorse correlate per altri servizi AWS da cui dipende la console Amazon Pinpoint, ad esempio Amazon SES, IAM e Amazon Kinesis. Tuttavia, la policy consente agli utenti di visualizzare ulteriori informazioni solo sul progetto specificato nella policy. Puoi modificare questa politica per consentire l'accesso a progetti o AWS regioni aggiuntivi.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",

```

```

    "Resource": "arn:aws:mobiletargeting:region:accountId:*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mobiletargeting:Get*",
      "mobiletargeting:List*"
    ],
    "Resource": [
      "arn:aws:mobiletargeting:region:accountId:apps/projectId",
      "arn:aws:mobiletargeting:region:accountId:apps/projectId/*",
      "arn:aws:mobiletargeting:region:accountId:reports"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "ses:Get*",
      "kinesis:ListStreams",
      "firehose:ListDeliveryStreams",
      "iam:ListRoles",
      "ses:List*",
      "sns:ListTopics",
      "ses:Describe*",
      "s3:List*"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "accountId"
      }
    }
  }
]
}

```

Nell'esempio precedente, sostituisci *region* con il nome di una AWS Region, sostituisci *AccountID* con l'ID del AWS tuo account e sostituisci *ProjectID* con l'ID del progetto Amazon Pinpoint a cui desideri fornire l'accesso.

Allo stesso modo, puoi creare politiche che concedano a un utente del tuo AWS account un accesso limitato in scrittura a uno dei tuoi progetti Amazon Pinpoint, ad esempio il progetto con l'ID del 810c7aab86d42fb2b56c8c966example progetto. In questo caso, vuoi consentire all'utente di

visualizzare, aggiungere e aggiornare i componenti del progetto, ad esempio segmenti e campagne, ma non di eliminare alcun componente.

Oltre a concedere le autorizzazioni per le operazioni `mobiletargeting:Get` e `mobiletargeting:List`, crea una policy che conceda le autorizzazioni per le operazioni seguenti: `mobiletargeting:Create`, `mobiletargeting:Update` e `mobiletargeting:Put`. Queste sono le autorizzazioni aggiuntive necessarie per creare e gestire la maggior parte dei componenti del progetto. Per esempio:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LimitedWriteProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*",
        "mobiletargeting:Create*",
        "mobiletargeting:Update*",
        "mobiletargeting:Put*"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example",
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example/*",
        "arn:aws:mobiletargeting:region:accountId:reports"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ses:Get*",
        "kinesis:ListStreams",
        "firehose:ListDeliveryStreams",
        "iam:ListRoles",
        "ses:List*"
      ]
    }
  ]
}
```

```

        "sns:ListTopics",
        "ses:Describe*",
        "s3:List*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        }
    }
}
]
}

```

## Esempio: visualizzazione di risorse Amazon Pinpoint in base ai tag

In una policy basata sulle identità puoi utilizzare le condizioni per controllare l'accesso alle risorse Amazon Pinpoint in base ai tag. In questa policy di esempio viene illustrato come creare questo tipo di policy per consentire la visualizzazione delle risorse Amazon Pinpoint. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag del servizio Owner è quello del nome utente dell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListResources",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ViewResourceIfOwner",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
    }
  ]
}

```

```

    "Resource": "arn:aws:mobiletargeting:*:*:*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Owner": "userName"
      },
      "StringEquals": {
        "aws:SourceAccount": "accountId"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:*"
      }
    }
  }
]
}

```

Puoi collegare questo tipo di policy agli utenti nel tuo account. Se un utente denominato `richard-roe` tenta di visualizzare una risorsa Amazon Pinpoint, la risorsa deve essere contrassegnata mediante il tag `Owner=richard-roe` o `owner=richard-roe`. In caso contrario l'accesso è negato. La chiave di tag di condizione `Owner` corrisponde a `Owner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

## Esempio: consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ]
    }
  ]
}

```

```

    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
}

```

## Esempi: autorizzazione dell'accesso alle azioni dell'API Amazon Pinpoint

In questa sezione vengono fornite policy di esempio che consentono l'accesso alle funzionalità disponibili nell'API Amazon Pinpoint, che è l'API primaria per Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la [documentazione di riferimento dell'API Amazon Pinpoint](#).

### Accesso in sola lettura

La seguente policy di esempio consente l'accesso in sola lettura a tutte le risorse del tuo account Amazon Pinpoint in una regione specifica. AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewAllResources",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    }
  ]
}

```

```

    }
  ]
}

```

Nell'esempio precedente, sostituisci *region* con il nome di una AWS Region e sostituisci *AccountID con l'ID* del AWS tuo account.

### Accesso amministratore

La policy di esempio seguente consente l'accesso completo a tutte le azioni e le risorse Amazon Pinpoint dell'account Amazon Pinpoint:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FullAccess",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    }
  ]
}

```

Nell'esempio precedente, sostituisci *accountId* con l'ID del tuo account AWS .

### Esempi: autorizzazione dell'accesso alle azioni dell'API SMS e Voce di Amazon Pinpoint

In questa sezione vengono fornite policy di esempio che consentono l'accesso alle funzionalità disponibili nell'API SMS e Voce di Amazon Pinpoint. Si tratta di un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la [documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint](#).

#### Accesso in sola lettura

La seguente policy di esempio consente l'accesso in sola lettura a tutte le azioni e le risorse dell'API Amazon Pinpoint SMS e Voice nel tuo account: AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceReadOnly",
      "Effect": "Allow",
      "Action": [
        "sms-voice:Get*",
        "sms-voice:List*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
        }
      }
    }
  ]
}

```

## Accesso amministratore

La seguente policy di esempio consente l'accesso completo a tutte le azioni e le risorse delle API Amazon Pinpoint SMS e Voice nel tuo AWS account:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceFullAccess",
      "Effect": "Allow",
      "Action": [
        "sms-voice:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {

```

```

        "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
    }
}
]
}

```

## Esempio: limitazione dell'accesso del progetto Amazon Pinpoint a indirizzi IP specifici

La policy di esempio seguente concede a qualsiasi utente le autorizzazioni per eseguire qualsiasi azione Amazon Pinpoint su un progetto specificato (*projectId*). La richiesta deve, tuttavia, avere origine dall'intervallo di indirizzi IP specificati nella condizione.

La condizione in questa istruzione identifica l'intervallo 54.240.143.\* di indirizzi Internet Protocol versione 4 (IPv4) consentiti, con un'unica eccezione: 54.240.143.188. Il Condition blocco utilizza le NotIpAddress condizioni IpAddress and e la chiave aws:SourceIp condition, che è una chiave di condizione AWS a livello globale. Per ulteriori informazioni su queste chiavi di condizione, consulta la sezione relativa alla [specificazione delle condizioni in una policy](#) nella Guida per l'utente di IAM. I valori IPv4 aws:SourceIp utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta [Operatori di condizione con indirizzo IP](#) nella Guida per l'utente di IAM.

```

{
  "Version": "2012-10-17",
  "Id": "AMZPinpointPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        },
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.188/32"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

## Esempio: limitazione dell'accesso di Amazon Pinpoint in base ai tag

La policy di esempio seguente concede le autorizzazioni per eseguire qualsiasi azione Amazon Pinpoint su un progetto specificato (*projectId*). Tuttavia, le autorizzazioni vengono concesse solo se la richiesta proviene da un utente il cui nome è un valore nel tag `Owner` della risorsa per il progetto, come specificato nella condizione.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "userName"
        }
      }
    }
  ]
}

```

## Esempio: autorizzazione all'invio di e-mail da parte di Amazon Pinpoint mediante identità verificate in Amazon SES

Quando verifichi un'identità e-mail, ad esempio un indirizzo e-mail o un dominio, tramite la console Amazon Pinpoint, tale identità viene configurata automaticamente in modo che possa essere utilizzata sia da Amazon Pinpoint che da Amazon SES. Tuttavia, se verifichi un'identità e-mail tramite Amazon SES e desideri utilizzare tale identità con Amazon Pinpoint, devi applicare una policy a tale identità.

La seguente policy di esempio concede ad Amazon Pinpoint l'autorizzazione a inviare e-mail utilizzando un'identità e-mail verificata tramite Amazon SES.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "PinpointEmail",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "ses:*",
      "Resource": "arn:aws:ses:region:accountId:identity/emailId",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:apps/*"
        }
      }
    }
  ]
}
```

Se utilizzi Amazon Pinpoint nella regione AWS GovCloud (Stati Uniti occidentali), utilizza invece il seguente esempio di policy:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "PinpointEmail",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "ses:*",
      "Resource": "arn:aws-us-gov:ses:us-gov-west-1:accountId:identity/emailId",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        }
      }
    }
  ]
}
```

```
    },
    "StringLike":{
      "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:accountId:apps/*"
    }
  }
]
```

## Ruoli IAM per attività comuni di Amazon Pinpoint

Un [ruolo IAM](#) è un'identità AWS Identity and Access Management (IAM) che puoi creare nel tuo AWS account e concedere autorizzazioni specifiche. Un ruolo IAM è un' AWS identità con politiche di autorizzazione che determinano ciò che l'identità può e non può fare. AWS Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque.

Inoltre, un ruolo non ha credenziali a lungo termine standard associate. Fornisce invece le credenziali di sicurezza temporanee per una sessione. Puoi utilizzare i ruoli IAM per delegare l'accesso a utenti, app, applicazioni o servizi che normalmente non hanno accesso alle tue AWS risorse.

Per questi motivi, puoi utilizzare i ruoli IAM per l'integrazione di Amazon Pinpoint con determinati servizi e risorse AWS associati al tuo account. Ad esempio, puoi consentire ad Amazon Pinpoint l'accesso alle definizioni di endpoint archiviate in un bucket Amazon Simple Storage Service (Amazon S3) e utilizzarle per i segmenti. Oppure potresti decidere di consentire ad Amazon Pinpoint lo streaming dei dati degli eventi in un flusso Amazon Kinesis per il tuo account. Allo stesso modo, potresti voler utilizzare i ruoli IAM per consentire alle app web o mobili di registrare endpoint o segnalare i dati di utilizzo per i progetti Amazon Pinpoint, senza AWS incorporare chiavi nelle app (dove possono essere difficili da ruotare e gli utenti possono potenzialmente estrarle).

Per questi scenari, è possibile delegare l'accesso ad Amazon Pinpoint utilizzando i ruoli IAM. Questa sezione illustra e fornisce esempi di attività Amazon Pinpoint comuni basate su ruoli IAM per utilizzare altri servizi AWS . Per informazioni sull'utilizzo dei ruoli IAM con app Web e per dispositivi mobili in modo più specifico, consulta [Fornire l'accesso a utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'utente di IAM.

### Argomenti

- [Ruolo IAM per l'importazione di endpoint o segmenti](#)
- [Ruolo IAM per l'esportazione di endpoint o segmenti](#)

- [Ruolo IAM per il recupero delle raccomandazioni da Amazon Personalize](#)
- [Ruolo IAM per lo streaming di eventi su Kinesis](#)
- [Ruolo IAM per l'invio di e-mail con Amazon SES](#)

## Ruolo IAM per l'importazione di endpoint o segmenti

Con Amazon Pinpoint, puoi definire un segmento di utenti importando le definizioni degli endpoint da un bucket Amazon Simple Storage Service (Amazon S3) nel tuo account. AWS Prima di eseguire l'importazione, è necessario delegare le autorizzazioni necessarie ad Amazon Pinpoint. A tale scopo, crei un ruolo AWS Identity and Access Management (IAM) e al ruolo colleghi le seguenti policy:

- La policy gestita `AmazonS3ReadOnlyAccess` AWS. Questa policy è creata e gestita da AWS e garantisce l'accesso in sola lettura al tuo bucket Amazon S3.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, puoi utilizzare Amazon Pinpoint per importare i segmenti da un bucket Amazon S3. Per informazioni sulla creazione di bucket e file di endpoint e sull'importazione di un segmento tramite la console, consulta [Importazione di segmenti nella Guida per l'utente di Amazon Pinpoint](#). Per un esempio di come importare un segmento a livello di codice utilizzando il, consulta questa guida. AWS SDK for Java [Importazione di segmenti](#)

### Creazione del ruolo IAM (AWS CLI)

Completa i seguenti passaggi per creare il ruolo IAM utilizzando AWS Command Line Interface (AWS CLI). Se non hai installato il AWS CLI, consulta [Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

Per creare il ruolo IAM utilizzando AWS CLI

1. Crea un file JSON contenente la policy di attendibilità per il ruolo e salva il file in locale. È possibile utilizzare la seguente politica di attendibilità.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
```

```

        "Principal": {
            "Service": "pinpoint.amazonaws.com"
        },
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "accountId"
            },
            "ArnLike": {
                "arn:aws:mobiletargeting:region:accountId:apps/application-id"
            }
        }
    }
}
]
}

```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci la *AWS regione* con la regione in cui utilizzi Amazon Pinpoint.
  - Sostituisci *AccountID* con l'ID univoco del tuo AWS account.
  - Sostituisci *application-id* con l'ID univoco del progetto.
2. Usare il comando [create-role](#) per creare il ruolo e collegare la policy di attendibilità:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
file://PinpointImportTrustPolicy.json
```

Dopo il prefisso `file://` specifica il percorso del file JSON che contiene la policy di attendibilità.

Dopo aver eseguito questo comando, nel terminale viene visualizzato un output simile al seguente:

```

{
  "Role": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
            "Service": "pinpoint.amazonaws.com"
          }
        }
      ]
    }
  }
}

```

```

        },
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "accountId"
            },
            "ArnLike": {
                "aws:SourceArn":
"arn:aws:mobiletargeting:region:accountId:apps/application-id"
            }
        }
    ]
},
"RoleId": "AIDACKCEVSQ6C2EXAMPLE",
"CreateDate": "2016-12-20T00:44:37.406Z",
"RoleName": "PinpointSegmentImport",
"Path": "/",
"Arn": "arn:aws:iam::accountId:role/PinpointSegmentImport"
}
}

```

3. Usa il [attach-role-policy](#) comando per allegare la politica AmazonS3ReadOnlyAccess AWS gestita al ruolo:

```

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport

```

## Ruolo IAM per l'esportazione di endpoint o segmenti

È possibile ottenere un elenco di endpoint creando un processo di esportazione. Quando crei un processo di esportazione, devi specificare un ID progetto e, facoltativamente, un ID segmento. Amazon Pinpoint esporta quindi un elenco degli endpoint associati al progetto o al segmento in un bucket Amazon Simple Storage Service (Amazon S3). Il file risultante contiene un elenco di endpoint in formato JSON con i relativi attributi, ad esempio canale, indirizzo, stato del consenso, data di creazione e ID degli endpoint.

Per creare un processo di esportazione, è necessario configurare un ruolo IAM che consenta ad Amazon Pinpoint di scrivere in un bucket Amazon S3. Il processo di configurazione del ruolo prevede due fasi:

1. Creazione di una policy IAM che consenta a un'entità (in questo caso, Amazon Pinpoint) di scrivere in un determinato bucket Amazon S3.
2. Creare un ruolo IAM e collegarvi la policy.

Questo argomento contiene le procedure per completare entrambe le fasi. Queste procedure presuppongono che sia già stato creato un bucket Amazon S3, con una cartella al suo interno, per archiviare i segmenti esportati. Per ulteriori informazioni sulla creazione di un bucket, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

Queste procedure si basano anche sul presupposto che l' AWS Command Line Interface (AWS CLI) sia già stata installata e configurata. Per informazioni sulla configurazione di AWS CLI, vedere [Installazione di AWS CLI nella](#) Guida per l'AWS Command Line Interface utente.

### Passaggio 1: creare la policy IAM

Una policy IAM definisce le autorizzazioni per un'entità, ad esempio un'identità o una risorsa. Per creare un ruolo per l'esportazione di endpoint Amazon Pinpoint, devi creare una policy che conceda le autorizzazioni di scrittura per una determinata cartella in un determinato bucket Amazon S3. Il seguente esempio di policy segue la pratica di sicurezza che prevede la concessione del privilegio minimo, ovvero concede solo le autorizzazioni necessarie per eseguire una singola attività.

### Co,e creare la policy IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUserToSeeBucketListInTheConsole",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::*" ]
    },
    {
      "Sid": "AllowRootAndHomeListingOfBucket",
      "Action": [
        "s3:ListBucket"
      ]
    }
  ]
}
```

```

    ],
    "Effect": "Allow",
    "Resource": [ "arn:aws:s3:::example-bucket" ],
    "Condition": {
      "StringEquals": {
        "s3:delimiter": [ "/" ],
        "s3:prefix": [
          "",
          "Exports/"
        ]
      }
    }
  },
  {
    "Sid": "AllowListingOfUserFolder",
    "Action": [
      "s3:ListBucket"
    ],
    "Effect": "Allow",
    "Resource": [ "arn:aws:s3:::example-bucket" ],
    "Condition": {
      "StringLike": {
        "s3:prefix": [
          "Exports/*"
        ]
      }
    }
  },
  {
    "Sid": "AllowAllS3ActionsInUserFolder",
    "Action": [ "s3:*" ],
    "Effect": "Allow",
    "Resource": [ "arn:aws:s3:::example-bucket/Exports/*" ]
  }
]
}

```

Nel codice riportato sopra, sostituisci tutte le istanze di *example-bucket* con il nome del bucket Amazon S3 contenente la cartella in cui vuoi esportare le informazioni sui segmenti. Sostituisci anche tutte le istanze di *Exports* con il nome della cartella stessa.

Al termine, salva il file come `s3policy.json`.

- Utilizzando il AWS CLI, accedete alla directory in cui si trova il `s3policy.json` file. Immetti il comando seguente per creare la policy:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
file://s3policy.json
```

Se la creazione della policy è andata a buon fine, verrà visualizzato un output simile al seguente:

```
{
  "Policy": {
    "CreateDate": "2018-04-11T18:44:34.805Z",
    "IsAttachable": true,
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PolicyId": "ANPAJ2YJQRJCG3EXAMPLE",
    "UpdateDate": "2018-04-11T18:44:34.805Z",
    "Arn": "arn:aws:iam::123456789012:policy/s3ExportPolicy",
    "PolicyName": "s3ExportPolicy",
    "Path": "/"
  }
}
```

Copia l'Amazon Resource Name (ARN) della policy

(`arn:aws:iam::123456789012:policy/s3ExportPolicy` nell'esempio precedente). Nella sezione successiva dovrai specificare questo ARN durante la creazione del ruolo.

#### Note

Se viene visualizzato un messaggio in cui è indicato che l'account non è autorizzato a eseguire l'operazione `CreatePolicy`, all'account utente è necessario collegare una policy che consenta di creare nuovi ruoli e nuove policy IAM. Per ulteriori informazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente di IAM .

## Passaggio 2: creare il ruolo IAM

Una volta creata una policy IAM, è possibile creare un ruolo e collegare la policy a tale ruolo. Ogni ruolo IAM contiene una policy di attendibilità, ovvero un set di regole che specifica le entità che

possono assumere il ruolo specifico. In questa sezione viene creata una policy di attendibilità che consente ad Amazon Pinpoint di assumere il ruolo. In seguito si creerà il ruolo stesso e vi si collegherà la policy creata nella sezione precedente.

Per creare il ruolo IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:mobiletargeting:region:accountId:apps/applicationId"
        }
      }
    }
  ]
}
```

Salva il file con nome `trustpolicy.json`.

2. Utilizzando AWS CLI, accedete alla directory in cui si trova il `trustpolicy.json` file. Immetti il comando seguente per creare un nuovo ruolo:

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
file://trustpolicy.json
```

3. Nella riga di comando immetti il comando seguente per collegare la policy creata nella sezione precedente al ruolo che hai appena creato:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/s3ExportPolicy --role-name s3ExportRole
```

Nel comando precedente, sostituisci *arn:aws:iam::123456789012:policy/s3 con ExportPolicy l'ARN della policy creata* nella sezione precedente.

## Ruolo IAM per il recupero delle raccomandazioni da Amazon Personalize

Puoi configurare Amazon Pinpoint per recuperare i dati delle raccomandazioni da una soluzione Amazon Personalize implementata come campagna Amazon Personalize. Puoi utilizzare questi dati per inviare suggerimenti personalizzati ai destinatari dei messaggi in base agli attributi e al comportamento di ciascun destinatario. Per ulteriori informazioni, consulta [Modelli di machine learning](#) nella Guida per l'utente di Amazon Pinpoint.

Prima di poter recuperare i dati delle raccomandazioni da una campagna Amazon Personalize, devi creare un ruolo AWS Identity and Access Management (IAM) che consenta ad Amazon Pinpoint di recuperare i dati dalla campagna. Amazon Pinpoint può creare questo ruolo automaticamente quando utilizzi la console per configurare un modello di raccomandazione in Amazon Pinpoint. In alternativa, è possibile creare questo ruolo manualmente.

Per creare manualmente il ruolo, utilizza l'API IAM per completare le fasi seguenti:

1. Creare una policy IAM che consenta a un'entità (in questo caso, Amazon Pinpoint) di recuperare i dati delle raccomandazioni da una campagna Amazon Personalize.
2. Creare un ruolo IAM e collegarvi la policy IAM.

Questo argomento spiega come completare questi passaggi AWS CLI utilizzando (). AWS Command Line Interface Si presuppone che la soluzione Amazon Personalize sia già stata creata e implementata come campagna Amazon Personalize. Per informazioni sulla creazione e sull'implementazione di una campagna, consulta l'argomento relativo alla [creazione di una campagna](#) nella Guida per gli sviluppatori di Amazon Personalize.

Questo argomento si basa sul presupposto che la AWS CLI sia già stata installata e configurata. Per informazioni sulla configurazione di AWS CLI, vedere [Installazione di AWS CLI nella Guida per l'utente](#) di AWS Command Line Interface.

## Passaggio 1: creare la policy IAM

Una policy IAM definisce le autorizzazioni per un'entità, ad esempio un'identità o una risorsa. Per creare un ruolo che consenta ad Amazon Pinpoint di recuperare i dati delle raccomandazioni da una campagna Amazon Personalize, devi innanzitutto creare una policy IAM per il ruolo. Questa policy deve consentire ad Amazon Pinpoint di:

- Recuperare le informazioni di configurazione per la soluzione distribuita dalla campagna (`DescribeSolution`).
- Controllare lo stato della campagna (`DescribeCampaign`).
- Recuperare i dati dei suggerimenti dalla campagna (`GetRecommendations`).

Nella procedura seguente, la policy di esempio consente questo accesso per una specifica soluzione Amazon Personalize che è stata implementata da una specifica campagna Amazon Personalize.

Come creare la policy IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RetrieveRecommendationsOneCampaign",
      "Effect": "Allow",
      "Action": [
        "personalize:DescribeSolution",
        "personalize:DescribeCampaign",
        "personalize:GetRecommendations"
      ],
      "Resource": [
        "arn:aws:personalize:region:accountId:solution/solutionId",
        "arn:aws:personalize:region:accountId:campaign/campaignId"
      ]
    }
  ]
}
```

Nell'esempio precedente, sostituisci il testo in *corsivo* con le tue informazioni:

- *region*: nome della regione AWS che ospita la soluzione e la campagna Amazon Personalize.
  - *accountId*: il tuo ID Account AWS .
  - *solutionId*: ID di risorsa univoco per la soluzione Amazon Personalize implementata dalla campagna.
  - *campaignId*: ID risorsa univoco per la campagna Amazon Personalize da cui recuperare i dati delle raccomandazioni.
2. Al termine, salva il file come `RetrieveRecommendationsPolicy.json`.
  3. Utilizzando l'interfaccia a riga di comando, passa alla directory in cui è stato salvato il file `RetrieveRecommendationsPolicy.json`.
  4. Immetti il comando seguente per creare una policy e denominarla `RetrieveRecommendationsPolicy`. Per utilizzare un nome diverso, *RetrieveRecommendationsPolicy* cambiatelo con il nome desiderato.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document file://RetrieveRecommendationsPolicy.json
```

#### Note

Se ricevi un messaggio in cui è indicato che l'account non è autorizzato a eseguire l'operazione `CreatePolicy`, all'account utente devi collegare una policy che consenta di creare nuovi ruoli e nuove policy IAM per l'account. Per ulteriori informazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente di IAM .

5. Copia l'Amazon Resource Name (ARN) della policy (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` nell'esempio precedente). Nella sezione successiva, avrai bisogno di questo ARN per creare il ruolo IAM.

## Passaggio 2: creare il ruolo IAM

Una volta creata la policy IAM, puoi creare un ruolo IAM a cui collegarla.

Ogni ruolo IAM contiene una policy di attendibilità, che è composta da un set di regole che specifica le entità che possono assumere tale ruolo. In questa sezione viene creata una policy di attendibilità

che consente ad Amazon Pinpoint di assumere il ruolo. Successivamente, crei il ruolo vero e proprio. Quindi, colleghi tale policy al ruolo.

Per creare il ruolo IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "accountId"
        },
        "ArnLike": {
          "AWS:SourceArn":
            "arn:aws:mobiletargeting:region:accountId:apps/*"
        }
      }
    }
  ]
}
```

2. Salva il file con nome `RecommendationsTrustPolicy.json`.
3. Utilizzando l'interfaccia a riga di comando, passa alla directory in cui è stato salvato il file `RecommendationsTrustPolicy.json`.
4. Immetti il comando seguente per creare un nuovo ruolo e denominarlo `PinpointRoleforPersonalize`. Per usare un nome diverso, *PinpointRoleforPersonalize* scegli il nome che desideri.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-document file://RecommendationsTrustPolicy.json
```

5. Immetti il comando seguente per collegare la policy creata nella sezione precedente al ruolo che hai appena creato:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

Nel comando precedente, sostituisci *arn:aws:iam::123456789012:policy/* con *l'ARN della policy creata RetrieveRecommendationsPolicy nella sezione precedente*. Inoltre, sostituiscilo *PinpointRoleforPersonalize* con il nome del ruolo specificato nel passaggio 4, se hai specificato un nome diverso per il ruolo.

## Ruolo IAM per lo streaming di eventi su Kinesis

Amazon Pinpoint può inviare automaticamente i dati sull'utilizzo delle app, o dati sugli eventi, dalla tua app a un flusso di dati Amazon Kinesis o a un flusso di distribuzione Amazon Data Firehose nel tuo account. AWS Prima che Amazon Pinpoint possa avviare lo streaming dei dati degli eventi, devi delegare le autorizzazioni necessarie ad Amazon Pinpoint.

Se usi la console per configurare lo streaming degli eventi, Amazon Pinpoint crea automaticamente un ruolo AWS Identity and Access Management (IAM) con le autorizzazioni necessarie. Per ulteriori informazioni, consulta [Streaming di eventi Amazon Pinpoint su Kinesis](#) nella Guida per l'utente di Amazon Pinpoint.

Se vuoi creare manualmente il ruolo, collega le policy seguenti al ruolo:

- Una policy di autorizzazione che consenta ad Amazon Pinpoint di inviare dati di eventi al flusso.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, è possibile configurare Amazon Pinpoint per l'invio automatico degli eventi al flusso. Per ulteriori informazioni sul tagging, consulta [Streaming di eventi Amazon Pinpoint su Kinesis](#) in questa guida.

### Creazione del ruolo IAM (AWS CLI)

Completa la procedura seguente per creare manualmente un ruolo IAM mediante l' AWS Command Line Interface (AWS CLI). Per informazioni su come creare il ruolo utilizzando la console Amazon Pinpoint, consulta [Streaming di eventi Amazon Pinpoint su Kinesis](#) nella Guida per l'utente di Amazon Pinpoint.

Se non hai installato il AWS CLI, consulta [Installazione di AWS CLI nella Guida per l'utente](#). AWS Command Line Interface È inoltre necessario aver creato uno stream Kinesis o uno stream Firehose.

Per informazioni sulla creazione di queste risorse, consulta [Creating and Managing Streams](#) nella Amazon Kinesis Data Streams Developer Guide o [Creazione di un flusso di distribuzione Amazon Data Firehose nella Amazon Data Firehose](#) Developer Guide.

Per creare il ruolo IAM utilizzando il AWS CLI

1. Crea un nuovo file. Incolla la seguente politica nel documento e apporta le seguenti modifiche:
  - Sostituisci la *AWS regione* con la regione in cui utilizzi Amazon Pinpoint.
  - Sostituisci *AccountID* con l'ID univoco del tuo AWS account.
  - Sostituisci *ApplicationID* con l'ID univoco del progetto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:mobiletargeting:region:accountId:apps/applicationId"
        }
      }
    }
  ]
}
```

Al termine, salva il file come `PinpointEventStreamTrustPolicy.json`.

2. Usa il comando [create-role](#) per creare il ruolo e collegare la policy di attendibilità:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

### 3. Crea un nuovo file contenente la policy di autorizzazione per il ruolo.

Se stai configurando Amazon Pinpoint per inviare dati a un flusso Kinesis, incolla la seguente policy nel file e sostituisci la seguente:

- Sostituisci la *AWS regione* con la regione in cui utilizzi Amazon Pinpoint.
- Sostituisci *AccountID* con l'ID univoco del tuo AWS account.
- Sostituisci *streamName* con il nome del tuo stream Kinesis.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [
      "kinesis:PutRecords",
      "kinesis:DescribeStream"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:kinesis:region:accountId:stream/streamName"
    ]
  }
}
```

In alternativa, se stai configurando Amazon Pinpoint per inviare dati a uno stream Firehose, incolla la seguente policy nel file e sostituisci la seguente:

- Sostituisci la *AWS regione* con la regione in cui utilizzi Amazon Pinpoint.
- Sostituisci *AccountID* con l'ID univoco del tuo AWS account.
- Sostituisci *delivery-stream-name* con il nome del tuo stream Firehose.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "firehose:PutRecordBatch",
      "firehose:DescribeDeliveryStream"
    ],
    "Resource": [
```

```
        "arn:aws:firehose:region:accountId:deliverystream/delivery-stream-name"
    ]
}
}
```

Al termine, salva il file come `PinpointEventStreamPermissionsPolicy.json`.

4. Utilizza il comando [put-role-policy](#) per collegare la policy di autorizzazione al ruolo:

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-  
name PinpointEventStreamPermissionsPolicy --policy-document file://  
PinpointEventStreamPermissionsPolicy.json
```

## Ruolo IAM per l'invio di e-mail con Amazon SES

Amazon Pinpoint utilizza le tue risorse Amazon SES per inviare e-mail relative alla tua campagna o al tuo percorso. Prima che Amazon Pinpoint possa utilizzare le tue risorse Amazon SES per inviare e-mail, devi concedere le autorizzazioni necessarie ad Amazon Pinpoint. Il tuo account deve disporre delle autorizzazioni `iam:PutRolePolicy` e dei `iam:UpdateAssumeRolePolicy` permessi per aggiornare o creare ruoli IAM.

La console Amazon Pinpoint può creare automaticamente un ruolo AWS Identity and Access Management (IAM) con le autorizzazioni richieste. Per ulteriori informazioni, consulta [Creazione di un ruolo di invio per l'orchestrazione delle e-mail nella Guida](#) per l'utente di Amazon Pinpoint.

Se vuoi creare manualmente il ruolo, collega le policy seguenti al ruolo:

- Una politica di autorizzazioni che concede ad Amazon Pinpoint l'accesso alle tue risorse Amazon SES.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, puoi configurare Amazon Pinpoint per utilizzare le tue risorse Amazon SES.

Puoi testare le politiche IAM con il simulatore di policy IAM. Per ulteriori informazioni, consulta [Testare le policy IAM con il simulatore di policy IAM](#) nella [IAM User Guide](#).

### Creazione del ruolo IAM (AWS Management Console)

Completa i seguenti passaggi per creare manualmente un ruolo IAM per la tua campagna o il tuo percorso di invio di e-mail.

1. Crea una nuova policy di autorizzazione seguendo le istruzioni riportate in [Creazione di policy using the JSON editor](#) nella [IAM User Guide](#).
  - Nella [fase 5](#), utilizza la seguente politica di autorizzazione per il ruolo IAM.
    - Sostituisci la *partizione* con la partizione in cui si trova la risorsa. Per lo standard Regioni AWS, la partizione è. aws Se sono presenti risorse in altre partizioni, la partizione è aws-partitionname. Ad esempio, la partizione per le risorse negli AWS GovCloud Stati Uniti occidentali è. aws-us-gov
    - Sostituisci la *regione* con il nome di Regione AWS quella che ospita il progetto Amazon Pinpoint.
    - Sostituisci *AccountID* con l'ID univoco per il tuo. Account AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PinpointUsesSESForEmailSends",
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": [
        "arn:partition:ses:region:accountId:identity/*",
        "arn:partition:ses:region:accountId:configuration-set/*"
      ]
    }
  ]
}
```

2. Crea una nuova policy di fiducia seguendo le istruzioni riportate nella sezione [Creazione di un ruolo utilizzando policy di fiducia personalizzate](#) nella [IAM User Guide](#).
  - a. Nella [fase 4](#), utilizza la seguente politica di fiducia.
    - Sostituisci *AccountID* con l'ID univoco per il tuo. Account AWS

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowPinpoint",
    "Effect": "Allow",
    "Principal": {
      "Service": "pinpoint.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "accountId"
      }
    }
  }
]
```

- b. Nel [passaggio 11](#), aggiungi la politica di autorizzazione creata nel passaggio precedente.

## Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon Pinpoint e IAM.

### Argomenti

- [Non sono autorizzato a eseguire un'azione in Amazon Pinpoint](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon Pinpoint](#)

### Non sono autorizzato a eseguire un'azione in Amazon Pinpoint

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` cerca di utilizzare la console per visualizzare i dettagli relativi a un progetto ma non dispone di autorizzazioni `mobiletargeting:GetApp`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
mobiletargeting:GetApp on resource: my-example-project
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `my-example-project` utilizzando l'azione `mobiletargeting:GetApp`.

## Non sono autorizzato a eseguire iam: PassRole

Se viene visualizzato un errore indicante che non disponi dell'autorizzazione per eseguire l'azione `iam:PassRole`, per poter passare un ruolo ad Amazon Pinpoint dovrai aggiornare le policy.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'azione in Amazon Pinpoint. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon Pinpoint

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo.

Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Pinpoint supporta queste funzionalità, consulta [Come Amazon Pinpoint funziona con IAM](#).
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente IAM.

## Registrazione di log e monitoraggio in Amazon Pinpoint

La registrazione di log e il monitoraggio sono importanti per garantire l'affidabilità, la disponibilità e le prestazioni dei progetti Amazon Pinpoint e di altri tipi di risorse Amazon Pinpoint. È necessario registrare e raccogliere dati di monitoraggio da tutte le parti dei progetti e delle risorse Amazon Pinpoint per eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti che possono aiutarti a registrare e raccogliere questi dati e a rispondere a potenziali incidenti:

### AWS CloudTrail

Amazon Pinpoint si integra con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese in Amazon Pinpoint da un utente, un ruolo o un altro servizio. AWS include le azioni dalla console Amazon Pinpoint e le chiamate programmatiche alle operazioni delle API Amazon Pinpoint. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare quali richieste sono state fatte ad Amazon Pinpoint. Per ogni richiesta, è possibile identificare quando è stata effettuata, l'indirizzo IP da cui è stata effettuata, chi l'ha effettuata e ulteriori dettagli. Per ulteriori

informazioni sul tagging, consulta [Registrazione delle chiamate API Amazon Pinpoint con AWS CloudTrail](#) in questa guida.

## Amazon CloudWatch

Puoi utilizzare Amazon CloudWatch per raccogliere, visualizzare e analizzare diverse metriche importanti relative al tuo account Amazon Pinpoint e ai tuoi progetti. Puoi anche utilizzarlo CloudWatch per creare allarmi che ti avvisano se il valore di una metrica soddisfa determinate condizioni e rientra o supera una soglia da te definita. Se crei un allarme, CloudWatch invia una notifica a un argomento di Amazon Simple Notification Service (Amazon SNS) da te specificato. Per ulteriori informazioni, consulta [Monitoring Amazon Pinpoint with amazon CloudWatch nella Amazon Pinpoint User Guide](#).

## AWS Health Pannelli di controllo

Utilizzando i AWS Health dashboard, puoi controllare e monitorare lo stato del tuo ambiente Amazon Pinpoint. Per verificare lo stato generale del servizio Amazon Pinpoint, utilizza il AWS Service Health Dashboard. Per controllare, monitorare e visualizzare i dati storici su eventuali eventi o problemi che potrebbero influire in modo più specifico sull' AWS ambiente, utilizza la AWS Personal Health Dashboard. Per ulteriori informazioni su questi dashboard, consulta la [Guida dell'utente di AWS Health](#).

## AWS Trusted Advisor

AWS Trusted Advisor ispeziona l' AWS ambiente e fornisce consigli sulle opportunità per colmare le lacune di sicurezza, migliorare la disponibilità e le prestazioni del sistema e risparmiare denaro. Tutti i AWS clienti hanno accesso a una serie di controlli di Trusted Advisor base. I clienti che dispongono di un piano di supporto Business o Enterprise hanno accesso a Trusted Advisor controlli aggiuntivi.

Molti di questi controlli possono aiutarti a valutare il livello di sicurezza delle tue risorse Amazon Pinpoint nell'ambito generale del AWS tuo account. Ad esempio, il set principale di controlli Trusted Advisor include quanto segue:

- Configurazioni di registrazione per il tuo AWS account, per ogni regione supportata. AWS
- Autorizzazioni di accesso per i bucket Amazon Simple Storage Service (Amazon S3) che potrebbero contenere file importati in Amazon Pinpoint per creare i segmenti.
- Utilizzo di AWS Identity and Access Management utenti, gruppi e ruoli per controllare l'accesso alle risorse di Amazon Pinpoint.
- Configurazioni e impostazioni delle policy IAM che potrebbero compromettere la sicurezza del tuo AWS ambiente e delle risorse Amazon Pinpoint.

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella Guida per l'utente di AWS Support .

## Convalida della conformità per Amazon Pinpoint

Revisori di terze parti valutano la sicurezza e la conformità di Amazon Pinpoint nell'ambito di più programmi di conformità AWS . Questi includono AWS System and Organization Controls (SOC), FedRAMP, HIPAA, ISO/IEC 27001:2013 per i controlli di gestione della sicurezza, ISO/IEC 27017:2015 per i controlli specifici del cloud, ISO/IEC 27018:2014 per la protezione dei dati personali, ISO/IEC 9001:2015 per i sistemi di gestione della qualità e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di programmi di conformità specifici, vedere [AWS Servizi nell'ambito del programma di conformitàAWS](#) . Per informazioni generali, vedere programmi di [AWS conformità, programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) Scaricamento dei . AWS

La tua responsabilità di conformità quando utilizzi Amazon Pinpoint è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive](#) su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Whitepaper [sull'architettura per la sicurezza e la conformità HIPAA: questo white paper](#) descrive in che modo le aziende possono utilizzare per creare applicazioni conformi all'HIPAA. AWS
- AWS risorse per [la conformità e risorse per la conformità](#): questa raccolta di riguardare il settore e la località in cui operate.
- [Valutazione delle risorse in base alle regole contenute](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Amazon Pinpoint è un servizio idoneo alla AWS normativa HIPAA quando i clienti utilizzano i canali di comunicazione appropriati. Per utilizzare Amazon Pinpoint per eseguire carichi di lavoro contenenti

informazioni sanitarie protette (PHI) come definito dall'HIPAA e dalla legislazione e dalle normative associate, è necessario utilizzare il canale e-mail, il canale di notifica push o il canale SMS per inviare messaggi contenenti informazioni sanitarie protette. Se utilizzi il canale SMS per inviare messaggi che contengono PHI, devi inviare tali messaggi utilizzando un [codice breve dedicato](#) che hai richiesto per il tuo AWS account allo scopo esplicito di inviare messaggi che contengano o possano contenere PHI. Il canale vocale non è idoneo alla AWS normativa HIPAA; non utilizzare il canale vocale per inviare messaggi contenenti PHI.

## Resilienza in Amazon Pinpoint

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle architetture di riferimento, consulta il manuale relativo all'[architettura resiliente di Amazon Pinpoint](#).

[Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura globale.AWS](#)

## Sicurezza dell'infrastruttura in Amazon Pinpoint

In quanto servizio gestito, Amazon Pinpoint è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Pinpoint attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Anche se è possibile eseguire queste chiamate API da qualsiasi percorso di rete, Amazon Pinpoint supporta le policy di accesso basate su risorse. Queste policy possono includere restrizioni basate sull'indirizzo IP di origine. Per ulteriori informazioni su questo tipo di policy, consulta [Gestione dell'accesso con policy](#).

Inoltre, puoi configurare e utilizzare varie funzionalità di AWS sicurezza per controllare l'accesso alle risorse di Amazon Pinpoint da qualsiasi app mobile o Web che integri con Amazon Pinpoint. Sono incluse le restrizioni sulle chiamate API per attività quali l'aggiunta di endpoint, l'aggiornamento dei dati degli endpoint, l'invio dei dati degli eventi e la creazione di report sui dati di utilizzo.

Per utilizzare queste funzionalità, ti consigliamo di utilizzare gli SDK o le AWS Amplify JavaScript librerie per AWS dispositivi mobili per integrare app mobili e Web con Amazon Pinpoint. Per le app Android o iOS, ti consigliamo di utilizzare rispettivamente il AWS Mobile SDK for Android o AWS Mobile SDK for iOS. Per le app mobili o web JavaScript basate su dispositivi mobili, ti consigliamo di utilizzare la AWS Amplify JavaScript Library for the Web o la AWS Amplify JavaScript Library for React Native. Per ulteriori informazioni su queste risorse, consulta [Guida introduttiva agli SDK AWS per dispositivi mobili](#), [Guida introduttiva alla libreria AWS Amplify per il Web](#) e [Guida introduttiva alla libreria AWS Amplify per React Native](#).

## Analisi della configurazione e delle vulnerabilità in Amazon Pinpoint

In quanto servizio gestito, Amazon Pinpoint è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper [Amazon Web Services: panoramica dei processi di sicurezza](#). Ciò significa che AWS gestisce ed esegue attività e procedure di sicurezza di base per rafforzare, applicare patch, aggiornare e mantenere in altro modo l'infrastruttura sottostante per il tuo account e le tue risorse Amazon Pinpoint. Queste procedure sono state riviste e certificate dalle terze parti appropriate.

Per ulteriori informazioni, consulta le seguenti risorse:

- [Convalida della conformità per Amazon Pinpoint](#)
- [Modello di responsabilità condivisa](#)
- [Amazon Web Services: panoramica dei processi di sicurezza](#) (whitepaper)

## Best practice di sicurezza per Amazon Pinpoint

Utilizza gli account AWS Identity and Access Management (IAM) per controllare l'accesso Amazon Pinpoint alle operazioni API, in particolare alle operazioni che creano, modificano o Amazon Pinpoint eliminano risorse. Per quanto riguarda l' Amazon Pinpoint API, tali risorse includono progetti, campagne e viaggi. Per l'API SMS e Voce di Amazon Pinpoint , tali risorse includono numeri di telefono, pool e set di configurazione.

- Crea un utente individuale per ogni persona che gestisce Amazon Pinpoint le risorse, incluso te stesso. Non utilizzare credenziali AWS root per gestire le risorse di Amazon Pinpoint.
- Assegna a ciascun utente un set minimo di autorizzazioni richieste per eseguire le proprie mansioni.
- Utilizza gruppi IAM per gestire in modo efficace le autorizzazioni per più utenti.
- Ruota periodicamente le credenziali IAM.

Per ulteriori informazioni sulla Amazon Pinpoint sicurezza, consulta [la sezione Sicurezza in Amazon Pinpoint](#). Per ulteriori informazioni su IAM, consulta [AWS Identity and Access Management](#). Per informazioni sulle best practice di IAM, consulta [Best practice di IAM](#).

# Quote di Amazon Pinpoint

Le sezioni seguenti elencano e descrivono le quote, precedentemente definite limiti, applicabili alle risorse e alle operazioni di Amazon Pinpoint. Alcune quote possono essere aumentate, mentre altre no. Per determinare se è possibile richiedere un aumento per una quota, consulta la colonna Possibilità di incremento in ogni sezione.

## Argomenti

- [Quote per i progetti](#)
- [Quote di richiesta API](#)
- [Quote per le campagne](#)
- [Quote per l'e-mail](#)
- [Quote per gli endpoint](#)
- [Quote per l'importazione degli endpoint](#)
- [Quote per l'inserimento di eventi](#)
- [Quote per i viaggi](#)
- [Quote di Lambda](#)
- [Quote di machine learning](#)
- [Quote per i modelli di messaggio](#)
- [Quote per le notifiche push](#)
- [Quote per i messaggi in-app](#)
- [Quote per i segmenti](#)
- [Quote per SMS](#)
- [Quote 10DLC](#)
- [Quote per i messaggi vocali](#)
- [Richiesta di aumento delle quote](#)

## Quote per i progetti

Nella tabella seguente sono elencate le quote relative ai progetti in Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Progetti	In ognuno Regione AWS puoi avere fino a 100 progetti.	No

## Quote di richiesta API

Amazon Pinpoint implementa quote che limitano la dimensione e il numero di richieste che puoi inviare all'API Amazon Pinpoint dal tuo account. AWS

La dimensione massima di un payload di invocazione (richiesta e risposta) è di 7 MB, se non diversamente specificato per un particolare tipo di risorsa. Per determinare se una risorsa dispone di una quota diversa, vedere la sezione appropriata di questo argomento per quel tipo di risorsa.

Il numero massimo di richieste varia in base al tipo di quota e all'operazione dell'API. Amazon Pinpoint implementa due tipi di quote per le richieste API:

- **Quote di velocità:** definite anche limiti di frequenza, questo tipo di quote definisce il numero massimo di richieste che è possibile effettuare al secondo per una determinata operazione. Controlla la frequenza di invio o ricezione delle richieste per account.
- **Quote di espansione:** note anche come limiti di espansione o capacità di espansione, questo tipo di quota definisce il numero massimo di richieste simultaneamente in corso per un account.

Nella tabella seguente sono elencate le quote tariffarie e di espansione per l'API Amazon Pinpoint.

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300
CreatePushTemplate	10

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4.000
SendUsersMessages	6.000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10
UpdateVoiceTemplate	10

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
Tutte le altre operazioni	300

Nella tabella seguente sono riportate elenca le quote per l'importazione di file per `CreateImportJob`.

Operazione	Quota predefinita	Idoneità all'incremento
Numero massimo di file di importazione	10.000 file per processo di importazione	No

Se si supera una di queste quote, Amazon Pinpoint applica la limitazione della larghezza di banda della rete alla richiesta, ovvero rifiuta una richiesta altrimenti valida e restituisce un errore `TooManyRequests`. La limitazione della larghezza di banda della rete viene calcolata in base al numero totale di richieste effettuate dall'account per un'operazione specifica in una Regione AWS data. Inoltre, le decisioni di throttling vengono calcolate in modo indipendente per ogni operazione. Ad esempio, se Amazon Pinpoint applica la limitazione della larghezza di banda della rete a una richiesta per l'operazione `SendMessage`, una richiesta simultanea per l'operazione `UpdateEndpoint` può essere completata correttamente.

## Quote per le campagne

Le quote seguenti si applicano alla risorsa [Campagne](#) dell'API Amazon Pinpoint.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta [Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas](#).

Risorsa	Quota predefinita	Idoneità all'incremento
Campagne attive	200 per account	No

Risorsa	Quota predefinita	Idoneità all'incremento
	<p> <b>Note</b></p> <p>Una campagna attiva è una campagna che non è stata completata o non è andata a buon fine. Lo stato delle campagne attive può essere SCHEDULED , EXECUTING o PENDING_N EXT_RUN .</p>	
Dimensione massima del segmento	<p>Per i segmenti importati: 100.000.000 per campagna</p> <p>Per segmenti dinamici: illimitato</p>	No

Risorsa	Quota predefinita	Idoneità all'incremento
Campagne basate su eventi	<p>Ogni progetto può includere fino a 25 campagne che vengono inviate quando si verificano determinati eventi.</p> <p>Le campagne che utilizzano trigger basati su eventi devono utilizzare segmenti dinamici. Non possono utilizzare segmenti importati.</p> <p>Se integri la tua app con Amazon Pinpoint utilizzando un SDK per AWS dispositivi mobili, i messaggi delle campagne basate sugli eventi vengono inviati solo ai clienti le cui app eseguono la AWS Mobile SDK for Android versione 2.7.2 o successiva oppure la versione 2.6.30 o successiva. AWS Mobile SDK for iOS</p> <p>Se non riesce a recapitare un messaggio da una campagna basata su eventi entro cinque minuti, Amazon Pinpoint elimina il messaggio e non esegue altri tentativi di consegna.</p>	No

## Quote per l'e-mail

Le quote nelle sezioni seguenti si applicano al canale e-mail.

## Quote per i messaggi e-mail

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensione massima dei messaggi, compresi gli allegati	10 MB per messaggio	No
Numero di identità verificate	10.000 identità	No

 **Note**

Per identità si intendono indirizzi e-mail o domini o una qualsiasi combinazione di questi. Ogni messaggio e-mail inviato tramite Amazon Pinpoint deve essere inviato da un'identità verificata.

## Quote per mittenti e destinatari di e-mail

Risorsa	Quota predefinita	Idoneità all'incremento
Indirizzo del mittente	Tutti gli indirizzi o domini di invio devono essere verificati.	No
Indirizzo del destinatario	Se l'account si trova nella sandbox, tutti gli indirizzi e-mail o domini dei destinatari devono essere verificati.	<u>Si</u>

Risorsa	Quota predefinita	Idoneità all'incremento
	Se l'account è esterno alla sandbox, puoi inviare a qualsiasi indirizzo valido.	
Numero di destinatari per messaggio	50 destinatari per messaggio	No
Numero di identità che è possibile verificare	10.000 identità per regione AWS  <div data-bbox="592 640 1031 1239" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>Per identità si intendono indirizzi e-mail o domini o una qualsiasi combinazione di questi. Ogni messaggio e-mail inviato tramite Amazon Pinpoint deve essere inviato da un'identità verificata.</p> </div>	No

## Quote di invio e-mail

La quota di invio, la velocità di invio e i limiti dell'ambiente di sperimentazione (sandbox) sono condivisi tra i due servizi nella stessa regione. Se utilizzi Amazon SES in us-east-1, sei stato rimosso dall'ambiente di sperimentazione (sandbox) e hai subito un aumento della quota/tariffa di invio, tutte queste modifiche si applicano al tuo account Pinpoint in us-east-1.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero di e-mail che è possibile inviare in un periodo	Se l'account si trova nella sandbox, 200 e-mail nell'arco di 24 ore.	<a href="#">Sì</a>

Risorsa	Quota predefinita	Idoneità all'incremento
<p>di tempo di 24 ore (quota di invio)</p>	<p>Se l'account è esterno alla sandbox, la quota varia in base al caso d'uso specifico.</p> <div data-bbox="591 382 1029 936" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Questa quota si basa sul numero di destinatari, anziché sul numero di messaggi univoci inviati. Per destinatario si intende qualsiasi indirizzo e-mail presente nella riga A:.</p> </div>	
<p>Numero di e-mail che è possibile inviare al secondo (frequenza di invio)</p>	<p>Se l'account si trova nella sandbox, 1 e-mail al secondo.</p> <p>Se l'account è esterno alla sandbox, la frequenza varia in base al caso d'uso specifico.</p> <div data-bbox="591 1306 1029 1860" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Questa frequenza si basa sul numero di destinatari, anziché sul numero di messaggi univoci inviati. Per destinatario si intende qualsiasi indirizzo e-mail presente nella riga A:.</p> </div>	<p><u>Si</u></p>

## Quote per gli endpoint

Le quote seguenti si applicano alla risorsa [Endpoint](#) dell'API Amazon Pinpoint.

Il numero massimo di attributi supportati per endpoint è 250 e la dimensione massima dell'endpoint è 15 KB. Tuttavia, è possibile che questo numero di attributi sia limitato dalla dimensione totale di un endpoint, che include tutti gli attributi. Se riscontri errori durante l'aggiunta di attributi al modello, valuta la possibilità di ridurre la quantità di dati in ogni attributo o di diminuire il numero di attributi.

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensioni dell'endpoint	Dimensione massima 15 KB	No
Attributi assegnati collettivamente ai parametri <code>Attributes</code> , <code>Metrics</code> e <code>UserAttributes</code>	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro <code>Attributes</code>	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro <code>Metrics</code>	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro <code>UserAttributes</code>	250 per tutti i parametri degli attributi per applicazione	No
Lunghezza del nome degli attributi	50 caratteri	No
Lunghezza del valore degli attributi	100 caratteri	No
Oggetti <code>EndpointBatchItem</code> in un payload <code>EndpointBatchRequest</code>	100 per payload. La dimensione del payload non può essere superiore a 7 MB.	No

Risorsa	Quota predefinita	Idoneità all'incremento
Endpoint con lo stesso ID utente	15 endpoint univoci per ID utente	No
Valori assegnati agli attributi del parametro <code>Attributes</code>	50 per attributo	No
Valori assegnati agli attributi del parametro <code>UserAttributes</code>	50 per attributo	No

## Quote per l'importazione degli endpoint

Le quote seguenti si applicano all'importazione di endpoint in Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Processi di importazione attivi	10 per account  I processi di importazione vengono conteggiati rispetto a questa quota solo se sono in esecuzione. Una volta completato, il processo di importazione non viene più conteggiato rispetto a questa quota.	No
Dimensione di importazione	1 GB per processo di importazione  Ad esempio, se ogni endpoint è da 4 KB o meno, è possibile importare 250.000 endpoint.	No

## Quote per l'inserimento di eventi

Le seguenti quote si applicano all'acquisizione di eventi utilizzando gli SDK AWS mobili e la risorsa [Events](#) dell'API Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di tipi di evento personalizzati	1.500 per app	No
Numero massimo di chiavi di attributi personalizzati	500 per app	No
Numero massimo di valori di attributi personalizzati per chiave di attributo	100.000. Qualsiasi numero superiore a 100.000 è ancora registrato, ma non sarà disponibile nella console di analisi Amazon Pinpoint.	No
Numero massimo di caratteri per chiave di attributo	50	No
Numero massimo di caratteri per valore di attributo	200. Se il numero di caratteri supera i 200, l'evento viene eliminato.	No
Numero massimo di chiavi di parametri personalizzati	500 per app	No
Il numero massimo di eventi in una richiesta	100 per richiesta	No
Dimensione massima di una richiesta	4 MB	No
Dimensione massima di un singolo evento	1.000 KB	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di chiavi di attributo e chiavi di parametro per ogni evento	40 per richiesta	No

## Quote per i viaggi

Le quote seguenti si applicano ai viaggi.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta [Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas](#).

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di viaggi attivi	50 per account	No
Numero massimo di attivi EventTriggeredJourneys	20 per account	No
Numero massimo di attività di viaggio	40 per viaggio	No
Dimensione massima del segmento	Per i segmenti importati: 100.000.000 GB per percorso  Per segmenti dinamici: illimitato	No
Numero massimo di attività del contact center	3 per percorso	No
Regole relative al numero massimo di giorni di chiusura	20 per canale	No

Risorsa	Quota predefinita	Idoneità all'incremento
Lunghezza massima del nome della regola relativa ai giorni di chiusura	150 caratteri	No
Numero massimo di giorni tra l'ora di inizio e di fine per una regola dei giorni di chiusura	7 giorni	No
Numero massimo di regole relative agli orari di apertura	4 al giorno	No

## Quote di Lambda

Le quote seguenti si applicano alle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati da Lambda.

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensione massima di un payload di richiamo (richiesta e risposta) per una funzione Lambda	6 MB	No
Quantità massima di tempo da attendere per l'elaborazione dei dati da parte di una funzione Lambda	15 secondi	No
Numero massimo di attributi di evento per endpoint	5	No
Numero massimo di caratteri per un nome di attributo di evento	128 caratteri	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di caratteri per valore di attributo di evento	128 caratteri	No
Numero massimo di giorni di durata di esecuzione di un percorso	540 giorni	No

## Quote di machine learning

Le quote seguenti si applicano alle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati dai modelli di machine learning (ML).

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di configurazioni del modello	1 per modello di messaggio 100 per account	No
Numero massimo di suggerimenti	5 per endpoint o utente	No
Numero massimo di attributi suggeriti per endpoint o utente	1, se i valori degli attributi non vengono elaborati da una funzione AWS Lambda 10, se i valori degli attributi vengono elaborati da una funzione AWS Lambda	No
Lunghezza massima del nome di un attributo suggerito	50 caratteri per un nome di attributo 25 caratteri per il nome visualizzato di un attributo (il nome visualizzato in Attribute)	No

Risorsa	Quota predefinita	Idoneità all'incremento
	finder (Finder attributi) sulla console)	
Lunghezza massima di un valore di attributo consigliato recuperato da Amazon Personalize	100 caratteri	No
Dimensione massima di un payload di richiamo (richiesta e risposta) per una funzione Lambda	6 MB	No
Quantità massima di tempo da attendere per l'elaborazione dei dati da parte di una funzione Lambda	15 secondi	No
Numero massimo di tentativi di richiamo una funzione Lambda	3 tentativi	No

A seconda di come configuri Amazon Pinpoint per l'utilizzo di un modello ML, è possibile che vengano applicate quote aggiuntive. Per ulteriori informazioni sulle quote di Amazon Personalize, consulta [Quote](#) nella Guida per gli sviluppatori di Amazon Personalize. Per informazioni sulle quote AWS Lambda, consulta [Quote](#) nella Guida per gli sviluppatori AWS Lambda.

## Quote per i modelli di messaggio

Le quote seguenti si applicano ai modelli di messaggio per l'account Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di modelli di messaggio	20.000 per account	No
Numero massimo di versioni	5.000 per modello	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di caratteri in un modello e-mail	600.000 caratteri	No
Numero massimo di caratteri in un modello in-app	200.000 caratteri	No
Numero massimo di caratteri nelle parti del modello di default di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di ADM di un modello di notifica push	6.000 caratteri	No
Numero massimo di caratteri nelle parti di modello specifich e del servizio APN di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di Baidu di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di FCM di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri in un modello SMS	1.600 caratteri	No
Numero massimo di caratteri in un modello vocale	10,000 caratteri	No

## Quote per le notifiche push

Le quote seguenti si applicano ai messaggi inviati da Amazon Pinpoint tramite i canali di notifiche push.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di notifiche push che possono essere inviate al secondo in una campagna	25.000 notifiche al secondo	<a href="#">Sì</a>
Dimensione del payload dei messaggi di Amazon Device Messaging (ADM)	6 KB per messaggio	No
Dimensione del payload dei messaggi del servizio di notifiche push Apple (APN)	4 KB per messaggio	No
Dimensione del payload dei messaggi nell'ambiente di sperimentazione (sandbox) per il servizio APN	4 KB per messaggio	No
Dimensione del payload dei messaggi di Baidu Cloud Push	4 KB per messaggio	No
Dimensione del payload dei messaggi Firebase Cloud Messaging (FCM)	4 KB per messaggio	No

## Quote per i messaggi in-app

La seguente quota si applica ai messaggi in-app gestiti mediante Amazon Pinpoint.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta [Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas](#).

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di chiamate all'API <code>GetInAppMessages</code> al secondo.	5.000 richieste al secondo	<a href="#">Sì</a>
Campagne di messaggistica in-app	Ogni progetto può includere fino a 25 campagne che utilizzano il canale di messaggistica in-app.	Sì, consulta <a href="#">Richiesta di un aumento di quota nella Guida per l'utente di Service Quotas</a> .

## Quote per i segmenti

Le quote seguenti si applicano alla risorsa [Segmenti](#) dell'API Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di dimensioni che è possibile utilizzare per creare un segmento	100 per segmento	No
Numero massimo di gruppi di segmenti per segmento	5	No
Numero massimo di segmenti di origine per segmento	5	No
Profondità massima dei segmenti di origine  Ad esempio, se un segmento ha un segmento di origine e anche quest'ultimo ha un	5	No

Risorsa	Quota predefinita	Idoneità all'incremento
segmento di origine, la catena di profondità non supera questo limite.		

## Quote per SMS

Le quote seguenti si applicano al canale SMS.

Per ulteriori informazioni sui costi degli SMS, consulta [Informazioni sui report di utilizzo e fatturazione di SMS per Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Soglia di spesa	1,00 USD per account	<a href="#">Sì</a> , ma i limiti di spesa variano in base alla regione. È necessario specificare le regioni in cui si richiede un aumento.
Numero di SMS che è possibile inviare al secondo (frequenza di invio)	Varia a seconda del paese di destinazione e del numero di telefono di origine. Per ulteriori informazioni, consulta la sezione relativa ai <a href="#">limiti di parti di messaggio al secondo</a> nella Guida per l'utente di Amazon Pinpoint.	<a href="#">Sì</a> , tuttavia, potrebbe essere necessario disporre di un numero di telefono che supporti una velocità di trasmissione effettiva più elevata. Se non sei sicuro del tipo di numero da utilizzare e, contatta AWS Support o contatta il tuo AWS Account Manager per ulteriori informazioni  Se utilizzi un ID mittente alfanumerico per inviare messaggi, è possibile aumentare la velocità di

Risorsa	Quota predefinita	Idoneità all'incremento
		trasmissione effettiva. Per scoprire se è disponibile un aumento della velocità di trasmissione effettiva per l'ID mittente, <a href="#">apri una richiesta relativa all'ID mittente</a> nella console del Centro di supporto. Nella richiesta, includi l'ID mittente esistente, il paese in cui utilizzi tale ID e la velocità di trasmissione effettiva che desideri richiedere.
Numero di messaggi SMS che possono essere inviati a un singolo destinatario ogni secondo	1 messaggio al secondo	No
Numero di argomenti Amazon SNS per SMS bidirezionale	100.000 per account	<a href="#">Sì</a>
Numero di parole chiave per SMS bidirezionale	30 parole chiave per numero	<a href="#">Sì</a>
Numero di numeri per messaggi SMS e vocali	25 per account e regione	<a href="#">Sì</a>
Numero di numeri di telefono dedicati	25 per account	<a href="#">Sì</a>
Numero di elenchi di opt-out  Nota: l'elenco di opt-out predefinito richiesto viene confrontato con questa quota.	25 per account	<a href="#">Sì</a>

Risorsa	Quota predefinita	Idoneità all'incremento
Numero di set di configurazioni	25 per account	<a href="#">Sì</a>
Numero di destinazioni di evento	5 per set di configurazione	No
Numero di numeri di telefono di destinazione verificati nell'ambiente di sperimentazione (sandbox) per SMS	10 per account	<a href="#">Sì</a>
Numero di pool di numeri di telefono	25 per account	<a href="#">Sì</a>
Numero di identità di origine associabili a un pool di numeri di telefono	100 per pool di numeri di telefono	<a href="#">Sì</a>

## Quote 10DLC

Le seguenti quote si applicano ai messaggi SMS inviati utilizzando numeri di telefono 10DLC. I numeri 10DLC possono essere utilizzati solo per inviare messaggi a destinatari negli Stati Uniti.

Risorsa	Quota predefinita	Idoneità all'incremento
Massimo 10 aziende DLC per Account AWS	25	<a href="#">Sì</a>
Numero massimo di campagne 10DLC per azienda 10DLC	10	<a href="#">Sì</a>
Numero massimo di numeri 10DLC per campagna 10DLC	49	No

## Quote per i messaggi vocali

Le quote seguenti si applicano al canale vocale.

### Note

Quando l'account viene rimosso dalla sandbox, è possibile ottenere automaticamente le quote massime indicate nella tabella riportata di seguito.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero di messaggi vocali che è possibile inviare nell'arco di 24 ore	Se l'account si trova nella sandbox: 20 messaggi	No
Numero di messaggi vocali che è possibile inviare a un unico destinatario nell'arco di 24 ore	5 messaggi	No
Numero di messaggi vocali che è possibile inviare al minuto	Se l'account si trova nella sandbox: 5 chiamate al minuto Se l'account è esterno alla sandbox: 20 chiamate al minuto	No
Numero di messaggi vocali che è possibile inviare da uno singolo numero telefonico di origine al secondo	1 messaggio al secondo	No
Lunghezza di un messaggio vocale	Se l'account si trova nella sandbox: 30 secondi Se l'account è esterno alla sandbox: 5 minuti	No

Risorsa	Quota predefinita	Idoneità all'incremento
Possibilità di inviare messaggi vocali a numeri di telefono internazionali	<p>Se l'account si trova nella sandbox, puoi inviare messaggi solo ai destinatari nei seguenti paesi:</p> <ul style="list-style-type: none"><li>• Australia</li><li>• Canada</li><li>• Germania</li><li>• Hong Kong</li><li>• Israele</li><li>• Giappone</li><li>• Messico</li><li>• Singapore</li><li>• Svezia</li><li>• Stati Uniti</li><li>• Regno Unito</li></ul> <p>Se l'account è esterno alla sandbox, è possibile inviare messaggi a destinatari in qualsiasi paese.</p> <div data-bbox="594 1352 1029 1808"><p> <b>Note</b></p><p>Le chiamate internazionali comportano l'addebito di costi aggiuntivi, che variano in base al paese o alla regione di destinazione.</p></div>	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero di caratteri in un messaggio vocale	3.000 caratteri fatturabili, nelle parole pronunciate	No
	6.000 caratteri totali, inclusi i caratteri fatturabili e i tag SSML	
Numero di set di configurazioni	10.000 set di configurazione vocale per AWS regione	No

## Richiesta di aumento delle quote

Se il valore nella colonna Eligible for Increase (Possibilità di incremento) in una delle tabelle riportate sopra è Yes (Sì), puoi richiedere un incremento di quella quota.

### Richiesta di un aumento delle quote

1. Accedi all' AWS Management Console indirizzo <https://console.aws.amazon.com/>.
2. Crea un nuovo caso di AWS supporto all'indirizzo <https://console.aws.amazon.com/support/home#/case/create>.
3. Nella scheda I tuoi casi di supporto, scegli Create caso.
4. Scegli Stai cercando di aumentare i limiti di servizio?
5. In Aumento della quota di servizio, per Assistenza, scegli una delle seguenti opzioni:
  - Per richiedere un aumento della quota relativa al canale e-mail, scegliere Pinpoint Email (E-mail Pinpoint).
  - Per richiedere un aumento della quota per i limiti di spesa o le tariffe di invio di SMS, scegli SMS Pinpoint. Per tutti gli altri aumenti di quote per SMS, scegli Pinpoint
  - Per richiedere un aumento della quota relativa al canale vocale, scegliere Pinpoint Voice (Voce Pinpoint).
  - Per richiedere un aumento della quota relativa a qualsiasi altra funzionalità di Amazon Pinpoint, scegli Pinpoint.
6. A seconda del servizio scelto, potrebbe esserti chiesto di inserire quanto segue:

- (Facoltativo) In Fornisci un link al sito o all'app che invierà messaggi SMS, fornisci informazioni relative al sito Web, all'applicazione o al servizio che invierà messaggi SMS.
  - (Facoltativo) In Tipo di messaggi di cui è previsto l'invio, scegli il tipo di messaggio che intendi inviare con codici lunghi:
    - One-Time Password (Password una tantum) - Messaggi che forniscono password che i clienti utilizzano per l'autenticazione a un sito o un'applicazione.
    - Promotional (Promozionale) - Messaggi non critici che promuovono l'azienda o un servizio, ad esempio offerte speciali o annunci.
    - Transactional (Transazionale) - Messaggi informativi importanti che supportano le transazioni con i clienti, come conferme d'ordine o avvisi dell'account. I messaggi transazionali non devono contenere contenuti promozionali o di marketing.
  - (Facoltativo) Da quale AWS regione invierai i messaggi, scegli la regione da cui invierai i messaggi.
  - (Facoltativo) In Paesi a cui si prevede di inviare messaggi, immetti il paese o la regione in cui desideri acquistare codici brevi.
  - (Facoltativo) Nella sezione In che modo i clienti decidono di ricevere messaggi dall'utente, fornisci dettagli sul processo di consenso esplicito.
  - (Facoltativo) Nel campo Fornire il modello di messaggio che si intende utilizzare per inviare messaggi ai clienti, includi il modello che utilizzerai.
7. In Requests (Richieste), procedere come segue:
- Per la regione scegli la tua Regione AWS.
  - In Resource Type (Tipo di risorsa) scegliere General Limits (Limiti generali). Il campo Tipo di risorsa è presente solo per alcuni Servizi.
  - Per Quota scegli la quota da modificare.
  - In Nuovo valore della quota, inserisci un nuovo valore per la quota.
  - Per richiedere un aumento della stessa quota in un'altra Regione AWS, scegli Aggiungi un'altra richiesta, quindi scegli quella aggiuntiva Regione AWS e compila la nuova richiesta.
8. Scegliere la quota che si vuole aumentare, quindi immettere il nuovo valore desiderato per la quota.
9. Nella sezione Descrizione del caso, spiega perché richiedi l'aumento della quota.
10. In Opzioni di contatto, per Lingua di contatto preferita, scegli la lingua che preferisci usare per comunicare con il team di AWS supporto.

11. Per Metodo di contatto, scegli il metodo di comunicazione preferito con il team di AWS Supporto.
12. Scegli Invia.

Il team di AWS Support fornisce una prima risposta alla tua richiesta entro 24 ore.

Per evitare che i nostri sistemi vengano utilizzati per l'invio di contenuti indesiderati o dannosi, ogni richiesta dovrà essere analizzata attentamente da parte nostra. In seguito a questa valutazione, saremo in grado di gestire la tua richiesta durante le prime 24 ore. Tuttavia, se la risoluzione richiede l'invio di ulteriori informazioni da parte tua, i tempi di gestione della richiesta potranno essere più lunghi.

Potremmo non essere in grado di soddisfare la tua richiesta qualora il tuo caso d'uso non fosse allineato alle nostre policy.

# Cronologia dei documenti per Amazon Pinpoint

La tabella seguente descrive le modifiche importanti introdotte in ogni rilascio della Guida per gli sviluppatori di Amazon Pinpoint dopo dicembre 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

- Ultimo aggiornamento della documentazione: 16 novembre 2023

Modifica	Descrizione	Data
<a href="#">Intestazioni e-mail</a>	Puoi aggiungere intestazioni di posta elettronica ai tuoi messaggi di posta elettronica. Per ulteriori informazioni, consulta <a href="#">Invio di un'e-mail con intestazioni di annullamento dell'iscrizione</a> .	7 maggio 2024
<a href="#">Orchestrazione della posta elettronica</a>	Amazon Pinpoint ha aggiornato il modo in cui utilizza le risorse Amazon SES per inviare e-mail. Per ulteriori informazioni, consulta il <a href="#">ruolo IAM per l'invio di e-mail con Amazon SES</a> .	30 aprile 2024
<a href="#">Amazon Pinpoint ha aggiornato o la documentazione relativa alla guida per l'utente</a>	Gli argomenti relativi alla gestione delle risorse SMS e vocali vengono ora reindirizzati alla Amazon Pinpoint SMS User Guide. Per ulteriori informazioni, consulta la Guida per l' <a href="#">utente di Amazon Pinpoint SMS</a> .	8 febbraio 2024
<a href="#">Quote di Amazon Pinpoint</a>	Sono state aggiunte quote per le regole relative al numero	19 dicembre 2023

massimo di giorni di chiusura, alla lunghezza massima del nome della regola dei giorni di chiusura, al numero massimo di giorni tra l'ora di inizio e di fine per una regola dei giorni di chiusura e al numero massimo di regole sugli orari di apertura. Per ulteriori informazioni, consulta [Quote di Amazon Pinpoint](#).

[Amazon Pinpoint ha aggiornato la documentazione della guida per l'utente](#)

Per ottenere le informazioni più recenti su come creare, configurare e gestire le risorse vocali e SMS di Amazon Pinpoint, consulta la nuova [Guida per l'utente per Amazon Pinpoint SMS](#).

16 novembre 2023

[Quote di Amazon Pinpoint](#)

Sono state aggiornate le quote per UpdateEndpointsBatch, UpdateEndpoint, PutEvents DeleteEndpoint, e. GetEndpoint Per ulteriori informazioni, consulta [Quote di Amazon Pinpoint](#).

22 settembre 2023

### [Quote di Amazon Pinpoint](#)

Sono state aggiornate le quote per CreateEmailTemplate e CreateSmsTemplate, CreatePushTemplate, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate, UpdatePushTemplate, UpdateInAppTemplate, UpdateVoiceTemplate e. CreateImportJob Per ulteriori informazioni, consulta [Quote di Amazon Pinpoint](#).

12 settembre 2023

### [Metriche relative all'esecuzione di percorsi e campagne](#)

Sono state aggiunte nuove metriche di analisi per percorsi e campagne. Per ulteriori informazioni, consulta [Metriche relative all'esecuzione di percorsi e campagne](#).

25 aprile 2023

### [Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint](#)

Amazon Pinpoint ora supporta gli endpoint VPC di interfaccia. Per ulteriori informazioni, consulta [Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint](#).

11 aprile 2023

### [Crittografia in transito](#)

A partire dal 22/03/2023 Amazon Pinpoint non supporterà più TLS 1.0 ma potrai comunque utilizzare TLS 1.2 o versione successiva. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#).

20 marzo 2023

---

<a href="#">Quote di Amazon Pinpoint</a>	È stata aggiornata la procedura per richiedere un aumento della quota per le campagne, i percorsi e i messaggi in-app. Per ulteriori informazioni, consulta <a href="#">Quote di Amazon Pinpoint</a> .	16 dicembre 2022
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint è ora disponibile nella seguente regione: Stati Uniti orientali (Ohio).	5 ottobre 2022
<a href="#">Aggiornamenti degli esempi di ruoli IAM</a>	Sono stati aggiornati diversi esempi di ruoli IAM in tutto il documento per allinearli meglio alle best practice della sicurezza.	27 maggio 2022
<a href="#">API SMS e Voce versione 2</a>	Amazon Pinpoint ora include un'API dedicata per l'invio di SMS e messaggi vocali. Questa API include nuove funzionalità, come set di configurazione, pool ed elenchi di opt-out, utili per i clienti che inviano SMS e messaggi vocali a livello transazionale. Per ulteriori informazioni, consulta <a href="#">Utilizzo dell'API SMS e Voce versione 2 di Amazon Pinpoint</a> .	1 aprile 2022

<a href="#">Creazione e convalida di codici OTP (One-Time Password)</a>	Amazon Pinpoint ora include una funzionalità che genera codici OTP (One-Time Password) e li invia agli utenti come messaggi SMS. Include anche un'API per la convalida dei codici OTP quando gli utenti li inseriscono nell'applicazione o nel sito. Per ulteriori informazioni, consulta <a href="#">Invio e convalida di codici OTP (One-Time Password)</a> .	26 novembre 2021
<a href="#">Messaggi in-app</a>	Sono state aggiunte informazioni sull'integrazione della funzionalità di <a href="#">messaggistica in-app</a> di Amazon Pinpoint con le app.	10 novembre 2021
<a href="#">Esempi di codice</a>	È stata aggiunta una <a href="#">libreria di esempi di codice</a> per le operazioni più comuni di Amazon Pinpoint.	3 novembre 2021
<a href="#">Quote per i progetti</a>	Il <a href="#">numero massimo di progetti Amazon Pinpoint</a> rimane a 100, ma questa quota può ora essere aumentata aprendo una richiesta di aumento del limite di servizio con. AWS Support	11 ottobre 2021

<a href="#">Aggiornamenti delle policy Lambda</a>	Alcune policy di autorizzazione Lambda devono ora includere una condizione AWS:SourceAccount . Sono state aggiornate le policy di esempio negli argomenti <a href="#">Creazione di canali personalizzati in Amazon Pinpoint</a> e <a href="#">Personalizzazione dei segmenti con AWS Lambda</a> per soddisfare questo requisito.	7 ottobre 2021
<a href="#">UpdateEndpoint</a>	L' <a href="#">UpdateEndpoint API</a> Amazon Pinpoint è ora registrata da CloudTrail	16 Novembre 2020
<a href="#">Attributi personalizzati</a>	Amazon Pinpoint ora supporta 250 attributi nei modelli di messaggistica e-mail. Consulta l'argomento <a href="#">Quote</a> .	18 settembre 2020
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint ora è disponibile nelle seguenti regioni: Asia Pacifico (Tokyo), Europa (Londra) e Canada (Centrale). L'API SMS e Voce di Amazon Pinpoint non è disponibile nelle seguenti regioni:	10 settembre 2020
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint ora è disponibile nella regione Asia Pacifico (Tokyo). Tieni presente che l'API SMS e Voce di Amazon Pinpoint non supporta la funzionalità vocale in questa regione.	2 settembre 2020

<a href="#">Eventi delle campagne</a>	Sono state aggiunte informazioni su un nuovo parametro <code>delivery_type</code> degli <a href="#">eventi della campagna</a> .	2 agosto 2020
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint ora è disponibile nella regione Asia Pacifico (Seoul). Tieni presente che l'API Amazon Pinpoint non supporta chiamate vocali o SMS in questa regione.	31 luglio 2020
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint è ora disponibile nella AWS GovCloud (US) regione.	30 aprile 2020
<a href="#">Canali personalizzati</a>	Sono state aggiornate le Informazioni sulla <a href="#">creazione di canali personalizzati mediante funzioni Lambda o webhook</a> .	23 aprile 2020
<a href="#">Machine learning</a>	Sono state aggiunte informazioni sul recupero di consigli personalizzati dai modelli di raccomandazione e, facoltativamente, sul <a href="#">miglioramento di tali</a> consigli utilizzando le funzioni. AWS Lambda	4 marzo 2020
<a href="#">Sicurezza</a>	È stato aggiunto un <a href="#">capitolo sulla sicurezza</a> , che fornisce informazioni su vari controlli di sicurezza e funzionalità di Amazon Pinpoint.	4 febbraio 2020

---

<a href="#">Percorsi</a>	<p>Sono state aggiunte informazioni sull'utilizzo dei percorsi Amazon Pinpoint per sviluppare e flussi di lavoro automatizzati che eseguono attività di messaggistica per i progetti. Sono state inoltre aggiunte informazioni <a href="#">sull'interrogazione e dei dati analitici</a> per un sottoinsieme di parametri applicabili ai percorsi.</p>	31 ottobre 2019
<a href="#">Analisi</a>	<p>Aggiunte procedure che spiegano come eseguire <a href="#">query sui dati di analisi</a> per campagne e messaggi transazionali e aggiunte informazioni sull'<a href="#">utilizzo dei risultati delle query</a>.</p>	17 ottobre 2019
<a href="#">Analisi</a>	<p>Sono state aggiunte informazioni sull'<a href="#">esecuzione di query sui dati di analisi</a> per un sottoinsieme di parametri applicabili ai messaggi SMS e alle e-mail transazionali.</p>	6 settembre 2019
<a href="#">Esempi di codice</a>	<p>Sono stati aggiunti <a href="#">esempi di codice</a> che possono essere utilizzati per inviare notifiche push transazionali utilizzando tutti i servizi supportati da Amazon Pinpoint.</p>	30 luglio 2019

---

<a href="#">Analisi</a>	Aggiunte informazioni sull' <a href="#">esecuzione di query per le attività di analisi dei dati</a> per un sottoinsieme di parametri che si applicano a progetti (applicazioni) e campagne.	24 luglio 2019
<a href="#">Segmenti</a>	È stato aggiunto un <a href="#">tutorial</a> che descrive una soluzione per l'importazione dei dati dei clienti in Amazon Pinpoint da sistemi esterni, ad esempio Salesforce o Marketo.	14 maggio 2019
<a href="#">Disponibilità regionale</a>	Amazon Pinpoint è ora disponibile nelle regioni AWS Asia Pacifico (Mumbai) e Asia Pacifico (Sydney).	25 aprile 2019
<a href="#">Utilizzo di postman con Amazon Pinpoint</a>	È stato aggiunto un <a href="#">tutorial</a> che descrive come utilizzare Postman per l'interazione con l'API Amazon Pinpoint.	8 Aprile 2019
<a href="#">Assegnazione di tag</a>	Sono state aggiunte informazioni sull' <a href="#">assegnazione di tag alle risorse Amazon Pinpoint</a> .	27 febbraio 2019
<a href="#">Registrazione SMS</a>	Aggiunta un <a href="#">capitolo di tutorial</a> e aggiunto un tutorial che descrive come creare <a href="#">una soluzione che gestisce la registrazione degli utenti tramite SMS</a> .	27 febbraio 2019

[Esempi di codice](#)

Aggiunti [esempi di codice](#) in vari linguaggi di programmazione che illustrano come inviare messaggi [e-mail](#), [SMS](#) e [vocali](#) a livello di programmazione.

6 febbraio 2019

## Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti introdotte in ogni rilascio della Guida per gli sviluppatori di Amazon Pinpoint prima di dicembre 2018.

Modifica	Descrizione	Data
Disponibilità regionale	Amazon Pinpoint è ora disponibile nelle regioni AWS Stati Uniti occidentali (Oregon) ed Europa (Francoforte).	21 dicembre 2018
Canale vocale	È possibile utilizzare il nuovo canale vocale Amazon Pinpoint per creare messaggi vocali e recapitarli ai clienti tramite telefono. Attualmente, è possibile inviare messaggi vocali solo utilizzando l'API API SMS e Voce di Amazon Pinpoint.	15 novembre 2018
Disponibilità in Europa (Irlanda)	Amazon Pinpoint è ora disponibile nella regione AWS Europa (Irlanda).	25 ottobre 2018
API Events	È possibile utilizzare l'API Amazon Pinpoint per <a href="#">registrare</a>	7 agosto 2018

Modifica	Descrizione	Data
	<a href="#">e eventi</a> e associarli agli endpoint.	
Esempi di codice per la definizione e la ricerca di endpoint	Sono stati aggiunti esempi di codice come illustrano come definire, aggiornare, eliminare e cercare endpoint. Vengono forniti esempi per AWS CLI AWS SDK for Java, e l'API Amazon Pinpoint. Per ulteriori informazioni, consulta <a href="#">Definizione del pubblico in Amazon Pinpoint</a> e <a href="#">Accesso ai dati del pubblico in Amazon Pinpoint</a> .	7 agosto 2018
Autorizzazioni di esportazione degli endpoint	<a href="#">Configura una policy IAM</a> che consente di esportare gli endpoint Amazon Pinpoint in un bucket Amazon S3.	1 maggio 2018
Verifica dei numeri di telefono per SMS	È possibile utilizzare l'API Amazon Pinpoint per <a href="#">verificare un numero di telefono</a> al fine di determinare se è una destinazione valida per i messaggi SMS.	23 aprile 2018
Aggiornamento degli argomenti per l'integrazione di Amazon Pinpoint	<a href="#">Integra Amazon Pinpoint</a> con il tuo Android, iOS o JavaScript un'applicazione utilizzando AWS SDK o librerie.	23 marzo 2018

Modifica	Descrizione	Data
AWS CloudTrail registrazione	Sono state aggiunte informazioni sulla <a href="#">registrazione delle chiamate API Amazon Pinpoint</a> con. CloudTrail	6 febbraio 2018
Service Quotas aggiornate	<a href="#">Quote</a> aggiornate con ulteriori informazioni sulle quote per l'e-mail.	19 gennaio 2018
Beta pubblica per le estensioni Amazon Pinpoint	Usa AWS Lambda le funzioni per <a href="#">personalizzare i segmenti</a> o <a href="#">creare canali di messaggistica personalizzati</a> .	28 novembre 2017
Quote del payload delle notifiche push	Le quote includono le <a href="#">dimensioni del payload per i messaggi push ai dispositivi mobili</a> .	25 ottobre 2017
Service Quotas aggiornate	Sono state aggiunte informazioni sui canali SMS ed e-mail in <a href="#">Quote</a> .	9 Ottobre 2017
Push per dispositivi mobili ADM e Baidu	È possibile aggiornare il codice dell'app per gestire le notifiche push dai canali push per dispositivi mobili Baidu e ADM.	27 settembre 2017

Modifica	Descrizione	Data
ID utente ed eventi di autenticazione con pool di utenti di Amazon Cognito.	Se si usano i pool di utenti di Amazon Cognito per gestire gli accessi degli utenti nelle app per dispositivi mobili, Amazon Cognito assegna gli ID utente agli endpoint e segnala gli eventi di autenticazione a Amazon Pinpoint.	26 settembre 2017
ID utente	È possibile assegnare ID utente agli endpoint per monitorare l'utilizzo delle app da parte di singoli utenti. Sono disponibili esempi per <a href="#">SDK mobile AWS</a> e <a href="#">SDK per Java</a> .	31 agosto 2017
Eventi di autenticazione	Il reporting degli eventi di autenticazione consente di scoprire la frequenza con cui gli utenti eseguono l'autenticazione nell'app. Alcuni esempi sono disponibili in <a href="#">Segnalazione degli eventi nell'applicazione</a> .	31 agosto 2017
Eventi di esempio aggiornati	Gli <a href="#">eventi di esempio</a> includono eventi che Amazon Pinpoint invia in streaming per le attività di e-mail e SMS.	08 giugno 2017
Gestione delle sessioni Android	È possibile gestire le sessioni nelle app Android utilizzando una classe fornita dall'app AWS Mobile Hub di esempio.	20 aprile 2017

Modifica	Descrizione	Data
Esempi di eventi di monetizzazione aggiornati	Il codice di esempio è stato aggiornato per il reporting degli eventi di monetizzazione. .	31 marzo 2017
Flussi di eventi	È possibile configurare Amazon Pinpoint per <a href="#">inviare gli eventi relativi ad app e campagne a un flusso Kinesis</a> .	24 marzo 2017
Autorizzazioni	<a href="#">Come Amazon Pinpoint funziona con IAM</a> Per informazioni su come concedere l'accesso ad Amazon Pinpoint agli utenti del tuo account e agli utenti della tua app AWS per dispositivi mobili, consulta la pagina.	12 gennaio 2017
Disponibilità generale di Amazon Pinpoint	Questo rilascio introduce Amazon Pinpoint.	1° dicembre 2016

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.