



Entwicklerhandbuch

Amazon Location Service



Amazon Location Service: Entwicklerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Willkommen	1
Was ist Amazon Location Service?	1
Schlüsselfeatures	2
Zugehörige Services	3
Schnellstart	5
Erstellen einer Webanwendung	5
Erstellen von -Ressourcen	6
Einrichtung der Authentifizierung	7
Erstellen von HTML	9
Hinzufügen der Karte	12
Hinzufügen der Suche	16
Endgültige Anwendung	21
Was ist als Nächstes?	26
Eine Android-App erstellen	26
Amazon-Standortressourcen für Ihre App erstellen	27
Authentifizierung einrichten	29
Erstellen Sie die App	32
Fügen Sie die Karte hinzu	33
Suche hinzufügen	36
Tracking hinzufügen	46
Was kommt als Nächstes	56
Eine iOS-App erstellen	56
Erstellen von -Ressourcen	56
Authentifizierung einrichten	58
Die App wird erstellt	61
Ursprünglicher Code	62
Fügen Sie eine Karte hinzu	65
Suche hinzufügen	69
Tracking hinzufügen	71
Was kommt als Nächstes	83
Konzepte Amazon Amazon-Standorte	84
Übersicht	85
Zuordnungen	86
Kartenstile	87

Politische Ansichten	87
Benutzerspezifische Layers	88
Rendern von Karten	88
Terminologie der Karten	90
Suche nach Orten	91
Konzepte zur Geokodierung	93
Suchergebnisse	93
Vielfältige Ergebnisse und Relevanz	94
Ergebnisse adressieren	95
Geokodierungsergebnisse speichern	97
Terminologie von Places	98
Routen	99
Ressourcen für den Routenrechner	99
Berechnung einer Route	100
Routen planen	101
Terminologie der Route	102
Geofences und Tracker	104
Geofences	104
Tracker	106
Geofence-Terminologie	111
Tracker-Terminologie	112
Häufige Anwendungsfälle	113
Anwendungen für Benutzerinteraktion und Geomarketing	114
Anwendungen zur Nachverfolgung von Vermögenswerten	116
Bereitstellungsanwendungen	117
Datenanbieter	119
Reichweite und Funktionen des Datenanbieters	120
Kartenstile	121
Weitere Details	121
Esri	121
GrabMaps	130
HERE Technologies	136
Daten öffnen	144
Funktionen nach Datenanbieter	153
Nutzungsbedingungen und Datenzuweisung	158
Regionen und Endpunkte	158

Regionen	159
Endpunkte	160
Endpunkte für den API-Betrieb	161
Servicekontingente	163
Verwaltung Ihrer Amazon Location Service-Kontingente	180
Entwicklung mit Amazon Location	182
Szenarien und Anwendungsfälle	182
SDKs und Tools	184
SDKs nach Sprache	184
MapLibre	188
Amazon-Standort-SDK	193
Amazon-Standort-APIs	219
Amazon Location mit einem AWS SDK verwenden	219
Aktualisierungen von Fehlermeldungen	219
Codebeispiele	251
Demo-Website für Amazon-Standorte	253
Tutorial: Schnellstart	253
Tutorial: Datenbankanreicherung	254
Beispiel: Explore App	255
Beispiel: Gestalte eine Karte	256
Beispiel: Markierungen zeichnen	256
Beispiel: Zeichnet geclusterte Punkte	257
Beispiel: Zeichne ein Polygon	257
Beispiel: Ändern Sie die Kartensprache	258
Blog: Benachrichtigungen zur voraussichtlichen Lieferzeit	259
Beispiel: Aktualisierungen der Stream-Position	260
Beispiel: Mobilanwendung für Geofencing und Tracking	260
So verwenden Sie Amazon Location	262
Voraussetzungen für das Konto	263
Melden Sie sich an für ein AWS-Konto	263
Erstellen Sie einen Benutzer mit Administratorzugriff	264
Zugriff auf Amazon Location Service gewähren	265
Verwenden von Karten	267
Voraussetzungen	268
Anzeigen von Karten	271
Zeichnen auf einer Karte	327

Einstellungserweiterungen für eine Zuordnung	328
Verwalten von Kartenressourcen	329
Ortssuche	333
Voraussetzungen	334
Geokodierung	338
Geokodierung umkehren	345
AutoVervollständigen	349
Verwenden von Orts-IDs	356
Kategorien und Filterung	358
Tutorial: Datenbankanreicherung	363
Verwalten von Ortsindexressourcen	378
Berechnen von Routen	382
Voraussetzungen	383
Berechnen der Route	386
Routenplanung	390
Positionen, die sich nicht auf einer Straßen befinden	397
Zeit für die Warteschlange	399
Reisemodus	400
Verwalten von Routenressourcen	402
Geofencing und Tracking	406
Schritt 1: Geofences hinzufügen	407
Schritt 2: Starten Sie das Tracking	415
Schritt 3: Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung	430
Schritt 4: Vergleichen Sie die Gerätepositionen mit Geofences	431
Überprüfen Sie die Gerätepositionen	435
Reagieren auf Ereignisse mit EventBridge	437
Nachverfolgung mit AWS IoT und MQTT	443
Verwalten von Geofence-Ressourcen	451
Verwalten von Tracker-Ressourcen	459
Beispiel für eine mobile Geofencing- und Tracking-Anwendung	463
Markieren Ihrer -Ressourcen	483
Einschränkungen	484
Gewährt die Berechtigung zum Markieren	484
Hinzufügen eines Tags zu einer Ressource	485
Nachverfolgen von Kosten nach Tag	486
Kontrollieren des Zugriffs auf -Ressourcen mithilfe von Tags	487

Weitere Informationen	487
Gewähren des Zugriffs auf Amazon Location	487
Verwenden von API-Schlüsseln	488
Verwenden von Amazon Cognito	496
Überwachen von Amazon Location Service	507
Überwachung mit CloudWatch	507
Verwenden von CloudTrail mit Amazon Location	513
Verwenden von AWS CloudFormation zum Erstellen von Ressourcen	517
Amazon Location und AWS CloudFormation Vorlagen	518
Weitere Informationen zu AWS CloudFormation	518
Sicherheit	519
Datenschutz	520
Datenschutz	521
Datenaufbewahrung	521
Verschlüsselung von Daten im Ruhezustand	521
Verschlüsselung von Daten bei der Übertragung	535
Identitäts- und Zugriffsverwaltung	535
Zielgruppe	536
Authentifizierung mit Identitäten	537
Verwalten des Zugriffs mit Richtlinien	541
So funktioniert Amazon Location Service mit IAM	543
So funktioniert Amazon Location Service mit nicht authentifizierten Benutzern	552
Beispiele für identitätsbasierte Richtlinien	553
Fehlerbehebung	566
Vorfallreaktion	568
Protokollieren und Überwachen	568
Compliance-Validierung	569
Ausfallsicherheit	571
Sicherheit der Infrastruktur	571
Konfigurations- und Schwachstellenanalyse	572
Confused-Deputy-Prävention	572
Bewährte Methoden für die Gewährleistung der Sicherheit	572
Bewährte Methoden für Detective	573
Bewährte präventive Verfahren	573
Bewährte Methoden	574
Sicherheit	575

Ressourcenmanagement	575
Fakturierung und Kostenmanagement	576
Kontingente und Nutzung	576
Dokumentverlauf	578
AWS-Glossar	589
.....	dxc

Willkommen beim Amazon Location Service

Willkommen beim Entwicklerhandbuch für Amazon Location Service .

Die folgenden Themen können Ihnen bei den ersten Schritten in der -Dokumentation helfen, je nachdem, was Sie versuchen zu tun.

Einen Überblick über Amazon Location erhalten

- Erfahren Sie mehr über die [Konzepte in Amazon Location](#) .
- Verschaffen Sie sich einen tieferen Einblick in die Funktionalität im [So verwenden Sie Amazon Location Service](#) Kapitel.
- Siehe Demo-Apps auf der [Demo-Website von Amazon Location](#) .
- Wenn Sie bereits über ein verfügbares AWS-Konto, können Sie die [Amazon Location Service-Konsole](#) verwenden, um die Funktionalität aus erster Hand zu erkunden.

Amazon Location als Entwickler verwenden

- Erstellen Sie Ihre erste App mit der [Schnellstart](#).
- Erfahren Sie im [So verwenden Sie Amazon Location Service](#) Kapitel, wie die verschiedenen Funktionen von Amazon Location Service funktionieren.
- Weitere Informationen finden Sie in den SDKs und Tools, die Ihnen im [Entwicklung mit Amazon Location](#) Kapitel zur Verfügung stehen.
- Sehen Sie sich [Codebeispiele und Tutorials](#) an, die Sie in Ihren eigenen Apps verwenden können. Sie können auch die Seite mit den Demo-[Websitebeispielen](#) von Amazon Location besuchen, um Beispiele zu finden, die nach Feature, Sprache oder Plattform gefiltert werden können.
- Informationen zu Amazon Location APIs finden Sie im [API-Referenzhandbuch für](#) .

Was ist Amazon Location Service?

Mit Amazon Location Service können Sie Standortdaten und Funktionen zu Anwendungen hinzufügen, die Funktionen wie Karten, Interessenpunkte, Geokodierung, Routing, Geofences und Nachverfolgung umfassen. Amazon Location bietet standortbasierte Services (LBS) unter Verwendung hochwertiger Daten von globalen, vertrauenswürdigen Anbietern Esri, Grab und HERE. Mit kostengünstigen Daten, Nachverfolgungs- und Geofencing-Funktionen und integrierten Metriken für die Zustandsüberwachung können Sie anspruchsvolle standortfähige Anwendungen erstellen.

Mit Amazon Location behalten Sie die Kontrolle über die Daten Ihrer Organisation. Amazon Location anonymisiert alle an Datenanbieter gesendeten Abfragen, indem Kundenmetadaten und Kontoinformationen entfernt werden. Darüber hinaus verlassen vertrauliche Standortinformationen wie Anlagen-, Komponenten- und Personalstandorte Ihr AWS Konto überhaupt nicht. Auf diese Weise können Sie vertrauliche Informationen von Drittanbietern schützen, den Datenschutz von Benutzern schützen und die Sicherheitsrisiken Ihrer Anwendung reduzieren. Mit Amazon Location haben Amazon und Dritte keine Rechte, Ihre Daten zu verkaufen oder für Werbung zu verwenden.

Amazon Location ist vollständig in -Services wie AWS CloudTrail, Amazon EventBridge, CloudWatchAmazon und AWS Identity and Access Management (IAM) integriert. Amazon Location vereinfacht Ihren Entwicklungsworkflow mit Datenintegration und verfolgt Apps mit integrierten Überwachungs-, Sicherheits- und Compliance-Funktionen schnell in die Produktion.

Highlights, Produktdetails und Preise finden Sie auf der Serviceseite für [Amazon Location Service](#) .

Wichtige Funktionen in Amazon Location

Amazon Location bietet die folgenden Funktionen:

Zuordnungen

Mit Amazon Location Service Maps können Sie Standortinformationen visualisieren und sind die Grundlage für viele standortbasierte Service-Funktionen. Amazon Location Service bietet Kartenkacheln verschiedener Stile, die von globalen Standortdatenanbietern Esri, Grab und HERE stammen, sowie Open Data Maps.

Orte

Mit Amazon Location Service Places können Sie Suchfunktionen in Ihre Anwendung integrieren, Adressen in geografische Koordinaten mit Breiten- und Längengrad (Geokodierung) konvertieren und eine Koordinate in eine Straßenadresse (Geokodierung umkehren) konvertieren. Amazon Location Service bezieht qualitativ hochwertige Geodaten von Esri, Grab und HERE, um Places-Funktionen zu unterstützen.

Routing

Mit Amazon Location Service Routes können Sie Routen finden und die Reisezeit auf der Grundlage von up-to-date Straßen- und Live-Datenverkehrsinformationen schätzen. Erstellen Sie

Funktionen, mit denen Ihre Anwendung die Reisezeit, Entfernung und Richtungen zwischen zwei beliebigen Standorten anfordern kann. Berechnen Sie die Zeit und Entfernung für eine Matrix von Routen, die bei der Routenplanung verwendet werden sollen.

Geofencing

Mit Amazon Location Service Geofences können Sie Ihrer Anwendung die Möglichkeit geben, zu erkennen und zu handeln, wenn ein Gerät eine definierte geografische Grenze betritt oder verlässt, die als Geofence bezeichnet wird. Senden Sie automatisch ein Ein- oder Ausgangsereignis an Amazon EventBridge, wenn eine Geofence-Verstöße erkannt werden. Auf diese Weise können Sie nachgelagerte Aktionen initiieren, z. B. das Senden einer Benachrichtigung an ein Ziel.

Tracker

Mit Amazon Location Service Trackers können Sie den aktuellen und historischen Standort der Geräte abrufen, auf denen Ihre Anwendung mit aktivierter Nachverfolgung ausgeführt wird. Sie können Tracker auch mit Amazon Location Service Geofences verknüpfen, um Standortaktualisierungen von Ihren Geräten automatisch anhand Ihrer Geofences auszuwerten. Tracker können Ihnen helfen, Kosten zu senken, indem sie Positionsaktualisierungen filtern, die sich vor dem Speichern oder Auswerten anhand von Geofences nicht bewegt haben.

Wenn Sie Tracker verwenden, verlassen vertrauliche Standortinformationen auf Ihren verfolgten Geräten Ihr AWS Konto nicht. Dies trägt dazu bei, vertrauliche Informationen vor Dritten zu schützen, den Datenschutz von Benutzern zu schützen und Sicherheitsrisiken zu reduzieren.

Services, die Sie mit Amazon Location verwenden können

Verwenden Sie die folgenden Services zusammen mit Amazon Location Service.

Integrierte Überwachung und Verwaltung

Amazon Location Service ist für eine EventBridge effiziente Überwachung und Datenverwaltung in Amazon CloudWatchAWS CloudTrail, und integriert:

- Amazon CloudWatch – Zeigen Sie Metriken zur Servicenutzung und zum Servicestatus an, einschließlich Anfragen, Latenz, Fehlern und Protokollen. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

- AWS CloudTrail – Protokollieren und überwachen Sie Ihre API-Aufrufe, die Aktionen eines Benutzers, einer Rolle oder eines AWS-Services umfassen. Weitere Informationen finden Sie unter [the section called “Verwenden von CloudTrail mit Amazon Location”](#).
- Amazon EventBridge – Aktivieren Sie eine ereignisgesteuerte Anwendungsarchitektur, damit Sie -AWS LambdaFunktionen verwenden können, um andere Teile Ihrer Anwendung und Workflows zu aktivieren. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).

Entwickler-Tools

Amazon Location Service bietet eine Vielzahl von Tools, mit denen Entwickler standortfähige Anwendungen erstellen können. Dazu gehören die Standard-AWSSDKs , mobile und Web-SDKs sowie Beispielcode, um sie mit Open-Source-Bibliotheken wie zu kombinieren MapLibre. Verwenden Sie die [Amazon Location Service-Konsole](#), um mehr über -Ressourcen zu erfahren und mit einem visuellen und interaktiven Lerntool zu beginnen.

Schneller Start mit Amazon Location Service

Der effizienteste Weg, um mit Amazon Location Service zu beginnen, ist die Verwendung der [Amazon Location-Konsole](#). Auf der [Explore-Seite können Sie Ihre Ressourcen erstellen und verwalten und die](#) Amazon-Standortfunktion ausprobieren.

Note

Um die Amazon Location Service Service-Konsole zu verwenden oder den Rest dieses Tutorials zu befolgen, müssen Sie zunächst den Vorgang abschließen [Voraussetzungen für die Nutzung von Amazon Location Service](#), einschließlich der Erstellung eines AWS Kontos und der Gewährung des Zugriffs auf Amazon Location.

Um sich mit den Amazon Location APIs vertraut zu machen, verwenden Sie das folgende Tutorial, um eine einfache Anwendung zu erstellen, die eine interaktive Karte anzeigt und Suchfunktionen verwendet. Es gibt drei Versionen des Tutorials: Eine zeigt Ihnen, wie Sie mit Hilfe einer einfachen Webseite erstellen JavaScript, die zweite zeigt dasselbe für eine Android-Anwendung mit Kotlin und die dritte zeigt dasselbe für eine iOS-Anwendung mit Swift.

Themen

- [Erstellen einer Webanwendung](#)
- [Eine Android-App erstellen](#)
- [Eine iOS-App erstellen](#)

Erstellen einer Webanwendung

In diesem Abschnitt erstellen Sie eine statische Webseite mit einer Karte und der Möglichkeit, an einem Standort zu suchen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen und einen API-Schlüssel für Ihre Anwendung.

Themen

- [Erstellen von Amazon Location-Ressourcen für Ihre App](#)
- [Einrichten der Authentifizierung für Ihre Anwendung](#)
- [Erstellen des HTML-Codes für Ihre Anwendung](#)

- [Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung](#)
- [Hinzufügen einer Suche zu Ihrer Anwendung](#)
- [Anzeigen der endgültigen Anwendung](#)
- [Was ist als Nächstes?](#)

Erstellen von Amazon Location-Ressourcen für Ihre App

Wenn Sie sie noch nicht haben, müssen Sie die Amazon Location-Ressourcen erstellen, die Ihre Anwendung verwenden wird. Hier erstellen Sie eine Kartenressource, um Karten in Ihrer Anwendung anzuzeigen, und einen Ortsindex, um nach Speicherorten auf der Karte zu suchen.

So fügen Sie Ihrer Anwendung Standortressourcen hinzu


1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Name nund eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie eine Zuordnung aus.

Note

Wenn Sie einen Kartenstil auswählen, wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
- e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie verwenden sie später in diesem Tutorial, um die richtige Authentifizierung zu erstellen.

2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie einen Datenanbieter aus.

 Note


Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem bereits ausgewählten Kartenanbieter entspricht. Auf diese Weise können Sie sicherstellen, dass die Suchen mit den Karten übereinstimmen.

Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher aus. Für dieses Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie Nein wählen können, nur für den einmaligen Gebrauch.
- e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Ortsindex erstellen aus.
- f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie verwenden sie, um die richtige Authentifizierung im nächsten Abschnitt dieses Tutorials zu erstellen.

Einrichten der Authentifizierung für Ihre Anwendung


Die Anwendung, die Sie in diesem Tutorial erstellen, hat eine anonyme Nutzung, was bedeutet, dass sich Ihre Benutzer nicht bei anmelden müssen, um die Anwendung AWS zu verwenden. Standardmäßig erfordern die Amazon Location Service APIs jedoch eine Authentifizierung, um sie zu verwenden. Sie können entweder Amazon Cognito oder API-Schlüssel verwenden, um Authentifizierung und Autorisierung für anonyme Benutzer bereitzustellen. In diesem Tutorial erstellen Sie API-Schlüssel zur Verwendung in der Beispielanwendung.

 Note

Weitere Informationen zur Verwendung von API-Schlüsseln oder Amazon Cognito mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

So richten Sie die Authentifizierung für Ihre Anwendung ein

1. Gehen Sie zur [Amazon Location-Konsole](#) und wählen Sie im linken Menü API-Schlüssel aus.
2. Klicken Sie auf API-Schlüssel erstellen.

 Important

Der API-Schlüssel, den Sie erstellen, muss sich in demselben AWS-Konto und derselben AWS Region befinden wie die Amazon Location Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

3. Geben Sie auf der Seite API-Schlüssel erstellen die folgenden Informationen ein.
 - Name – Ein Name für Ihren API-Schlüssel, z. B. MyWebAppKey.
 - Ressourcen – Wählen Sie die Amazon Location Map- und Place-Indexressourcen aus, die Sie im vorherigen Abschnitt erstellt haben. Sie können mehr als eine Ressource hinzufügen, indem Sie Ressource hinzufügen auswählen. Auf diese Weise kann der API-Schlüssel mit diesen Ressourcen verwendet werden.
 - Aktionen – Geben Sie die Aktionen an, die Sie mit diesem API-Schlüssel autorisieren möchten. Sie müssen mindestens `geo:GetMap*` und `geo:SearchPlaceIndexForPosition` auswählen, damit das Tutorial wie erwartet funktioniert.
 - Sie können Ihrem API-Schlüssel optional eine Beschreibung, eine Ablaufzeit oder Tags hinzufügen. Sie können auch einen Referer (z. B. `*.example.com`) hinzufügen, um den Schlüssel auf die Verwendung von einer bestimmten Domain zu beschränken. Das bedeutet, dass das Tutorial nur von dieser Domain aus funktioniert.

Note

Es wird empfohlen, Ihre API-Schlüsselnutzung zu schützen, indem Sie entweder eine Ablaufzeit oder einen Referer festlegen, falls nicht beides der Fall ist.

4. Wählen Sie API-Schlüssel erstellen, um den API-Schlüssel zu erstellen.
5. Wählen Sie API-Schlüssel anzeigen und kopieren Sie den Schlüsselwert zur Verwendung später im Tutorial. Es wird das Format `habenv1.public.a1b2c3d4...`

Important

Sie benötigen diesen Schlüssel, wenn Sie den Code für Ihre Anwendung später in diesem Tutorial schreiben.

Erstellen des HTML-Codes für Ihre Anwendung

In diesem Tutorial erstellen Sie eine statische HTML-Seite, die eine Karte einbettet und es dem Benutzer ermöglicht, zu finden, was sich an einem Speicherort auf der Karte befindet. Die App besteht aus drei Dateien: einer HTML-Datei und einer CSS-Datei für die Webseite und einer JavaScript (.js)-Datei für den Code, der die Karte erstellt und auf die Interaktionen und Kartenereignisse des Benutzers reagiert.

Lassen Sie uns zunächst das HTML- und CSS-Framework erstellen, das für die Anwendung verwendet wird. Dies ist eine einfache Seite mit einem `<div>`-Element, das den Kartencontainer enthält, und einem `<pre>`-Element, um die JSON-Antworten auf Ihre Abfragen anzuzeigen.

So erstellen Sie den HTML-Code für Ihre Schnellstartanwendung

1. Erstellen Sie eine neue Datei mit dem Namen `quickstart.html`.
2. Bearbeiten Sie die Datei im Texteditor oder in einer Umgebung Ihrer Wahl. Fügen Sie der Datei den folgenden HTML-Code hinzu.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>
```

```
<!-- Styles -->
<link href="main.css" rel="stylesheet" />
</head>

<body>
  <header>
    <h1>Quick start tutorial</h1>
  </header>
  <main>
    <div id="map"></div>
    <aside>
      <h2>JSON Response</h2>
      <pre id="response"></pre>
    </aside>
  </main>
  <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
  see details about entities close to a point.</footer>

</body>
</html>
```

Dieser HTML-Code enthält einen Zeiger auf die CSS-Datei, die Sie im nächsten Schritt erstellen werden, einige Platzhalterelemente für die Anwendung und etwas erklärenden Text.

Es gibt zwei Platzhalterelemente, die Sie später in diesem Tutorial verwenden werden. Die erste ist das `<div id="map">`Element, das die Kartensteuerung enthält. Die zweite ist das `<pre id="response">`Element, das die Ergebnisse der Suche auf der Karte anzeigt.

3. Speichern Sie Ihre Datei.

Fügen Sie nun das CSS für die Webseite hinzu. Dadurch wird der Stil des Textes und der Platzhalterelemente für die Anwendung festgelegt.

So erstellen Sie das CSS für Ihre Schnellstartanwendung

1. Erstellen Sie eine neue Datei mit dem Namen im selben Ordner wie die Datei `quickstart.htmlmain.css`, die im vorherigen Verfahren erstellt wurde.
2. Bearbeiten Sie die Datei in einem beliebigen Editor, den Sie verwenden möchten. Fügen Sie der Datei den folgenden Text hinzu.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}

h1 {
  margin: 0;
  text-align: center;
  font-size: 1.5rem;
  color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0
#001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

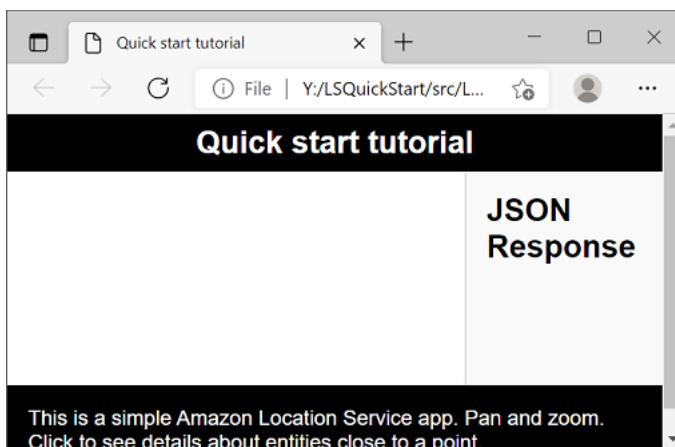
h2 {
  margin: 0;
}
```

```
pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

Dadurch wird die Karte so eingestellt, dass sie den Raum ausfüllt, der nicht von anderen verwendet wird, der Bereich für unsere Antworten wird so eingestellt, dass er 30 % der Breite der App in Anspruch nimmt, und Farbe und Stile für den Titel und den erklärenden Text festgelegt.

3. Speichern Sie die Datei.
4. Sie können die `quickstart.html` Datei jetzt in einem Browser anzeigen, um das Layout der Anwendung anzuzeigen.



Als Nächstes fügen Sie das Kartensteuerelement der Anwendung hinzu.

Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung

Nachdem Sie nun ein Framework und einen div-Platzhalter haben, können Sie die Kartensteuerung zu Ihrer Anwendung hinzufügen. In diesem Tutorial wird [MapLibre microSD JS](#) als Kartensteuerung verwendet, um Daten von Amazon Location Service abzurufen. Sie verwenden auch die [JavaScript](#)

[Authentifizierungshelfer](#), um das Signieren von Aufrufen an die Amazon Location APIs mit Ihrem API-Schlüssel zu erleichtern.

So fügen Sie Ihrer Anwendung eine interaktive Zuordnung hinzu

1. Öffnen Sie die `quickstart.html` Datei, die Sie im vorherigen Abschnitt erstellt haben.
2. Fügen Sie Verweise auf die benötigten Bibliotheken und die Skriptdatei hinzu, die Sie erstellen werden. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigtgreen**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>

    <!-- Styles -->
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <link href="main.css" rel="stylesheet" />
  </head>

  <body>
    ...
    <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
    see details about entities close to a point.</footer>

    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

    <!-- JavaScript for the app -->
    <script src="main.js"></script>
  </body>
</html>
```

Dadurch werden Ihrer App die folgenden Abhängigkeiten hinzugefügt:

- MapLibre Bol JS . Diese Bibliothek und dieses Stylesheet enthalten ein Kartensteuerelement, das Kartenkacheln anzeigt und Interaktivität wie Pan und Zoom enthält. Die Kontrolle ermöglicht auch Erweiterungen, z. B. das Zeichnen eigener Features auf der Karte.
- Amazon Location-Client . Dies bietet Schnittstellen für die Amazon Location-Funktionalität, die zum Abrufen von Kartendaten und zum Suchen nach Orten auf der Karte erforderlich ist. Der Amazon Location-Client basiert auf dem AWS SDK für JavaScript v3.
- Amazon Location Authentication Helper . Dies bietet hilfreiche Funktionen für die Authentifizierung von Amazon Location Service mit API-Schlüsseln oder Amazon Cognito .

In diesem Schritt wird auch ein Verweis auf hinzugefügt `main.js`, den Sie als Nächstes erstellen.

3. Speichern Sie die `quickstart.html`-Datei.
4. Erstellen Sie eine neue Datei mit dem Namen `main.js` im selben Ordner wie Ihre HTML- und CSS-Dateien und öffnen Sie sie zur Bearbeitung.
5. Fügen Sie der Datei das folgende Skript hinzu. Der Text in *Rot* sollte durch den API-Schlüsselwert, den Kartenressourcennamen und den Ortsressourcennamen ersetzt werden, den Sie zuvor erstellt haben, sowie durch die Regions-ID für Ihre Region (z. B. `us-east-1`).

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4..."

// Initialize a map
async function initializeMap() {
  const mglMap = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
using an API key
  });

  // Add navigation control to the top left of the map
  mglMap.addControl(new maplibregl.NavigationControl(), "top-left");
```

```
    return mlg1Map;
  }

  async function main() {
    // Initialize map and Amazon Location SDK client:
    const map = await initializeMap();
  }

  main();
```

Dieser Code richtet Amazon Location-Ressourcen ein, konfiguriert und initialisiert dann eine MapLibre microSD JS-Kartensteuerung und platziert sie in Ihrem Element `<div>` mit der ID `map`.

Die `initializeMap()` Funktion ist wichtig zu verstehen. Es erstellt ein neues MapLibre Kartensteuerelement (mlglMaplokal genannt, aber `map` im Rest des Codes aufgerufen), das zum Rendern der Karte in Ihrer Anwendung verwendet wird.

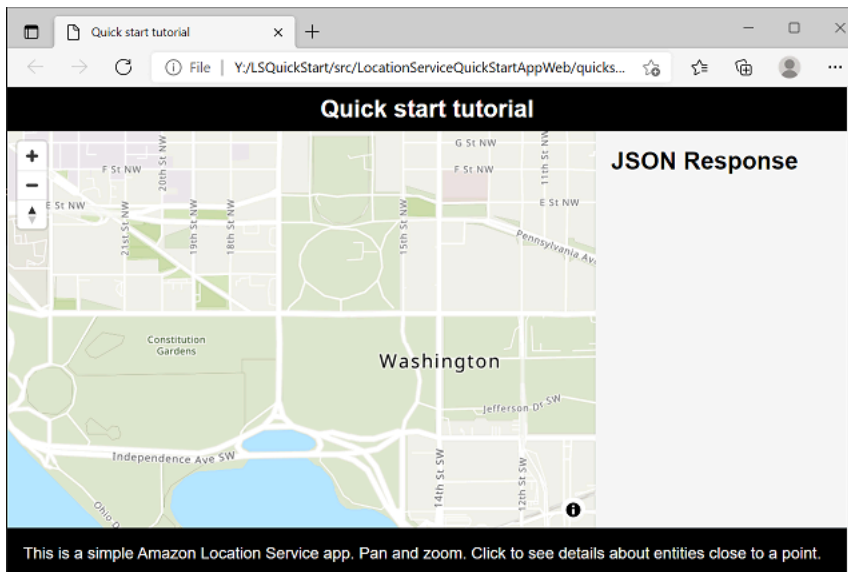
```
// Initialize the map
const mlg1Map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
  using an API key
});
```

Wenn Sie ein neues MapLibre Kartensteuerelement erstellen, geben die von Ihnen übergebenen Parameter den Anfangsstatus des Kartensteuerelements an. Hier legen wir die folgenden Parameter fest.

- HTML-Container, der das Map-Div-Element in unserem HTML verwendet.
 - Die anfängliche Mitte der Zuordnung zu einem Punkt in Bol, DC.
 - Die Zoomstufe auf 16 (in eine Nachbarschafts- oder Blockstufe gezoomt).
 - Der für die Karte zu verwendende Stil, der MapLibre eine URL zum Abrufen der Kartenkacheln und andere Informationen zum Rendern der Karte bereitstellt. Beachten Sie, dass diese URL Ihren API-Schlüssel für die Authentifizierung enthält.
6. Speichern Sie Ihre JavaScript Datei und öffnen Sie sie mit einem Browser. Sie haben jetzt eine Karte auf Ihrer Seite, auf der Sie Pan- und Zoom-Aktionen verwenden können.

Note

Sie können diese App verwenden, um zu sehen, wie sich die MapLibre Kartensteuerung verhält. Sie können versuchen, Strg oder Verschiebung zu verwenden, während Sie eine Ziehen-Operation verwenden, um andere Möglichkeiten für die Interaktion mit der Karte anzuzeigen. Alle diese Funktionen sind anpassbar.



Ihre App ist fast vollständig. Im nächsten Abschnitt übernehmen Sie die Auswahl eines Standorts auf der Karte und zeigen die Adresse des ausgewählten Standorts an. Sie zeigen auch den resultierenden JSON-Code auf der Seite an, um die vollständigen Ergebnisse anzuzeigen.

Hinzufügen einer Suche zu Ihrer Anwendung

Der letzte Schritt für Ihre Anwendung besteht darin, die Suche auf der Karte hinzuzufügen. In diesem Fall fügen Sie eine umgekehrte Geocodierungssuche hinzu, bei der Sie die Elemente an einem Speicherort finden.

Note

Amazon Location Service bietet auch die Möglichkeit, nach Namen oder Adresse zu suchen, um die Standorte auf der Karte zu finden.

So fügen Sie Ihrer Anwendung Suchfunktionen hinzu

1. Öffnen Sie die `main.js` Datei, die Sie im vorherigen Abschnitt erstellt haben.
2. Ändern Sie die `main` Funktion wie gezeigt. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigtgreen**.

```
async function main() {
  // Create an authentication helper instance using an API key
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

  // Initialize map and Amazon Location SDK client:
  const map = await initializeMap();

  const client = new amazonLocationClient.LocationClient({
    region,
    ...authHelper.getLocationClientConfig(), // Provides configuration required to
    make requests to Amazon Location
  });

  // On mouse click, display marker and get results:
  map.on("click", async function (e) {
    // Set up parameters for search call
    let params = {
      IndexName: placesName,
      Position: [e.lngLat.lng, e.lngLat.lat],
      Language: "en",
      MaxResults: "5",
    };

    // Set up command to search for results around clicked point
    const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

    try {
      // Make request to search for results around clicked point
      const data = await client.send(searchCommand);

      // Write JSON response data to HTML
      document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);
    }
  });
}
```

```
// Display place label in an alert box
alert(data.Results[0].Place.Label);
} catch (error) {
  // Write JSON response error to HTML
  document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

  // Display error in an alert box
  alert("There was an error searching.");
}
});
}
```

Dieser Code beginnt mit der Einrichtung des Amazon Location Authentication Helpers für die Verwendung Ihres API-Schlüssels.

```
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);
```

Dann verwendet es dieses Authentifizierungshelferobjekt und die Region, die Sie zum Erstellen eines neuen Amazon-Location-Clients verwenden.

```
const client = new amazonLocationClient.LocationClient({
  region,
  ...authHelper.getLocationClientConfig(),
});
```

Als Nächstes antwortet der Code dem Benutzer, der einen Spot auf dem Kartensteuerelement auswählt. Dazu wird ein MapLibre bereitgestelltes Ereignis für `abgefangenClick`.

```
map.on("click", async function(e) {
  ...
});
```

Das MapLibre `click` Ereignis stellt Parameter bereit, die den Breiten- und Längengrad enthalten, den der Benutzer ausgewählt hat (`e.lngLat`). Innerhalb des `click` Ereignisses erstellt der Code die `searchPlaceIndexForPositionCommand` um die Entitäten mit dem angegebenen Breiten- und Längengrad zu finden.

```
// Set up parameters for search call
let params = {
```

```
    IndexName: placesName,  
    Position: [e.lngLat.lng, e.lngLat.lat],  
    Language: "en",  
    MaxResults: "5"  
  };  
  
  // Set up command to search for results around clicked point  
  const searchCommand = new  
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);  
  
  try {  
    // Make request to search for results around clicked point  
    const data = await client.send(searchCommand);  
    ...  
  });
```

Hier `IndexName` ist der Name der zuvor erstellten Ortsindex-Ressource, der `Position` Breiten- und Längengrad, nach dem gesucht werden soll, `Language` die bevorzugte Sprache für Ergebnisse und `MaxResults` weist Amazon Location an, maximal fünf Ergebnisse zurückzugeben.

Der verbleibende Code prüft auf einen Fehler und zeigt dann die Ergebnisse der Suche im `<pre>` Element mit dem Namen `response` und zeigt das oberste Ergebnis in einem Warnfeld an.

3. (Optional) Wenn Sie die `quickstart.html` Datei jetzt in einem Browser speichern und öffnen, wird Ihnen bei Auswahl eines Speicherorts auf der Karte der Name oder die Adresse des ausgewählten Speicherorts angezeigt.
4. Der letzte Schritt in der Anwendung besteht darin, die `MapLibre` Funktionalität zu verwenden, um eine Markierung an dem Punkt hinzuzufügen, den der Benutzer ausgewählt hat. Ändern Sie die `main` Funktion wie folgt. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigt green**.

```
async function main() {  
  // Create an authentication helper instance using an API key  
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);  
  
  // Initialize map and Amazon Location SDK client  
  const map = await initializeMap();  
  const client = new amazonLocationClient.LocationClient({  
    region,
```

```
...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
});

// Variable to hold marker that will be rendered on click
let marker;

// On mouse click, display marker and get results:
map.on("click", async function (e) {
  // Remove any existing marker
  if (marker) {
    marker.remove();
  }

  // Render a marker on clicked point
  marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

  // Set up parameters for search call
  let params = {
    IndexName: placesName,
    Position: [e.lngLat.lng, e.lngLat.lat],
    Language: "en",
    MaxResults: "5",
  };

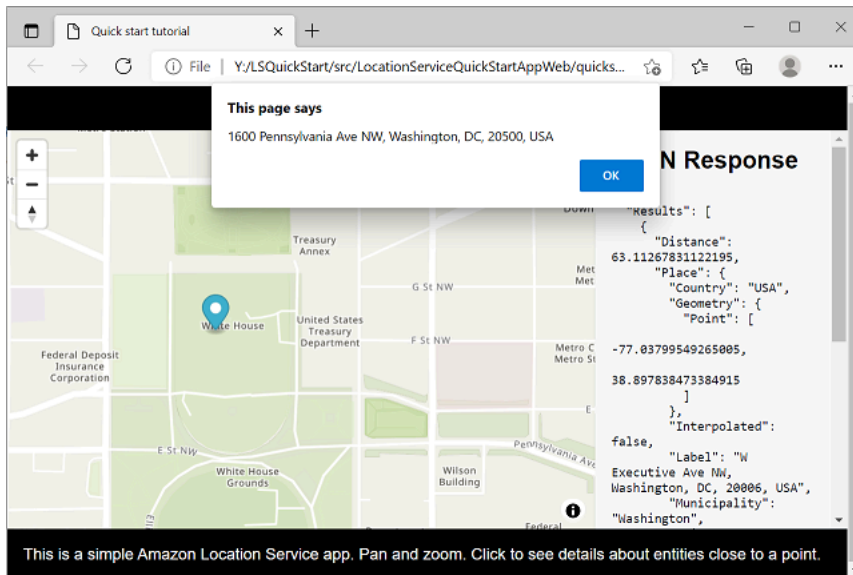
  // Set up command to search for results around clicked point
  const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

...

```

Dieser Code deklariert eine `marker` Variable, die jedes Mal ausgefüllt wird, wenn der Benutzer einen Standort auswählt, und zeigt, wo er ihn ausgewählt hat. Die Markierung wird automatisch vom Kartensteuerelement gerendert, sobald sie der Karte mit hinzugefügt wurde `.addTo(map)`. Der Code sucht auch nach einer vorherigen Markierung und entfernt sie, sodass jeweils nur eine Markierung auf dem Bildschirm angezeigt wird.

5. Speichern Sie die `main.js` Datei und öffnen Sie die `quickstart.html` Datei in einem Browser. Sie können die Karte wie zuvor schwenken und zoomen, aber wenn Sie jetzt einen Standort auswählen, werden Ihnen Details zu dem ausgewählten Standort angezeigt.



Ihre Schnellstartanwendung ist abgeschlossen. In diesem Tutorial erfahren Sie, wie Sie eine statische HTML-Anwendung erstellen, die:

- Erstellt eine Zuordnung, mit der Benutzer interagieren können.
- Verarbeitet ein Karteneignis (`click`).
- Ruft eine Amazon Location Service-API auf, insbesondere zum Durchsuchen der Karte an einem Standort mit `searchPlaceIndexForPosition`.
- Verwendet das MapLibre Kartensteuerelement, um eine Markierung hinzuzufügen.

Anzeigen der endgültigen Anwendung

Der endgültige Quellcode für diese Anwendung ist in diesem Abschnitt enthalten. Sie finden das endgültige Projekt auch [unter GitHub](#).

Sie finden auch eine Version der Anwendung, die Amazon Cognito anstelle von API-Schlüsseln [auf GitHub](#) verwendet.

Overview

Wählen Sie jede Registerkarte aus, um den endgültigen Quellcode der Dateien in diesem Schnellstart-Tutorial anzuzeigen.

Die Dateien sind:

- quickstart.html – das Framework für Ihre Anwendung, einschließlich der HTML-Elementinhaber für die Karte und die Suchergebnisse.
- main.css – das Stylesheet für die Anwendung.
- main.js – das Skript für Ihre Anwendung, das den Benutzer authentifiziert, die Zuordnung erstellt und nach einem click Ereignis sucht.

quickstart.html

Das HTML-Framework für die Schnellstartanwendung.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>

    <!-- Styles -->
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <link href="main.css" rel="stylesheet" />
  </head>

  <body>
    ...
    <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to see
details about entities close to a point.</footer>

    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

    <!-- JavaScript for the app -->
    <script src="main.js"></script>
  </body>
</html>
```

main.css

Das Stylesheet für die Schnellstartanwendung.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}

h1 {
  margin: 0;
  text-align: center;
  font-size: 1.5rem;
  color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0 #001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

h2 {
  margin: 0;
}
```

```
pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

main.js

Der Code für die Schnellstartanwendung. Der Text in *Rot* sollte durch die entsprechenden Amazon Location-Objektnamen ersetzt werden.

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4...

// Initialize a map
async function initializeMap() {
  // Initialize the map
  const mlglMap = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}`, // Defines the appearance of the map and authenticates
using an API key
  });

  // Add navigation control to the top left of the map
  mlglMap.addControl(new maplibregl.NavigationControl(), "top-left");

  return mlglMap;
}

async function main() {
```

```
// Create an authentication helper instance using an API key
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

// Initialize map and Amazon Location SDK client
const map = await initializeMap();
const client = new amazonLocationClient.LocationClient({
  region,
  ...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
});

// Variable to hold marker that will be rendered on click
let marker;

// On mouse click, display marker and get results:
map.on("click", async function (e) {
  // Remove any existing marker
  if (marker) {
    marker.remove();
  }

  // Render a marker on clicked point
  marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

  // Set up parameters for search call
  let params = {
    IndexName: placesName,
    Position: [e.lngLat.lng, e.lngLat.lat],
    Language: "en",
    MaxResults: "5",
  };

  // Set up command to search for results around clicked point
  const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

  try {
    // Make request to search for results around clicked point
    const data = await client.send(searchCommand);

    // Write JSON response data to HTML
    document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);
```

```
// Display place label in an alert box
alert(data.Results[0].Place.Label);
} catch (error) {
    // Write JSON response error to HTML
    document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

    // Display error in an alert box
    alert("There was an error searching.");
}
});
}

main();
```

Was ist als Nächstes?

Sie haben das Schnellstart-Tutorial abgeschlossen und sollten sich ein Bild davon machen, wie Amazon Location Service zum Erstellen von Anwendungen verwendet wird. Um Amazon Location besser zu nutzen, können Sie sich die folgenden Ressourcen ansehen:

- Weitere Informationen zu den [Konzepten von Amazon Location Service](#)
- Weitere Informationen [zur Verwendung von Amazon Location-Funktionen und -Funktionen](#)
- Sehen Sie sich [Codebeispiele mit Amazon Location](#) an, um dieses Beispiel zu erweitern und komplexere Anwendungen zu erstellen

Eine Android-App erstellen

In diesem Abschnitt erstellen Sie eine Android-Anwendung mit einer Karte, der Möglichkeit, an einem Ort zu suchen und das Tracking im Vordergrund zu verfolgen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen, eine Amazon Cognito Cognito-Identität und einen API-Schlüssel für Ihre Anwendung.

Themen

- [Amazon-Standortressourcen für Ihre App erstellen](#)
- [Richten Sie die Authentifizierung für Ihre Anwendung ein](#)
- [Erstellen der Android-Basisanwendung](#)

- [Fügen Sie Ihrer Anwendung eine interaktive Karte hinzu](#)
- [Reverse-Geocoding-Suche zu Ihrer Anwendung hinzufügen](#)
- [Hinzufügen von Tracking zu Ihrer Anwendung](#)
- [Was kommt als Nächstes](#)

Amazon-Standortressourcen für Ihre App erstellen

Wenn Sie sie noch nicht haben, müssen Sie die Amazon-Standortressourcen erstellen, die Ihre Anwendung verwenden wird. Hier erstellen Sie eine Kartenressource, um Karten in Ihrer Anwendung anzuzeigen, einen Ortsindex, um nach Orten auf der Karte zu suchen, und einen Tracker, um ein Objekt auf der Karte zu verfolgen.

Um Ihrer Anwendung Standortressourcen hinzuzufügen


1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen ihn, wenn Sie Ihre Skriptdatei später im Tutorial erstellen.
 - c. Wir empfehlen Ihnen, den HERE-Kartenstil für Ihre Karte zu wählen.

Note

Durch die Auswahl eines Kartenstils wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Ressourcen verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Lieferfahrzeuge oder Mitarbeiter, dürfen Sie HERE nur als Ihren Geolocation-Anbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen von Amazon Location zu und wählen Sie dann Create map aus. Sie können mit der Karte, die Sie ausgewählt haben, interagieren: vergrößern, verkleinern oder in eine beliebige Richtung schwenken.

- e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie werden ihn später in diesem Tutorial verwenden, um die richtige Authentifizierung zu erstellen.
2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen ihn, wenn Sie Ihre Skriptdatei später im Tutorial erstellen.
 - c. Wählen Sie einen Datenanbieter.

 Note

Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem Kartenanbieter entspricht, den Sie bereits ausgewählt haben. Auf diese Weise können Sie sicherstellen, dass die Suchanfragen mit den Karten übereinstimmen. Wenn Ihre Anwendung Ressourcen verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Lieferfahrzeuge oder Mitarbeiter, dürfen Sie HERE nur als Ihren Geolocation-Anbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher. In diesem Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie „Nein, nur zur einmaligen Verwendung“ wählen können.
 - e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon-Standorte zu und wählen Sie dann Ortsindex erstellen.
 - f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie werden ihn verwenden, um im nächsten Abschnitt dieses Tutorials die richtige Authentifizierung zu erstellen.
3. Um einen Tracker mit der Amazon Location-Konsole zu erstellen.
 - a. Öffnen Sie die [Amazon Location Service Service-Konsole](#).
 - b. Wählen Sie im linken Navigationsbereich Trackers aus.
 - c. Wählen Sie Tracker erstellen.
 - d. Füllen Sie alle erforderlichen Felder aus.

- e. Unter Positionsfiltrierung empfehlen wir Ihnen, die Standardeinstellung zu verwenden: TimeBased.
- f. Wählen Sie Tracker erstellen, um den Vorgang abzuschließen.

Richten Sie die Authentifizierung für Ihre Anwendung ein

Die Anwendung, die Sie in diesem Tutorial erstellen, wird anonym verwendet, was bedeutet, dass sich Ihre Benutzer nicht anmelden müssen AWS, um die Anwendung zu verwenden. Für die Verwendung der Amazon Location Service APIs ist jedoch eine Authentifizierung erforderlich. Sie können entweder API-Schlüssel oder Amazon Cognito verwenden, um anonyme Benutzer zu authentifizieren und zu autorisieren. In diesem Tutorial werden Amazon Cognito und API-Schlüssel verwendet, um Ihre Anwendung zu authentifizieren.

Note

Weitere Informationen zur Verwendung von Amazon Cognito oder API-Schlüsseln mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Die folgenden Tutorials zeigen Ihnen, wie Sie die Authentifizierung für die Karte, den Ortsindex und den Tracker einrichten, in denen Sie erstellt haben, sowie wie Sie Berechtigungen für Amazon Location einrichten.

Richten Sie die Authentifizierung ein

1. Navigieren Sie zur [Amazon Location-Konsole](#) und wählen Sie im Menü auf der linken Seite API-Schlüssel aus.
2. Klicken Sie auf „API-Schlüssel erstellen“. Denken Sie daran, dass sich der API-Schlüssel in demselben AWS Konto und derselben Region befinden muss wie die zuvor erstellten Amazon Location Service Service-Ressourcen.
3. Geben Sie die erforderlichen Details auf der Seite „API-Schlüssel erstellen“ ein:
 - Name: Geben Sie einen Namen für Ihren API-Schlüssel ein, z. MyAppKey
 - Ressourcen: Wählen Sie die zuvor erstellten Amazon Location Service Map and Place Index-Ressourcen aus. Sie können mehrere Ressourcen hinzufügen, indem Sie „Ressource

hinzufügen“ auswählen. Dadurch kann der API-Schlüssel mit bestimmten Ressourcen verwendet werden.

- **Aktionen:** Geben Sie autorisierte Aktionen für diesen API-Schlüssel an. Wählen Sie mindestens `geo:GetMap` und, `geo:SearchPlaceIndexForPosition` um sicherzustellen, dass das Tutorial wie vorgesehen funktioniert.
 - Optional können Sie eine Beschreibung, eine Ablaufzeit, Tags oder einen Verweis hinzufügen, `https://www.example.com` um beispielsweise die Verwendung des Schlüssels auf eine bestimmte Domain zu beschränken, sodass das Tutorial nur innerhalb dieser Domain funktioniert.
4. Klicken Sie auf **API-Schlüssel erstellen**, um den API-Schlüssel zu generieren.
 5. Wählen Sie **API-Schlüssel anzeigen** aus und kopieren Sie den Schlüsselwert, `v1.public.a1b2c3d4` um ihn beispielsweise später im Tutorial zu verwenden.

Erstellen Sie eine IAM-Richtlinie für das Tracking

1. Melden Sie sich bei der IAM-Konsole unter <https://console.aws.amazon.com/iam/> mit Ihrem Benutzer an, der über Administratorrechte verfügt.
2. Wählen Sie im Navigationsbereich **Policies**.
3. Wählen Sie im Inhaltsbereich die Option **Create policy** (Richtlinie erstellen).
4. Wählen Sie die **JSON**-Option, kopieren Sie dann diese JSON-Richtlinie und fügen Sie sie in das JSON-Textfeld ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapStyleDescriptor",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:SearchPlaceIndexForPosition",
        "geo:GetDevicePositionHistory",
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
```

```
        "arn:aws:geo:{Region}:{Account}:map/{MapName}",
        "arn:aws:geo:{Region}:{Account}:place-index/{IndexName}",
        "arn:aws:geo:{Region}:{Account}:tracker/{TrackerName}"
    ]
}
}
```

Dies ist ein Richtlinienbeispiel für Tracking. Um das Beispiel für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die TrackerNamePlatzhalter RegionAccount, und.

Note

Nicht authentifizierte Identitätspools sind zwar für die Veröffentlichung auf ungesicherten Internetseiten vorgesehen, beachten Sie jedoch, dass sie gegen standardmäßige, zeitlich begrenzte AWS-Anmeldeinformationen ausgetauscht werden.

Es ist wichtig, die IAM-Rollen, die mit Pools nicht authentifizierter Identitäten verknüpft sind, angemessen abzugrenzen. Weitere Informationen zur Verwendung und angemessenen Festlegung von Richtlinien in Amazon Cognito mit Amazon Location Service finden Sie unter [Zugriff auf Amazon Location Service gewähren](#).

5. Geben Sie auf der Seite Überprüfen und erstellen einen Namen für das Feld Richtlinienname ein. Überprüfen Sie die durch Ihre Richtlinie gewährten Berechtigungen und wählen Sie dann Richtlinie erstellen aus, um Ihre Arbeit zu speichern.

Die neue Richtlinie wird in der Liste der verwalteten Richtlinien angezeigt und ist bereit.

Richten Sie die Authentifizierung für Ihr Tracking ein

1. Richten Sie die Authentifizierung für Ihre Kartenanwendung in der [Amazon Cognito Cognito-Konsole](#) ein.
2. Öffnen Sie die Seite Identitätspools.

Note

Der Pool, den Sie erstellen, muss sich in demselben AWS Konto und derselben AWS Region befinden wie die Amazon Location Service Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

3. Wählen Sie Identitätspool erstellen.
4. Beginnen Sie mit dem Vertrauensschritt Identitätspool konfigurieren. Wählen Sie für die Benutzerzugriffsauthentifizierung die Option Gastzugriff aus und klicken Sie auf Weiter.
5. Wählen Sie auf der Seite Berechtigungen konfigurieren die Option Bestehende IAM-Rolle verwenden aus und geben Sie den Namen der IAM-Rolle ein, die Sie im vorherigen Schritt erstellt haben. Wenn Sie bereit sind, klicken Sie auf Weiter, um mit dem nächsten Schritt fortzufahren.
6. Geben Sie auf der Seite Eigenschaften konfigurieren einen Namen für Ihren Identitätspool ein. Drücken Sie dann auf Weiter.
7. Überprüfen Sie auf der Seite Überprüfen und erstellen alle vorhandenen Informationen und klicken Sie dann auf Identitätspool erstellen.
8. Öffnen Sie die Seite Identitätspools und wählen Sie den Identitätspool aus, den Sie gerade erstellt haben. Kopieren Sie dann den IdentityPoolId , den Sie später verwenden werden, oder notieren Sie ihn in Ihrem Browser-Skript.

Erstellen der Android-Basisanwendung

In diesem Tutorial erstellen Sie eine Android-Anwendung, die eine Karte einbettet und es dem Benutzer ermöglicht, herauszufinden, was sich an einem Ort auf der Karte befindet.

Erstellen Sie zunächst eine leere Kotlin-Anwendung mit dem neuen Projekt-Assistenten von Android Studio.

Um eine leere Anwendung zu erstellen () AndroidStudio

1. Fangen Sie an AndroidStudio. Öffnen Sie das Menü und wählen Sie Datei, Neu, Neues Projekt.
2. Wählen Sie auf der Registerkarte „Telefon und Tablet“ die Option „Aktivität leeren“ und dann „Weiter“ aus.
3. Wählen Sie einen Namen, einen Paketnamen und einen Speicherort für Ihre Anwendung.
4. Wählen Sie in der Dropdownliste für Sprache die Option Kotlin aus.
5. Wählen Sie Fertig stellen, um Ihre leere Anwendung zu erstellen.

Fügen Sie Ihrer Anwendung eine interaktive Karte hinzu

Nachdem Sie eine Basisanwendung erstellt haben, können Sie Ihrer Anwendung Map Control hinzufügen. In diesem Tutorial werden API-Schlüssel für die Verwaltung der Kartenansicht verwendet. Das Kartensteuerelement selbst ist Teil der [MapLibre Native-Bibliothek](#) mit dem API-Schlüssel und MapLibre, und die Kartendaten stammen von Amazon Location.

Um Ihrer Anwendung eine Karte hinzuzufügen, müssen Sie die folgenden Aktionen ausführen:

- Fügen Sie die MapLibre Abhängigkeit zu Ihrem Projekt hinzu.
- Richten Sie den Kartenansichtscode mit Compose ein.
- Schreiben Sie den Code, um die Karte anzuzeigen.

Gehen Sie wie folgt vor, um die Karte zu Ihrer App hinzuzufügen:

1. Fügen Sie die MapLibre Abhängigkeit zu Ihrem Projekt hinzu
 - a. Wählen Sie in AndroidStudio das Menü Ansicht und dann Werkzeugfenster, Projekt aus. Dadurch wird das Projektfenster geöffnet, über das Sie auf alle Dateien in Ihrem Projekt zugreifen können.
 - b. Öffnen Sie im Projektfenster Gradle und öffnen Sie die `libs.versions.toml` Datei dann in der Baumansicht. Dadurch wird die `libs.versions.toml` Datei zur Bearbeitung geöffnet. Fügen Sie nun die folgenden Versions- und Bibliotheksdaten in die `libs.versions.toml` Datei ein.

```
[versions]
...
auth = "0.2.4"
tracking = "0.2.4"

[libraries]
...
auth = { group = "software.amazon.location", name = "auth", version.ref =
"auth" }
tracking = { module = "software.amazon.location:tracking", version.ref =
"tracking" }

[plugins]
...
```

- c. Nachdem Sie die Bearbeitung der `libs.versions.toml` Datei abgeschlossen haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Am oberen Rand des `libs.versions.toml` Bearbeitungsfensters werden Sie AndroidStudio zur Synchronisierung aufgefordert. Wählen Sie „Jetzt synchronisieren“, um Ihr Projekt zu synchronisieren, bevor Sie fortfahren.
- d. Öffnen Sie im Projektfenster Gradle Scripts in der Baumansicht und wählen Sie die `build.gradle` Datei für Ihr Anwendungsmodul aus. Dadurch wird die `build.gradle` Datei zur Bearbeitung geöffnet.
- e. Fügen Sie am Ende der Datei im Abschnitt Abhängigkeiten die folgende Abhängigkeit hinzu.

```
dependencies {  
    ...  
    implementation(libs.org.maplibre.gl)  
}
```

- f. Nachdem Sie das Hinzufügen der Gradle-Abhängigkeiten abgeschlossen haben, muss Android Studio das Projekt erneut synchronisieren. Wählen Sie oben im `build.gradle`-Bearbeitungsfenster von Android Studio die Option Jetzt synchronisieren aus, um Ihr Projekt zu synchronisieren, bevor Sie fortfahren.
2. Jetzt richten Sie den Kartenansichtscode mit Compose ein. Gehen Sie dazu wie folgt vor:
 - a. Öffnen Sie im Projektfenster App, Java, *Ihren Paketnamen* in der Baumansicht und wechseln Sie zum UI-Ordner. Erstellen Sie im UI-Ordner ein Ansichtsverzeichnis.
 - b. Erstellen Sie im View-Verzeichnis eine `MapLoadScreen.kt` Datei.
 - c. Fügen Sie nun folgenden Code in die Datei `MapLoadScreen.kt` ein:

```
import androidx.compose.foundation.layout.Box  
import androidx.compose.foundation.layout.fillMaxHeight  
import androidx.compose.foundation.layout.fillMaxWidth  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.viewinterop.AndroidView  
import org.maplibre.android.maps.OnMapReadyCallback  
  
@Composable  
fun MapLoadScreen(  
    mapReadyCallback: OnMapReadyCallback,  
) {  
    Box(  

```

```

        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}

```

3. Schreiben Sie Code, um die Karte anzuzeigen.

a. Fügen Sie nun folgenden Code in die Datei MainActivity.kt ein:

```

// ...other imports
import org.maplibre.android.MapLibre
import org.maplibre.android.camera.CameraPosition
import org.maplibre.android.geometry.LatLng
import org.maplibre.android.maps.MapLibreMap
import org.maplibre.android.maps.OnMapReadyCallback
import org.maplibre.android.maps.Style

class MainActivity : ComponentActivity(), OnMapReadyCallback {
    private val region = "YOUR_AWS_REGION"
    private val mapName = "YOUR_AWS_MAP_NAME"
    private val apiKey = "YOUR_AWS_API_KEY"
    override fun onCreate(savedInstanceState: Bundle?) {
        MapLibre.getInstance(this)
        super.onCreate(savedInstanceState)
        setContent {
            TestMapAppTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                )
            }
        }
    }
}

```

```
        ) {
            MapLoadScreen(this)
        }
    }
}

override fun onMapReady(map: MapLibreMap) {
    map.setStyle(
        Style.Builder()
            .fromUri(
                "https://maps.geo.$region.amazonaws.com/maps/v0/maps/
$mapName/style-descriptor?key=$apiKey"
            ),
        ) {
        map.uiSettings.isAttributionEnabled = true
        map.uiSettings.isLogoEnabled = false
        map.uiSettings.attributionGravity = Gravity.BOTTOM or Gravity.END
        val initialPosition = LatLng(47.6160281982247,
-122.32642111977668)
        map.cameraPosition = CameraPosition.Builder()
            .target(initialPosition)
            .zoom(14.0)
            .build()
        }
    }
}
```

- b. Speichern Sie die MainActivity.kt-Datei. Sie können die Anwendung jetzt erstellen. Um sie auszuführen, müssen Sie möglicherweise ein Gerät einrichten, auf dem sie emuliert wird AndroidStudio, oder die App auf Ihrem Gerät verwenden. Verwenden Sie diese App, um zu sehen, wie sich die Kartensteuerung verhält. Sie können es schwenken, indem Sie es auf die Karte ziehen und zum Zoomen zusammendrücken.

Im nächsten Abschnitt fügen Sie der Karte eine Markierung hinzu und zeigen die Adresse des Standorts an, an dem sich die Markierung befindet, wenn Sie die Karte bewegen.

Reverse-Geocoding-Suche zu Ihrer Anwendung hinzufügen

Sie fügen Ihrer Anwendung nun die umgekehrte Geokodierungssuche hinzu, bei der Sie die Elemente an einem Ort finden. Um die Verwendung einer Android-App zu vereinfachen, suchen

wir in der Mitte des Bildschirms. Um einen neuen Standort zu finden, verschieben Sie die Karte an die Stelle, an der Sie suchen möchten. Wir platzieren eine Markierung in der Mitte der Karte, um zu zeigen, wo wir suchen.

Das Hinzufügen einer Reverse-Geocoding-Suche wird aus zwei Teilen bestehen.

- Fügen Sie in der Mitte des Bildschirms eine Markierung hinzu, um dem Benutzer zu zeigen, wo wir suchen.
- Fügen Sie ein Textfeld für Ergebnisse hinzu, suchen Sie dann nach dem, was sich an der Position der Markierung befindet, und zeigen Sie es im Textfeld an.

Um Ihrer Anwendung eine Markierung hinzuzufügen

1. Speichern Sie dieses Bild in Ihrem Projekt im `app/res/drawable` Ordner unter `red_marker.png` (Sie können das Bild auch von [aufrufen GitHub](#). Alternativ können Sie Ihr Bild erstellen. Sie können auch eine PNG-Datei mit Transparenz für die Teile verwenden, die nicht angezeigt werden sollen.
2. Fügen Sie Ihrer `MapLoadScreenKT`-Datei den folgenden Code hinzu.

```
// ...other imports
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.size
import androidx.compose.ui.Alignment
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import com.amazon.testmapapp.R

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
        Box(
            modifier = Modifier
                .align(Alignment.Center),
```

```
        ) {
            Image(
                painter = painterResource(id = R.drawable.red_marker),
                contentDescription = "marker",
                modifier = Modifier
                    .size(40.dp)
                    .align(Alignment.Center),
            )
        }
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}
```

3. Erstellen Sie Ihre App und führen Sie sie aus, um eine Vorschau der Funktionalität zu erhalten.

Ihre App hat jetzt eine Markierung auf dem Bildschirm. In diesem Fall handelt es sich um ein statisches Bild, das sich nicht bewegt. Es wird verwendet, um die Mitte der Kartenansicht anzuzeigen, in der wir suchen werden. Im folgenden Verfahren fügen wir die Suche an dieser Stelle hinzu.

So fügen Sie Ihrer App die umgekehrte Geokodierungssuche an einem Standort hinzu

1. Öffnen Sie im Projektfenster Gradle to `libs.versions.toml` File in der Baumansicht. Dadurch wird die `libs.versions.toml` Datei zur Bearbeitung geöffnet. Fügen Sie nun die folgenden Versions- und Bibliotheksdaten in die `libs.versions.toml` Datei ein.

```
[versions]
...
okhttp = "4.12.0"

[libraries]
...
```

```
com.squareup.okhttp3 = { group = "com.squareup.okhttp3", name = "okhttp",
    version.ref = "okhttp" }

[plugins]
...
```

2. Nachdem Sie die Bearbeitung der `libs.versions.toml` Datei abgeschlossen haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Am oberen Rand des `libs.versions.toml` Bearbeitungsfensters werden Sie AndroidStudio zur Synchronisierung aufgefordert. Wählen Sie „Jetzt synchronisieren“, um Ihr Projekt zu synchronisieren, bevor Sie fortfahren.
3. Öffnen Sie im Projektfenster Gradle Scripts in der Baumansicht und wählen Sie die `build.gradle` Datei für Ihr Anwendungsmodul aus. Dadurch wird die `build.gradle` Datei zur Bearbeitung geöffnet.
4. Fügen Sie am Ende der Datei im Abschnitt Abhängigkeiten die folgende Abhängigkeit hinzu.

```
dependencies {
    ...
    implementation(libs.com.squareup.okhttp3)
}
```

5. Nachdem Sie die Bearbeitung der Gradle-Abhängigkeiten abgeschlossen haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Am oberen Rand des `build.gradle` Bearbeitungsfensters werden Sie AndroidStudio zur Synchronisierung aufgefordert. Wählen Sie aus SyncNow, ob Sie Ihr Projekt synchronisieren möchten, bevor Sie fortfahren.
6. Fügen Sie nun in der Baumansicht die Daten zum Anforderungsverzeichnis hinzu und erstellen Sie die `ReverseGeocodeRequest.kt` Datenklasse. Fügen Sie der Klasse den folgenden Code hinzu.

```
import com.google.gson.annotations.SerializedName

data class ReverseGeocodeRequest(
    @SerializedName("Language")
    val language: String,
    @SerializedName("MaxResults")
    val maxResults: Int,
    @SerializedName("Position")
    val position: List<Double>
)
```

7. Fügen Sie nun in der Baumansicht die Daten zum Antwortverzeichnis hinzu und erstellen Sie die `ReverseGeocodeResponse.kt` Datenklasse. Fügen Sie den folgenden Code hinzu.

```
import com.google.gson.annotations.SerializedName

data class ReverseGeocodeResponse(
    @SerializedName("Results")
    val results: List<Result>
)

data class Result(
    @SerializedName("Place")
    val place: Place
)

data class Place(
    @SerializedName("Label")
    val label: String
)
```

8. Öffnen Sie nun im Projektfenster App, Java, *Ihren Paketnamen* in der Baumansicht und wechseln Sie zum UI-Ordner. Erstellen Sie im UI-Ordner das ViewModel-Verzeichnis.
9. Erstellen Sie im ViewModel-Verzeichnis eine `MainViewModel.kt` Datei.
10. Fügen Sie nun folgenden Code in die Datei `MainViewModel.kt` ein:

```
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
import androidx.lifecycle.ViewModel
import com.amazon.testmapapp.data.request.ReverseGeocodeRequest
import com.amazon.testmapapp.data.response.ReverseGeocodeResponse
import com.google.gson.Gson
import java.io.IOException
import okhttp3.Call
import okhttp3.Callback
import okhttp3.MediaType.Companion.toMediaTypeOrNull
import okhttp3.OkHttpClient
import okhttp3.Request
import okhttp3.RequestBody.Companion.toRequestBody
import okhttp3.Response
import org.maplibre.android.geometry.LatLng
import org.maplibre.android.maps.MapLibreMap
```

```
class MainViewModel : ViewModel() {
    var label by mutableStateOf("")
    var isLabelAdded: Boolean by mutableStateOf(false)
    var client = OkHttpClient()
    var mapLibreMap: MapLibreMap? = null

    fun reverseGeocode(latLng: LatLng, apiKey: String) {
        val region = "YOUR_AWS_REGION"
        val indexName = "YOUR_AWS_PLACE_INDEX"
        val url =
            "https://places.geo.{$region}.amazonaws.com/places/v0/indexes/
            {$indexName}/search/position?key={$apiKey}"

        val requestBody = ReverseGeocodeRequest(
            language = "en",
            maxResults = 1,
            position = listOf(latLng.longitude, latLng.latitude)
        )
        val json = Gson().toJson(requestBody)

        val mediaType = "application/json".toMediaTypeOrNull()
        val request = Request.Builder()
            .url(url)
            .post(json.toRequestBody(mediaType))
            .build()

        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                e.printStackTrace()
            }

            override fun onResponse(call: Call, response: Response) {
                if (response.isSuccessful) {
                    val jsonResponse = response.body?.string()

                    val reverseGeocodeResponse =
                        Gson().fromJson(jsonResponse,
ReverseGeocodeResponse::class.java)

                    val responseLabel =
reverseGeocodeResponse.results.firstOrNull()?.place?.label

                    if (responseLabel != null) {
```



```
        .align(Alignment.Center),
    ) {
        Image(
            painter = painterResource(id = R.drawable.red_marker),
            contentDescription = "marker",
            modifier = Modifier
                .size(40.dp)
                .align(Alignment.Center),
        )
    }
    if (mainViewModel.isLabelAdded) {
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Bottom
        ) {
            Box(
                modifier = Modifier
                    .fillMaxWidth()
                    .background(Color.White),
            ) {
                Text(
                    text = mainViewModel.label,
                    modifier = Modifier
                        .padding(16.dp)
                        .align(Alignment.Center)
                        .testTag("label")
                        .semantics {
                            contentDescription = "label"
                        },
                    fontSize = 14.sp,
                )
            }
            Spacer(modifier = Modifier.height(80.dp))
        }
    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
        }
    )
}
```

```

        mapView.getMapAsync(mapReadyCallback)
        mapView
    },
)
}

```

12. Öffnen Sie in der App unter Java im Paketnamenordner in AndroidStudio die MainActivity.kt Datei. Ändern Sie den Code wie gezeigt.

```

// ...other imports
import androidx.activity.viewModels
import com.amazon.testmapapp.ui.viewModel.MainViewModel

class MainActivity : ComponentActivity(), OnMapReadyCallback,
MapLibreMap.OnCameraMoveStartedListener, MapLibreMap.OnCameraIdleListener {

    private val mainViewModel: MainViewModel by viewModels()
    private val region = "YOUR_AWS_REGION"
    private val mapName = "YOUR_AWS_MAP_NAME"
    private val apiKey = "YOUR_AWS_API_KEY"
    override fun onCreate(savedInstanceState: Bundle?) {
        MapLibre.getInstance(this)
        super.onCreate(savedInstanceState)
        setContentView {
            TestMapAppTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    MapLoadScreen(this, mainViewModel)
                }
            }
        }
    }

    override fun onMapReady(map: MapLibreMap) {
        map.setStyle(
            Style.Builder()
                .fromUri(
                    "https://maps.geo.$region.amazonaws.com/maps/v0/maps/$mapName/
style-descriptor?key=$apiKey"
                ),
        ) {

```


- `ViewModel` hat eine Labelvariable: Diese Variable legt Daten in der Texterstellungsansicht fest.
 - `onMapReady` hat eine Label-Variable: Diese Variable legt Daten in der Texterstellungsansicht fest. ----SEP----:Diese Funktion wurde aktualisiert, um zwei neue Ereignisse zu registrieren.
 - Das `onCameraMove` Ereignis tritt immer dann ein, wenn der Benutzer die Karte bewegt. Im Allgemeinen möchten wir beim Verschieben der Karte die Suche ausblenden, bis der Benutzer mit dem Verschieben der Karte fertig ist.
 - Das `onCameraIdle` Ereignis tritt ein, wenn der Benutzer das Verschieben der Karte pausiert. Dieses Ereignis ruft unsere Reverse-Geocode-Funktion auf, um in der Kartenmitte zu suchen.
 - `reverseGeocode(latLng: LatLng, apiKey: String)`: Diese Funktion, die im Event aufgerufen wird `onCameraIdle`, sucht in der Kartenmitte nach einem Ort und aktualisiert die Bezeichnung, sodass die Ergebnisse angezeigt werden. Sie verwendet das Kameraziel, das den Mittelpunkt der Karte definiert (wohin die Kamera schaut).
13. Speichern Sie Ihre Dateien und erstellen Sie Ihre App und starten Sie sie, um zu sehen, ob sie funktioniert.

Ihre Schnellstartanwendung mit Suchfunktion ist fertig.

Hinzufügen von Tracking zu Ihrer Anwendung

Gehen Sie wie folgt vor, um Ihrer Beispielanwendung Tracking hinzuzufügen:

1. Fügen Sie Ihrem Projekt SDK-Abhängigkeiten für die Nachverfolgung und Authentifizierung hinzu.
 2. Fügen Sie Berechtigungs- und Serviceeinträge in Ihre `AndroidManifest XML`-Datei ein.
 3. Richten Sie den Code für die Start-/Stopp-Tracking-Schaltfläche mit Compose ein.
 4. Fügen Sie Code zum Erstellen eines `LocationTracker` Objekts hinzu und starten und beenden Sie das Tracking.
 5. Erstellen Sie eine Testroute mit Android Emulator.
-
1. Fügen Sie Ihrem Projekt SDK-Abhängigkeiten für Tracking und Authentifizierung hinzu.
 - a. Öffnen Sie im Projektfenster Gradle und öffnen Sie die `libs.versions.toml` Datei dann in der Baumansicht. Dadurch wird die `libs.versions.toml` Datei zur Bearbeitung geöffnet. Fügen Sie nun die folgenden Versions- und Bibliotheksdaten in die `libs.versions.toml` Datei ein.

```
[versions]
...
auth = "0.0.1"
tracking = "0.0.1"

[libraries]
...
auth = { group = "software.amazon.location", name = "auth", version.ref =
"auth" }
tracking = { module = "software.amazon.location:tracking", version.ref =
"tracking" }

[plugins]
...
```

- b. Nachdem Sie die Bearbeitung der `libs.versions.toml` Datei abgeschlossen haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Am oberen Rand des `libs.versions.toml` Bearbeitungsfensters werden Sie AndroidStudio zur Synchronisierung aufgefordert. Wählen Sie „Jetzt synchronisieren“, um Ihr Projekt zu synchronisieren, bevor Sie fortfahren.
- c. Öffnen Sie im Projektfenster Gradle Scripts in der Baumansicht und wählen Sie die `build.gradle` Datei für Ihr Anwendungsmodul aus. Dadurch wird die `build.gradle` Datei zur Bearbeitung geöffnet.
- d. Fügen Sie am Ende der Datei im Abschnitt Abhängigkeiten die folgende Abhängigkeit hinzu.

```
dependencies {
    ...
    implementation(libs.auth)
    implementation(libs.tracking)
}
```

- e. Nachdem Sie die Bearbeitung der Gradle-Abhängigkeiten abgeschlossen haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Am oberen Rand des `build.gradle`-Bearbeitungsfensters werden Sie zur Synchronisierung aufgefordert. AndroidStudio Wählen Sie aus SyncNow, ob Sie Ihr Projekt synchronisieren möchten, bevor Sie fortfahren.
2. Fügen Sie Berechtigungs- und Diensteeinträge in Ihre AndroidManifest XML-Datei ein.

- Um die richtigen Berechtigungs- und Dienstinträge in Ihre aufzunehmen `AndroidManifest.xml` file, aktualisieren Sie die Datei mit dem folgenden Code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AndroidQuickStartApp"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.AndroidQuickStartApp">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

3. Richten Sie den Code für die Start-/Stopp-Tracking-Schaltfläche mit Compose ein.
 - a. Fügen Sie zwei Bilder von Play und Pause in res unter dem Drawable mit den Namen `ic_pause` und `ic_play` hinzu. Sie können auf das Bild auch von zugreifen. [GitHub](#)

- b. Wenn es noch nicht geöffnet ist, öffnen Sie die `MapLoadScreen.kt` Datei wie im vorherigen Verfahren. Fügen Sie den folgenden Code hinzu. Dadurch wird eine Compose-Button-Ansicht erstellt, in der wir darauf klicken können, um das Tracking zu starten und zu beenden.

```
// ...other imports
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
    mainViewModel: MainViewModel,
    onStartStopTrackingClick: () -> Unit
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .fillMaxHeight(),
    ) {
        MapView(mapReadyCallback)
        Box(
            modifier = Modifier
                .align(Alignment.Center),
        ) {
            Image(
                painter = painterResource(id = R.drawable.red_marker),
                contentDescription = "marker",
                modifier = Modifier
                    .size(40.dp)
                    .align(Alignment.Center),
            )
        }
        if (mainViewModel.isLabelAdded) {
            Column(
                modifier = Modifier.fillMaxSize(),
                verticalArrangement = Arrangement.Bottom
            ) {
                Box(
                    modifier = Modifier
                        .fillMaxWidth()
                        .background(Color.White),
                ) {
```

```
        Text(
            text = mainViewModel.label,
            modifier = Modifier
                .padding(16.dp)
                .align(Alignment.Center)
                .testTag("label")
                .semantics {
                    contentDescription = "label"
                },
            fontSize = 14.sp,
        )
    }
    Spacer(modifier = Modifier.height(80.dp))
}
Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(bottom = 16.dp),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Bottom,
) {
    Button(
        onClick = onStartStopTrackingClick,
        modifier = Modifier
            .padding(horizontal = 16.dp)
    ) {
        Text(
            text = if
(mainViewModel.isLocationTrackingForegroundActive) "Stop tracking" else "Start
tracking",
            color = Color.Black
        )
        Spacer(modifier = Modifier.size(ButtonDefaults.IconSpacing))
        Image(
            painter = painterResource(id = if
(mainViewModel.isLocationTrackingForegroundActive) R.drawable.ic_pause else
R.drawable.ic_play),
            contentDescription = if
(mainViewModel.isLocationTrackingForegroundActive) "stop_tracking" else
"start_tracking"
        )
    }
}
```

```

    }
}

@Composable
fun MapView(mapReadyCallback: OnMapReadyCallback) {
    AndroidView(
        factory = { context ->
            val mapView = org.maplibre.android.maps.MapView(context)
            mapView.onCreate(null)
            mapView.getMapAsync(mapReadyCallback)
            mapView
        },
    )
}

```

4. Fügen Sie Code zum Erstellen eines LocationTracker Objekts hinzu und starten und beenden Sie das Tracking.
 - a. Fügen Sie der `MainViewModel.kt` Datei den folgenden Code hinzu.

```

...
var isLocationTrackingForegroundActive: Boolean by mutableStateOf(false)
var locationTracker: LocationTracker? = null

```

- b. Fügen Sie nun folgenden Code in die Datei `MainActivity.kt` ein:

```

// ...other imports
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
import software.amazon.location.tracking.LocationTracker
import software.amazon.location.tracking.aws.LocationTrackingCallback
import software.amazon.location.tracking.config.LocationTrackerConfig
import software.amazon.location.tracking.database.LocationEntry
import software.amazon.location.tracking.filters.DistanceLocationFilter
import software.amazon.location.tracking.filters.TimeLocationFilter
import software.amazon.location.tracking.util.TrackingSdkLogLevel

class MainActivity : ComponentActivity(), OnMapReadyCallback,
    MapLibreMap.OnCameraMoveStartedListener, MapLibreMap.OnCameraIdleListener {

    private val mainViewModel: MainViewModel by viewModels()
    private val poolId = "YOUR_AWS_IDENTITY_POOL_ID"
    private val trackerName = "YOUR_AWS_TRACKER_NAME"

```

```

private val region = "YOUR_AWS_REGION"
private val mapName = "YOUR_AWS_MAP_NAME"
private val apiKey = "YOUR_AWS_API_KEY"
private val coroutineScope = MainScope()
private lateinit var locationCredentialsProvider:
LocationCredentialsProvider
private lateinit var authHelper: AuthHelper

override fun onCreate(savedInstanceState: Bundle?) {
    MapLibre.getInstance(this)
    super.onCreate(savedInstanceState)
    setContent {
        TestMapAppTheme {
            Surface(
                modifier = Modifier.fillMaxSize(),
                color = MaterialTheme.colorScheme.background
            ) {
                MapLoadScreen(this, mainViewModel, onStartStopTrackingClick
= {
                    if (mainViewModel.isLocationTrackingForegroundActive) {
                        mainViewModel.isLocationTrackingForegroundActive =
false
                        mainViewModel.locationTracker?.stop()
                    } else {
                        if (checkLocationPermission(this))
return@MapLoadScreen
                            mainViewModel.isLocationTrackingForegroundActive =
true

mainViewModel.locationTracker?.start(locationTrackingCallback = object :
LocationTrackingCallback {
onLocationAvailabilityChanged(locationAvailable: Boolean) {
}

                                override fun onLocationReceived(location:
LocationEntry) {
}

                                override fun onUploadSkipped(entries:
LocationEntry) {
}

```



```
        override fun onUploadStarted(entries:
List<LocationEntry>) {
            }

        override fun onUploaded(entries:
List<LocationEntry>) {
            }
    })
}
})
}
}
}
authenticateUser()
}

private fun authenticateUser() {
    coroutineScope.launch {
        authHelper = AuthHelper(applicationContext)
        locationCredentialsProvider =
authHelper.authenticateWithCognitoIdentityPool(
            poolId,
        )
        locationCredentialsProvider.let {
            val config = LocationTrackerConfig(
                trackerName = trackerName,
                logLevel = TrackingSdkLogLevel.DEBUG,
                latency = 1000,
                frequency = 5000,
                waitForAccurateLocation = false,
                minUpdateIntervalMillis = 5000,
            )
            mainViewModel.locationTracker = LocationTracker(
                applicationContext,
                it,
                config,
            )

            mainViewModel.locationTracker?.enableFilter(TimeLocationFilter())

            mainViewModel.locationTracker?.enableFilter(DistanceLocationFilter())
        }
    }
}
```

```
    }

    private fun checkLocationPermission(context: Context) =
    ActivityCompat.checkSelfPermission(
        context,
        Manifest.permission.ACCESS_FINE_LOCATION,
    ) != PackageManager.PERMISSION_GRANTED &&
    ActivityCompat.checkSelfPermission(
        context,
        Manifest.permission.ACCESS_COARSE_LOCATION,
    ) != PackageManager.PERMISSION_GRANTED

    override fun onMapReady(map: MapLibreMap) {
        map.setStyle(
            Style.Builder()
                .fromUri(
                    "https://maps.geo.$region.amazonaws.com/maps/v0/maps/
$mapName/style-descriptor?key=$apiKey"
                ),
        ) {
            mainViewModel.mapLibreMap = map
            map.uiSettings.isAttributionEnabled = true
            map.uiSettings.isLogoEnabled = false
            map.uiSettings.attributionGravity = Gravity.BOTTOM or Gravity.END
            val initialPosition = LatLng(47.6160281982247, -122.32642111977668)
            map.cameraPosition = CameraPosition.Builder()
                .target(initialPosition)
                .zoom(14.0)
                .build()

            map.addOnCameraMoveStartedListener(this)
            map.addOnCameraIdleListener(this)
            map.cameraPosition.target?.let { latLng ->
                mainViewModel.reverseGeocode(
                    LatLng(
                        latLng.latitude,
                        latLng.longitude
                    ), apiKey
                )
            }
        }
    }

    override fun onCameraMoveStarted(p0: Int) {
```

```
        mainViewModel.label = ""
        mainViewModel.isLabelAdded = false
    }

    override fun onCameraIdle() {
        if (!mainViewModel.isLabelAdded) {
            mainViewModel.mapLibreMap?.cameraPosition?.target?.let { latLng ->
                mainViewModel.reverseGeocode(
                    LatLng(
                        latLng.latitude,
                        latLng.longitude
                    ), apiKey
                )
            }
        }
    }
}
```

Der obige Code zeigt, wie Sie ein `LocationTracker` Objekt mit erstellen `AuthHelper` und wie Sie das Tracking mit starten und beenden können `LocationTracker`.

- `authenticateUser()`: Diese Methode erstellt `AuthHelper` `LocationTracker` Objekte.
- `onStartStopTrackingClick`: Dieser Callback wird ausgelöst, wenn der Benutzer auf die Schaltfläche „Tracking starten/beenden“ klickt, wodurch das Tracking mit dem Tracking SDK gestartet/beendet wird.

5. Erstellen Sie eine Testroute mit dem Android Emulator.

- Öffnen Sie den Emulator, indem Sie den AVD mit Android Studio starten.
- Öffnen Sie Extended Controls, indem Sie in der Emulator-Symbolleiste auf das Symbol Mehr (drei Punkte) klicken.
- Öffnen Sie den Standort, indem Sie in der Seitenleiste Standort auswählen.
- Erstelle eine Route mit GPX-Daten oder indem du auf die Karte klickst und Quell- und Zieldaten auswählst.
- Starten Sie die Simulation, indem Sie auf ROUTE ABSPIELEN klicken, um mit der Simulation der GPS-Route zu beginnen.
- Testen Sie die Anwendung, indem Sie Ihre Anwendung ausführen und beobachten, wie sie mit der simulierten Route umgeht.

Dies ist die vollständige Demo der Android Quick Start-Anwendung.

Was kommt als Nächstes

Der Quellcode für diese Anwendung ist verfügbar unter [GitHub](#).

Um mehr aus Amazon Location herauszuholen, können Sie sich die folgenden Ressourcen ansehen:

- Tauchen Sie tiefer in die [Konzepte von Amazon Location Service](#) ein
- Erfahren Sie mehr darüber, [wie Sie die Funktionen und Funktionen von Amazon Location nutzen](#) können
- Sehen Sie sich [anhand von Codebeispielen mit Amazon Location](#) an, wie Sie dieses Beispiel erweitern und komplexere Anwendungen erstellen können.

Eine iOS-App erstellen

In diesem Abschnitt erstellen Sie eine iOS-Anwendung mit der Möglichkeit, an einem Standort zu suchen und im Vordergrund zu verfolgen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen und eine Amazon Cognito Cognito-Identität für Ihre Anwendung.

Themen

- [Amazon-Standortressourcen für Ihre App erstellen](#)
- [Richten Sie die Authentifizierung für Ihre Anwendung ein](#)
- [Erstellen der iOS-Basisanwendung](#)
- [Den ursprünglichen Code einrichten](#)
- [Fügen Sie Ihrer Anwendung eine interaktive Karte hinzu](#)
- [Suchfunktion zu Ihrer Anwendung hinzufügen](#)
- [Tracking zu Ihrer Anwendung hinzufügen](#)
- [Was kommt als Nächstes](#)


Amazon-Standortressourcen für Ihre App erstellen

Wenn Sie sie noch nicht haben, müssen Sie die Amazon-Standortressourcen erstellen, die Ihre Anwendung verwenden wird. Sie erstellen eine Kartenressource, um Karten in Ihrer Anwendung

anzuzeigen, einen Ortsindex, um nach Orten auf der Karte zu suchen, und einen Tracker, um ein Objekt auf der Karte zu verfolgen.

Um Ihrer Anwendung Standortressourcen hinzuzufügen

1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen ihn, wenn Sie Ihre Skriptdatei später im Tutorial erstellen.
 - c. Wählen Sie eine Karte aus.

 Note

Durch die Auswahl eines Kartenstils wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Ressourcen verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Lieferfahrzeuge oder Mitarbeiter, dürfen Sie HERE nur als Ihren Geolocation-Anbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen von Amazon Location zu und wählen Sie dann Create map aus. Sie können mit der Karte, die Sie ausgewählt haben, interagieren: vergrößern, verkleinern oder in eine beliebige Richtung schwenken.
 - e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie werden ihn später in diesem Tutorial verwenden, um die richtige Authentifizierung zu erstellen.
2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen ihn, wenn Sie Ihre Skriptdatei später im Tutorial erstellen.
 - c. Wählen Sie einen Datenanbieter.

Note

Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem Kartenanbieter entspricht, den Sie bereits ausgewählt haben. Auf diese Weise können Sie sicherstellen, dass die Suchanfragen mit den Karten übereinstimmen. Wenn Ihre Anwendung Ressourcen verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Lieferfahrzeuge oder Mitarbeiter, dürfen Sie HERE nur als Ihren Geolocation-Anbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher. In diesem Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie „Nein, nur zur einmaligen Verwendung“ wählen können.
 - e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon-Standorte zu und wählen Sie dann Ortsindex erstellen.
 - f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie werden ihn verwenden, um im nächsten Abschnitt dieses Tutorials die richtige Authentifizierung zu erstellen.
3. Um einen Tracker mit der Amazon Location-Konsole zu erstellen.
- a. Öffnen Sie die [Amazon Location Service Service-Konsole](#).
 - b. Wählen Sie im linken Navigationsbereich Trackers aus.
 - c. Wählen Sie Tracker erstellen aus.
 - d. Füllen Sie alle erforderlichen Felder aus.
 - e. Unter Positionsfilterung empfehlen wir Ihnen, die Standardeinstellung zu verwenden: TimeBased.
 - f. Wählen Sie Tracker erstellen, um den Vorgang abzuschließen.

Richten Sie die Authentifizierung für Ihre Anwendung ein

Die Anwendung, die Sie in diesem Tutorial erstellen, wird anonym verwendet, was bedeutet, dass sich Ihre Benutzer nicht anmelden müssen AWS, um die Anwendung zu verwenden. Für die Verwendung der Amazon Location Service APIs ist jedoch eine Authentifizierung erforderlich. Sie verwenden Amazon Cognito, um anonyme Benutzer zu authentifizieren und zu autorisieren. In diesem Tutorial wird Amazon Cognito verwendet, um Ihre Anwendung zu authentifizieren.

Note

Weitere Informationen zur Verwendung von Amazon Cognito mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Die folgenden Tutorials zeigen Ihnen, wie Sie die Authentifizierung für die Karte, den Ortsindex und den Tracker einrichten, in denen Sie erstellt haben, sowie wie Sie Berechtigungen für Amazon Location einrichten.

Erstellen Sie eine IAM-Richtlinie für die Nachverfolgung

1. Melden Sie sich bei der IAM-Konsole unter <https://console.aws.amazon.com/iam/> mit Ihrem Benutzer an, der über Administratorrechte verfügt.
2. Wählen Sie im Navigationsbereich Policies.
3. Wählen Sie im Inhaltsbereich die Option Create policy (Richtlinie erstellen).
4. Wählen Sie die JSON-Option, kopieren Sie dann diese JSON-Richtlinie und fügen Sie sie in das JSON-Textfeld ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapStyleDescriptor",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:SearchPlaceIndexForPosition",
        "geo:GetDevicePositionHistory",
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:{Region}:{Account}:map/{MapName}",
        "arn:aws:geo:{Region}:{Account}:place-index/{IndexName}",
        "arn:aws:geo:{Region}:{Account}:tracker/{TrackerName}"
      ]
    }
  ]
}
```

```
}
```

Dies ist ein Richtlinienbeispiel für Tracking. Um das Beispiel für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die TrackerNamePlatzhalter Region AccountIndexName,, MapName und.

Note

Nicht authentifizierte Identitätspools sind zwar dafür vorgesehen, auf ungesicherten Websites veröffentlicht zu werden, beachten Sie jedoch, dass sie gegen standardmäßige, zeitlich begrenzte Anmeldeinformationen ausgetauscht werden. AWS Es ist wichtig, die IAM-Rollen, die mit nicht authentifzierten Identitätspools verknüpft sind, angemessen abzugrenzen. Weitere Informationen zur Verwendung und angemessenen Festlegung von Richtlinien in Amazon Cognito mit Amazon Location Service finden Sie unter [Zugriff auf Amazon Location Service gewähren](#).

5. Geben Sie auf der Seite Überprüfen und erstellen einen Namen für das Feld Richtlinienname ein. Überprüfen Sie die durch Ihre Richtlinie gewährten Berechtigungen und wählen Sie dann Richtlinie erstellen aus, um Ihre Arbeit zu speichern.

Die neue Richtlinie wird in der Liste der verwalteten Richtlinien angezeigt und ist bereit.

Richten Sie die Authentifizierung für Ihr Tracking ein

1. Richten Sie die Authentifizierung für Ihre Kartenanwendung in der [Amazon Cognito Cognito-Konsole](#) ein.
2. Öffnen Sie die Seite Identitätspools.

Note

Der Pool, den Sie erstellen, muss sich in demselben AWS Konto und derselben AWS Region befinden wie die Amazon Location Service Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

3. Wählen Sie Identitätspool erstellen.
4. Beginnen Sie mit dem Vertrauensschritt Identitätspool konfigurieren. Wählen Sie für die Benutzerzugriffsauthentifizierung die Option Gastzugriff aus und klicken Sie auf Weiter.

5. Wählen Sie auf der Seite Berechtigungen konfigurieren die Option Bestehende IAM-Rolle verwenden aus und geben Sie den Namen der IAM-Rolle ein, die Sie im vorherigen Schritt erstellt haben. Wenn Sie bereit sind, klicken Sie auf Weiter, um mit dem nächsten Schritt fortzufahren.
6. Geben Sie auf der Seite Eigenschaften konfigurieren einen Namen für Ihren Identitätspool ein. Drücken Sie dann auf Weiter.
7. Überprüfen Sie auf der Seite Überprüfen und erstellen alle vorhandenen Informationen und klicken Sie dann auf Identitätspool erstellen.
8. Öffnen Sie die Seite Identitätspools und wählen Sie den Identitätspool aus, den Sie gerade erstellt haben. Kopieren Sie dann den IdentityPoolId , den Sie später verwenden werden, oder notieren Sie ihn in Ihrem Browser-Skript.

Erstellen der iOS-Basisanwendung

In diesem Tutorial erstellen Sie eine iOS-Anwendung, die eine Karte einbettet und es dem Benutzer ermöglicht, herauszufinden, was sich an einem Ort auf der Karte befindet.

Lassen Sie uns zunächst eine Swift-Anwendung mit dem Projektassistenten von Xcode erstellen.

Um eine leere Anwendung (Xcode) zu erstellen

1. Öffnen Sie Xcode und wählen Sie im Menü Datei, Neu, Neues Projekt.
2. Wählen Sie auf der Registerkarte iOS App und dann Weiter aus.
3. Geben Sie einen Produktnamen und eine Organisations-ID ein und geben Sie im Feld Schnittstelle eine Eingabe einSwiftUI. Wählen Sie Weiter, um die Auswahl abzuschließen.
4. Wählen Sie einen Ort aus, an dem Sie Ihr Projekt speichern möchten, und klicken Sie auf die Schaltfläche Erstellen, um die leere Anwendung zu erstellen.

Sobald Sie die Basisanwendung erstellt haben, müssen Sie die erforderlichen Pakete für die Beispiel-App installieren.

Installation der erforderlichen Abhängigkeiten

1. Klicken Sie in Xcode mit der rechten Maustaste auf das Projekt und wählen Sie Pakete hinzufügen... . Dadurch wird das Fenster Pakete geöffnet, in dem Sie Ihrem Projekt Pakete hinzufügen können.

2. Fügen Sie im Fenster Pakete die folgenden Pakete hinzu:
 - [Verwenden Sie für das native Maplibre-Paket diese URL: https://github.com/maplibre/maplibre-gl-native-distribution](https://github.com/maplibre/maplibre-gl-native-distribution) Fügen Sie der URL die folgenden Pakete hinzu:maplibre-gl-native-distribution, und. Mapbox
 - Verwenden Sie für das Amazon Location Authentication iOS SDK diese URL: [https://github.com/aws-geospatial/ amazon-location-mobile-auth -sdk-ios](https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios). Fügen Sie über die URL die folgenden Pakete hinzu:amazon-location-mobile-auth-sdk-ios, und. AmazonLocationiOSAuthSDK
 - Verwenden Sie für das Amazon Location Tracking iOS SDK diese URL: [https://github.com/ aws-geospatial/ amazon-location-mobile-tracking -sdk-ios](https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios). Fügen Sie über die URL die folgenden Pakete hinzu:amazon-location-mobile-tracking-sdk-ios, und. AmazonLocationiOSTrackingSDK

Den ursprünglichen Code einrichten

Aktivieren Sie Standortberechtigungen in Ihrer App

1. Öffnen Sie Ihr Xcode-Projekt.
2. Suchen Sie die Info.plist Datei des Projekts.
3. Fügen Sie die erforderlichen Schlüssel für Standortberechtigungen hinzu, die den Anforderungen Ihrer App entsprechen. Hier sind die Schlüssel:
 - `NSLocationWhenInUseUsageDescription`: Beschreibung, warum Ihre App Standortzugriff benötigt, wenn sie verwendet wird.
 - `NSLocationAlwaysAndWhenInUseUsageDescription`: Beschreibung, warum Ihre App kontinuierlichen Standortzugriff benötigt.

Jetzt müssen Sie die Ressourcenwerte in Ihrer App konfigurieren. Fügen Sie eine neue Datei mit dem Namen hinzu `Config.xcconfig` und geben Sie die Werte ein, die Sie zuvor in der Amazon-Konsole erstellt haben.

```
REGION =  
INDEX_NAME =  
MAP_NAME =  
IDENTITY_POOL_ID =
```

```
TRACKER_NAME =
```

1. Wählen Sie im Navigationsbereich auf der linken Seite das Projekt aus.
2. Wählen Sie im Bereich Ziele Ihre App aus und klicken Sie auf den Info-Tab.
3. Fügen Sie Informationseigenschaften mit Werten wie den folgenden hinzu:
4. Fügen Sie die `Config.swift` Datei mit dem folgenden Inhalt hinzu, die die Konfigurationswerte aus der Bundle-Infodatei liest.

```
import Foundation

enum Config {
    static let region = Bundle.main.object(forKey: "Region") as!
    String
    static let mapName = Bundle.main.object(forKey: "MapName") as!
    String
    static let indexName = Bundle.main.object(forKey: "IndexName")
    as! String
    static let identityPoolId = Bundle.main.object(forKey:
    "IdentityPoolId") as! String
    static let trackerName = Bundle.main.object(forKey:
    "TrackerName") as! String
}
```

5. Erstellen Sie einen neuen Ordner mit dem Namen `ViewModel` und fügen Sie eine `TrackingViewModel.swift` Datei darin hinzu.

```
import SwiftUI
import AmazonLocationiOSAuthSDK
import MapLibre

final class TrackingViewModel : ObservableObject {
    @Published var trackingButtonText = NSLocalizedString("StartTrackingLabel",
    comment: "")
    @Published var trackingButtonColor = Color.blue
    @Published var trackingButtonIcon = "play.circle"
    @Published var region : String
    @Published var mapName : String
    @Published var indexName : String
    @Published var identityPoolId : String
    @Published var trackerName : String
    @Published var showAlert = false
}
```

```
@Published var alertTitle = ""
@Published var alertMessage = ""
@Published var centerLabel = ""

var clientIntialised: Bool
var client: LocationTracker!
var authHelper: AuthHelper
var credentialsProvider: LocationCredentialsProvider?
var mlnMapView: MLNMapView?
var mapViewDelegate: MapViewDelegate?
var lastGetTrackingTime: Date?
var trackingActive: Bool

init(region: String, mapName: String, indexName: String, identityPoolId:
String, trackerName: String) {
    self.region = region
    self.mapName = mapName
    self.indexName = indexName
    self.identityPoolId = identityPoolId
    self.trackerName = trackerName
    self.authHelper = AuthHelper()
    self.trackingActive = false
    self.clientIntialised = false
}

func authWithCognito(identityPoolId: String?) {
    guard let identityPoolId =
identityPoolId?.trimmingCharacters(in: .whitespacesAndNewlines)
    else {
        alertTitle = NSLocalizedString("Error", comment: "")
        alertMessage = NSLocalizedString("NotAllFieldsAreConfigured", comment:
"")
        showAlert = true
        return
    }
    credentialsProvider =
authHelper.authenticateWithCognitoUserPool(identityPoolId: identityPoolId)
    initializeClient()
}

func initializeClient() {
    client = LocationTracker(provider: credentialsProvider!, trackerName:
trackerName)
    clientIntialised = true
}
```

```
}  
}
```

Fügen Sie Ihrer Anwendung eine interaktive Karte hinzu

Sie werden nun das Kartensteuerelement zu Ihrer Anwendung hinzufügen. In diesem Tutorial wird die AWS API zur Verwaltung der Kartenansicht in der Anwendung verwendet MapLibre . Das Kartensteuerelement selbst ist Teil der [MapLibre GL Native iOS-Bibliothek](#).

1. Fügen Sie dem Ordner Views eine `MapView.swift` Datei mit dem folgenden Code hinzu:

```
import SwiftUI  
import MapLibre  
  
struct MapView: UIViewRepresentable {  
    var onMapViewAvailable: ((MLNMapView) -> Void)?  
    var mlnMapView: MLNMapView?  
    var trackingViewModel: TrackingViewModel  
  
    func makeCoordinator() -> MapView.Coordinator {  
        return Coordinator(self, trackingViewModel: trackingViewModel)  
    }  
  
    func makeUIView(context: Context) -> MLNMapView {  
        let styleURL = URL(string: "https://maps.geo.  
\(trackingViewModel.region).amazonaws.com/maps/v0/maps/  
\(trackingViewModel.mapName)/style-descriptor")  
        let mapView = MLNMapView(frame: .zero, styleURL: styleURL)  
        mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]  
        mapView.setZoomLevel(15, animated: true)  
        mapView.showsUserLocation = true  
        mapView.userTrackingMode = .follow  
        context.coordinator.mlnMapView = mapView  
        mapView.delegate = context.coordinator  
  
        mapView.logoView.isHidden = true  
        context.coordinator.addCenterMarker()  
  
        onMapViewAvailable?(mapView)  
        trackingViewModel.mlnMapView = mapView  
        return mapView  
    }  
}
```

```
func updateUIView(_ uiView: MLNMapView, context: Context) {
}

class Coordinator: NSObject, MLNMapViewDelegate, MapViewDelegate {
    var control: MapView
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel
    var centerMarker: MLNPointAnnotation?

    public init(_ control: MapView, trackingViewModel: TrackingViewModel) {
        self.control = control
        self.trackingViewModel = trackingViewModel
        super.init()
        self.trackingViewModel.mapViewDelegate = self
    }

    func mapViewDidFinishRenderingMap(_ mapView: MLNMapView, fullyRendered:
Bool) {
        if(fullyRendered) {
            mapView.accessibilityIdentifier = "MapView"
            mapView.isAccessibilityElement = false
        }
    }

    func addCenterMarker() {
        guard let mlnMapView = mlnMapView else {
            return
        }

        let centerCoordinate = mlnMapView.centerCoordinate
        let marker = MLNPointAnnotation()
        marker.coordinate = centerCoordinate
        marker.accessibilityLabel = "CenterMarker"
        mlnMapView.addAnnotation(marker)
        centerMarker = marker

        trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
    }

    func mapView(_ mapView: MLNMapView, regionDidChangeAnimated animated: Bool)
{
        if let marker = centerMarker {
```



```
struct UserLocationView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        Button(action: {
            trackingViewModel.locateMe()
        }) {
            Image(systemName: "scope")
                .resizable()
                .frame(width: 24, height: 24)
                .padding(5)
                .background(Color.white)
                .foregroundColor(.blue)
                .clipShape(RoundedRectangle(cornerRadius: 8))
                .shadow(color: Color.black.opacity(0.3), radius: 3, x: 0, y: 2)
        }
        .accessibility(identifier: "LocateMeButton")
        .padding(.trailing, 10)
        .padding(.bottom, 10)
        .frame(maxWidth: .infinity, alignment: .trailing)
    }
}
```

4. Fügen Sie die `TrackingView.swift` Datei mit dem folgenden Code hinzu:

```
import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            MapView(trackingViewModel: trackingViewModel)
            VStack {
                UserLocationView(trackingViewModel: trackingViewModel)
            }
        }
        .onAppear() {
            if !trackingViewModel.identityPoolId.isEmpty {
                trackingViewModel.authWithCognito(identityPoolId:
                trackingViewModel.identityPoolId)
            }
        }
    }
}
```



```
}
```

Sie können die Anwendung jetzt erstellen. Um es auszuführen, müssen Sie möglicherweise ein Gerät einrichten, um es in Xcode zu emulieren, oder die App auf Ihrem Gerät verwenden. Verwenden Sie diese App, um zu sehen, wie sich das Kartensteuerelement verhält. Sie können die Karte schwenken, indem Sie auf der Karte ziehen und die Maus zusammendrücken, um sie zu vergrößern. Sie können die Funktionsweise des Kartensteuerelements selbst ändern, um es an die Bedürfnisse Ihrer Anwendung anzupassen.

Suchfunktion zu Ihrer Anwendung hinzufügen

Jetzt fügen Sie der Anwendung die umgekehrte Geokodierungssuche hinzu, in der Sie die Elemente an einem Ort finden. Um die Verwendung einer iOS-App zu vereinfachen, suchen wir in der Mitte des Bildschirms. Um einen neuen Standort zu finden, verschieben Sie die Karte an die Stelle, an der Sie suchen möchten. Wir platzieren eine Markierung in der Mitte der Karte, um zu zeigen, wo wir suchen.

1. Fügen Sie den folgenden Code in die Datei `TrackingViewModel.swift` ein, der sich auf die Suche nach umgekehrter Geokodierung bezieht

```
func reverseGeocodeCenter(centerCoordinate: CLLocationCoordinate2D, marker:
    MLNPointAnnotation) {
    let position = [NSNumber(value: centerCoordinate.longitude), NSNumber(value:
        centerCoordinate.latitude)]
    searchPositionAPI(position: position, marker: marker)
}

func searchPositionAPI(position: [Double], marker: MLNPointAnnotation) {
    if let amazonClient = authHelper.getLocationClient() {
        Task {
            let searchRequest = SearchPlaceIndexForPositionInput(indexName:
                indexName, language: "en" , maxResults: 10, position: position)
            let searchResponse = try? await amazonClient.searchPosition(indexName:
                indexName, input: searchRequest)
            DispatchQueue.main.async {
                self.centerLabel = searchResponse?.results?.first?.place?.label ??
                ""
                self.mlnMapView?.selectAnnotation(marker, animated: true,
                    completionHandler: {})
            }
        }
    }
}
```

```
    }  
}
```

2. Aktualisieren Sie die `TrackingView.swift` Datei mit dem folgenden Code, der die Adresse der zentrierten Position der Kartenansicht anzeigt

```
import SwiftUI  
  
struct TrackingView: View {  
    @ObservedObject var trackingViewModel: TrackingViewModel  
    var body: some View {  
        ZStack(alignment: .bottom) {  
            if trackingViewModel.mapSigningInitialised {  
                MapView(trackingViewModel: trackingViewModel)  
                VStack {  
                    UserLocationView(trackingViewModel: trackingViewModel)  
                    CenterAddressView(trackingViewModel: trackingViewModel)  
                }  
            }  
            else {  
                Text("Loading...")  
            }  
        }  
        .onAppear() {  
            if !trackingViewModel.identityPoolId.isEmpty {  
                Task {  
                    do {  
                        try await trackingViewModel.authWithCognito(identityPoolId:  
trackingViewModel.identityPoolId)  
                    }  
                    catch {  
                        print(error)  
                    }  
                }  
            }  
        }  
    }  
}
```

Tracking zu Ihrer Anwendung hinzufügen

Der letzte Schritt für Ihre Anwendung besteht darin, Ihrer App Tracking-Funktionen hinzuzufügen. In diesem Fall fügen Sie Ihrer App Tracking-Punkte zum Starten und Stoppen sowie zum Abrufen und Anzeigen von Tracker-Punkten hinzu.

1. Fügen Sie die `TrackingBottomView.swift` Datei zu Ihrem Projekt hinzu. Welches hat eine Schaltfläche, die die Verfolgung von Benutzerstandorten startet und stoppt und Tracking-Punkte auf der Karte anzeigt.

```
import SwiftUI

struct TrackingBottomView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        Button(action: {
            Task {
                if(trackingViewModel.trackingButtonText ==
NSLocalizedString("StartTrackingLabel", comment: "")) {
                    trackingViewModel.startTracking()
                } else {
                    trackingViewModel.stopTracking()
                }
            }
        }) {
            HStack {
                Spacer()
                Text("Tracking")
                    .foregroundColor(trackingViewModel.trackingButtonColor)
                    .background(.white)
                    .cornerRadius(15.0)

                Image(systemName: trackingViewModel.trackingButtonIcon)
                    .resizable()
                    .frame(width: 24, height: 24)
                    .padding(5)
                    .background(.white)
                    .foregroundColor(trackingViewModel.trackingButtonColor)
            }
        }
        .accessibility(identifier: "TrackingButton")
    }
}
```

```
        .background(.white)
        .clipShape(RoundedRectangle(cornerRadius: 8))
        .padding(.trailing, 10)
        .padding(.bottom, 40)
        .frame(width: 130, alignment: .trailing)
        .shadow(color: Color.black.opacity(0.3), radius: 3, x: 0, y: 2)
    }
}
```

2. Aktualisiere die `TrackingView.swift` Datei mit dem folgenden Code

```
import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            if trackingViewModel.mapSigningInitialised {
                MapView(trackingViewModel: trackingViewModel)
                VStack {
                    UserLocationView(trackingViewModel: trackingViewModel)
                    CenterAddressView(trackingViewModel: trackingViewModel)
                    TrackingBottomView(trackingViewModel: trackingViewModel)
                }
            }
            else {
                Text("Loading...")
            }
        }
        .onAppear() {
            if !trackingViewModel.identityPoolId.isEmpty {
                Task {
                    do {
                        try await trackingViewModel.authWithCognito(identityPoolId:
trackingViewModel.identityPoolId)
                    }
                    catch {
                        print(error)
                    }
                }
            }
        }
    }
}
```

```
}
```

3. Fügen Sie der `TrackingViewModel.swift` Datei den folgenden Code hinzu. Diese Funktionen sind für das Starten und Stoppen des Trackings verantwortlich. Außerdem wird eine Fehlermeldung angezeigt, wenn die Standortberechtigung des Benutzers verweigert wird.
4. Um das Tracking im Vordergrund zu implementieren, kopieren Sie das folgende Codebeispiel:

```
func showLocationDeniedRationale() {
    alertTitle = NSLocalizedString("locationManagerAlertTitle", comment: "")
    alertMessage = NSLocalizedString("locationManagerAlertText", comment: "")
    showAlert = true
}

// Required in info.plist: Privacy - Location When In Use Usage Description
func startTracking() {
    do {
        print("Tracking Started...")
        if(client == nil) {
            initializeClient()
        }
        try client.startTracking()
        DispatchQueue.main.async { [self] in
            self.trackingButtonText = NSLocalizedString("StopTrackingLabel",
comment: "")
            self.trackingButtonColor = .red
            self.trackingButtonIcon = "pause.circle"
            trackingActive = true
        }
    } catch TrackingLocationError.permissionDenied {
        showLocationDeniedRationale()
    } catch {
        print("error in tracking")
    }
}

func stopTracking() {
    print("Tracking Stopped...")
    client.stopTracking()
    trackingButtonText = NSLocalizedString("StartTrackingLabel", comment: "")
    trackingButtonColor = .blue
    trackingButtonIcon = "play.circle"
    trackingActive = false
}
```

}

Note

Der `startTracking` fragt den Benutzer nach der Erlaubnis, sich dort aufzuhalten. Die Anwendung muss die Berechtigungen „Wann verwendet“ oder „Nur einmal“ verwenden. Andernfalls gibt die Anwendung den Fehler „Zugriff verweigert“ aus.

Gehen Sie wie folgt vor, um Tracking-Standorte abzurufen und anzuzeigen:

1. Um die Standorte vom Gerät des Benutzers abzurufen, müssen Sie das Start- und Enddatum sowie die Uhrzeit angeben. Ein einziger Aufruf gibt maximal 100 Tracking-Standorte zurück, aber wenn es mehr als 100 Tracking-Standorte gibt, wird ein `nextToken`-Wert zurückgegeben. Sie müssen nachfolgende `getTrackerDevice`Location``-Aufrufe mit `nextToken` aufrufen, um weitere Trackingpunkte für die angegebene Start- und Endzeit zu laden.

```
func getTrackingPoints(nextToken: String? = nil) async throws {
    guard trackingActive else {
        return
    }
    // Initialize startTime to 24 hours ago from the current date and time.
    let startTime: Date = Date().addingTimeInterval(-86400)
    var endTime: Date = Date()
    if lastGetTrackingTime != nil {
        endTime = lastGetTrackingTime!
    }
    let result = try await client?.getTrackerDeviceLocation(nextToken:
nextToken, startTime: startTime, endTime: endTime)
    if let trackingData = result {

        lastGetTrackingTime = Date()
        let devicePositions = trackingData.devicePositions

        let positions = devicePositions!.sorted { (pos1:
LocationClientTypes.DevicePosition, pos2: LocationClientTypes.DevicePosition) ->
Bool in
            guard let date1 = pos1.sampleTime,
                  let date2 = pos2.sampleTime else {
                return false
            }
    }
}
```

```

        return date1 < date2
    }

    let trackingPoints = positions.compactMap { position ->
CLLocationCoordinate2D? in
        guard let latitude = position.position!.last, let longitude =
position.position!.first else {
            return nil
        }
        return CLLocationCoordinate2D(latitude: latitude, longitude:
longitude)
    }
    DispatchQueue.main.async {
        self.mapViewDelegate!.drawTrackingPoints( trackingPoints:
trackingPoints)
    }
    if let nextToken = trackingData.nextToken {
        try await getTrackingPoints(nextToken: nextToken)
    }
}
}
}

```

2. Ersetzen Sie nun den Code in der `MapView.swift` Datei durch den folgenden Code:

```

import SwiftUI
import MapLibre

struct MapView: UIViewRepresentable {
    var onMapViewAvailable: ((MLNMapView) -> Void)?
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel

    func makeCoordinator() -> MapView.Coordinator {
        return Coordinator(self, trackingViewModel: trackingViewModel)
    }

    func makeUIView(context: Context) -> MLNMapView {
        let styleURL = URL(string: "https://maps.geo.
\\(trackingViewModel.region).amazonaws.com/maps/v0/maps/
\\(trackingViewModel.mapName)/style-descriptor")
        let mapView = MLNMapView(frame: .zero, styleURL: styleURL)
        mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
        mapView.setZoomLevel(15, animated: true)
        mapView.showsUserLocation = true
    }
}

```

```
mapView.userTrackingMode = .follow
context.coordinator.mlnMapView = mapView
mapView.delegate = context.coordinator

mapView.logoView.isHidden = true
context.coordinator.addCenterMarker()

onMapViewAvailable?(mapView)
trackingViewModel.mlnMapView = mapView
return mapView
}

func updateUIView(_ uiView: MLNMapView, context: Context) {
}

class Coordinator: NSObject, MLNMapViewDelegate, MapViewDelegate {
    var control: MapView
    var mlnMapView: MLNMapView?
    var trackingViewModel: TrackingViewModel
    var centerMarker: MLNPointAnnotation?

    public init(_ control: MapView, trackingViewModel: TrackingViewModel) {
        self.control = control
        self.trackingViewModel = trackingViewModel
        super.init()
        self.trackingViewModel.mapViewDelegate = self
    }

    func mapViewDidFinishRenderingMap(_ mapView: MLNMapView, fullyRendered:
Bool) {
        if(fullyRendered) {
            mapView.accessibilityIdentifier = "MapView"
            mapView.isAccessibilityElement = false
        }
    }

    func addCenterMarker() {
        guard let mlnMapView = mlnMapView else {
            return
        }

        let centerCoordinate = mlnMapView.centerCoordinate
        let marker = MLNPointAnnotation()
        marker.coordinate = centerCoordinate
    }
}
```



```
        marker.accessibilityLabel = "CenterMarker"
        mlnMapView.addAnnotation(marker)
        centerMarker = marker

        trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
    }

    func mapView(_ mapView: MLNMapView, regionDidChangeAnimated animated: Bool)
    {
        if let marker = centerMarker {
            DispatchQueue.main.asyncAfter(deadline: .now() + 1.0) {
                mapView.deselectAnnotation(marker, animated: false)
                marker.coordinate = mapView.centerCoordinate
                let centerCoordinate = mapView.centerCoordinate
                self.trackingViewModel.reverseGeocodeCenter(centerCoordinate:
centerCoordinate, marker: marker)
            }
        }
    }

    func mapView(_ mapView: MLNMapView, viewFor annotation: MLNAnnotation) ->
MLNAnnotationView? {
        guard let pointAnnotation = annotation as? MLNPointAnnotation else {
            return nil
        }

        let reuseIdentifier: String
        var color: UIColor = .black
        if pointAnnotation.accessibilityLabel == "Tracking" {
            reuseIdentifier = "TrackingAnnotation"
            color = UIColor(red: 0.00784313725, green: 0.50588235294, blue:
0.58039215686, alpha: 1)
        } else if pointAnnotation.accessibilityLabel == "LocationChange" {
            reuseIdentifier = "LocationChange"
            color = .gray
        } else {
            reuseIdentifier = "DefaultAnnotationView"
        }

        var annotationView =
mapView.dequeueReusableAnnotationView(withIdentifier: reuseIdentifier)

        if annotationView == nil {
```

```

        if reuseIdentifier != "DefaultAnnotationView" {
            annotationView = MLNAnnotationView(annotation: annotation,
reuseIdentifier: reuseIdentifier)
            //If point annotation is an uploaded Tracking point the radius
is 20 and color is blue, otherwise radius is 10 and color is gray
            let radius = pointAnnotation.accessibilityLabel == "Tracking" ?
20:10

            annotationView?.frame = CGRect(x: 0, y: 0, width: radius,
height: radius)

            annotationView?.backgroundColor = color
            annotationView?.layer.cornerRadius = 10

            if pointAnnotation.accessibilityLabel == "Tracking" {
                annotationView?.layer.borderColor = UIColor.white.cgColor
                annotationView?.layer.borderWidth = 2.0
                annotationView?.layer.shadowColor = UIColor.black.cgColor
                annotationView?.layer.shadowOffset = CGSize(width: 0,
height: 2)

                annotationView?.layer.shadowRadius = 3
                annotationView?.layer.shadowOpacity = 0.2
                annotationView?.clipsToBounds = false
            }
        }
        else {
            return nil
        }
    }

    return annotationView
}

func mapView(_ mapView: MLNMapView, didUpdate userLocation:
MLNUserLocation?) {
    if (userLocation?.location) != nil {
        if trackingViewModel.trackingActive {
            let point = MLNPointAnnotation()
            point.coordinate = (userLocation?.location!.coordinate)!
            point.accessibilityLabel = "LocationChange"
            mapView.addAnnotation(point)
            Task {
                do {
                    try await trackingViewModel.getTrackingPoints()
                }
                catch {

```

```

        print(error)
    }
}
}

func checkIfTrackingAnnotationExists(on mapView: MLNMapView, at
coordinates: CLLocationCoordinate2D) -> Bool {
    let existingAnnotation = mapView.annotations?.first(where: { annotation
in
        guard let annotation = annotation as? MLNPointAnnotation else
{ return false }
        return annotation.coordinate.latitude == coordinates.latitude &&
        annotation.coordinate.longitude == coordinates.longitude &&
annotation.accessibilityLabel == "Tracking" })
    return existingAnnotation != nil
}

public func drawTrackingPoints(trackingPoints: [CLLocationCoordinate2D]?) {
    guard let mapView = mlnMapView, let newTrackingPoints =
trackingPoints, !newTrackingPoints.isEmpty else {
        return
    }

    let uniqueCoordinates = newTrackingPoints.filter { coordinate in
        !checkIfTrackingAnnotationExists(on: mapView, at: coordinate)
    }

    let points = uniqueCoordinates.map { coordinate -> MLNPointAnnotation
in
        let point = MLNPointAnnotation()
        point.coordinate = coordinate
        point.accessibilityLabel = "Tracking"
        return point
    }
    mapView.addAnnotations(points)
}
}

protocol MapViewDelegate: AnyObject {
    func drawTrackingPoints(trackingPoints: [CLLocationCoordinate2D]?)

```

```
}
```

Gehen Sie wie folgt vor, um Zeichenkettenwerte zu lokalisieren.

1. Erstellen und fügen Sie eine neue Datei mit dem Namen `Localizable.xcstrings` hinzu.
2. Klicken Sie mit der rechten Maustaste auf die `Localizable.xcstrings` Datei und öffnen Sie sie als Quellcode.
3. Ersetzen Sie den Inhalt durch Folgendes:

```
{
  "sourceLanguage" : "en",
  "strings" : {
    "Cancel" : {
      "extractionState" : "manual",
      "localizations" : {
        "en" : {
          "stringUnit" : {
            "state" : "translated",
            "value" : "Cancel"
          }
        }
      }
    },
    "Error" : {
      "extractionState" : "manual",
      "localizations" : {
        "en" : {
          "stringUnit" : {
            "state" : "translated",
            "value" : "Error"
          }
        }
      }
    },
    "Loading..." : {

  },
  "locationManagerAlertText" : {
    "extractionState" : "manual",
    "localizations" : {
      "en" : {
```

```
    "stringUnit" : {
      "state" : "translated",
      "value" : "Allow \\\\"Quick Start App\\" to use your location"
    }
  }
},
"locationManagerAlertTitle" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "We need your location to detect your location in map"
      }
    }
  }
},
"NotAllFieldsAreConfigured" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "Not all the fields are configured"
      }
    }
  }
},
"OK" : {
  "extractionState" : "manual",
  "localizations" : {
    "en" : {
      "stringUnit" : {
        "state" : "translated",
        "value" : "OK"
      }
    }
  }
},
"StartTrackingLabel" : {
  "localizations" : {
    "en" : {
      "stringUnit" : {
```

```
        "state" : "translated",
        "value" : "Start Tracking"
    }
}
},
"StopTrackingLabel" : {
    "localizations" : {
        "en" : {
            "stringUnit" : {
                "state" : "translated",
                "value" : "Stop Tracking"
            }
        }
    }
},
"Tracking" : {

}
},
"version" : "1.0"
}
```

4. Speichern Sie Ihre Dateien und erstellen Sie Ihre App und führen Sie sie aus, um eine Vorschau der Funktionen anzuzeigen.
5. Erlauben Sie die Standortgenehmigung und tippen Sie auf die Tracking-Schaltfläche. Die App beginnt mit dem Hochladen von Benutzerstandorten und lädt sie auf den Amazon Location Tracker hoch. Außerdem werden Standortänderungen der Benutzer, Tracking-Punkte und die aktuelle Adresse auf der Karte angezeigt.

Ihre Schnellstart-Bewerbung ist abgeschlossen. Dieses Tutorial hat Ihnen gezeigt, wie Sie eine iOS-Anwendung erstellen, die:

- Erstellt eine Karte, mit der Benutzer interagieren können.
- Behandelt mehrere Kartenereignisse, die damit verbunden sind, dass der Benutzer die Kartenansicht ändert.
- Ruft eine Amazon Location Service API auf, um die Karte an einem Standort mithilfe der `searchByPosition` API von Amazon Location zu durchsuchen.

Was kommt als Nächstes

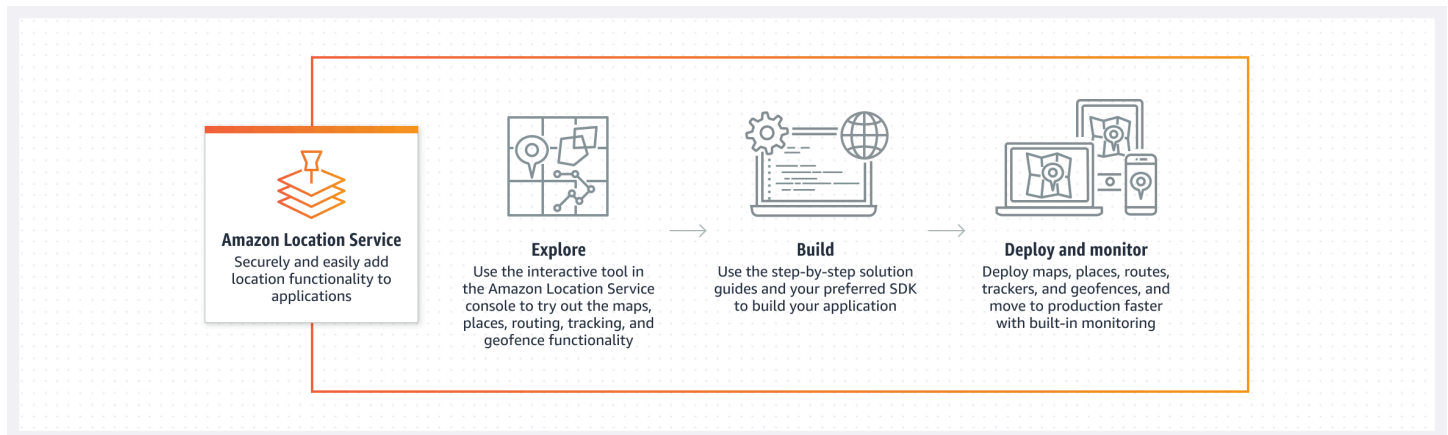
Der Quellcode für diese Anwendung ist verfügbar unter [GitHub](#).

Um mehr aus Amazon Location herauszuholen, können Sie sich die folgenden Ressourcen ansehen:

- Tauchen Sie tiefer in die [Konzepte von Amazon Location Service](#) ein
- Erfahren Sie mehr darüber, [wie Sie die Funktionen und Funktionen von Amazon Location nutzen](#) können
- Sehen Sie sich [anhand von Codebeispielen mit Amazon Location an, wie Sie dieses Beispiel](#) erweitern und komplexere Anwendungen erstellen können

Konzepte von Amazon Location Service

Mit Amazon Location Service können Sie Ihrer Anwendung auf sichere Weise Standortdaten hinzufügen. Erkunden Sie einige der Funktionen mithilfe des [visuellen und interaktiven Tools](#), das in der Amazon Location-Konsole verfügbar ist. Mit dem Tool „Erkunden“ können Sie eine Standardkarte bearbeiten, nach Sonderzielen suchen, Geofences um interessante Bereiche ziehen und das Senden von Gerätestandorten an einen Tracker simulieren.



Wenn Sie bereit sind zu bauen, erstellen Sie Ihre Ressourcen und wählen Sie aus einer Vielzahl von Kartenstilen und Datenanbietern. Anschließend können Sie das SDK installieren, das zu Ihrer Entwicklungsumgebung passt, und die Amazon Location APIs gemäß den Anweisungen in diesem Handbuch verwenden. Darüber hinaus können Sie die Überwachung mithilfe von Amazon CloudWatch und integrieren AWS CloudTrail.

Die Themen in diesem Abschnitt bieten Ihnen einen Überblick über die Kernkonzepte von Amazon Location und bereiten Sie darauf vor, in Ihren eigenen Anwendungen mit dem Thema Standort zu arbeiten.

Themen

- [Überblick über den Amazon-Standort](#)
- [Zuordnungen](#)
- [Suche nach Orten](#)
- [Routen](#)
- [Geofences und Tracker](#)
- [Häufige Anwendungsfälle für die Nutzung von Amazon Location Service](#)

- [Was ist ein Datenanbieter?](#)
- [Amazon-Standortregionen und Endpunkte](#)
- [Kontingente für Amazon Location Service](#)

Überblick über den Amazon-Standort

Amazon Location Service bietet über AWS Ressourcen Zugriff auf standortbezogene Funktionen und Datenanbieter. Amazon Location bietet fünf Arten von AWS Ressourcen, abhängig von der Art der Funktionalität, die Sie benötigen. Verwenden Sie die verschiedenen Ressourcen zusammen, um eine vollständige standortbasierte Anwendung zu erstellen. Sie können eine oder mehrere dieser Ressourcen mithilfe der Amazon Location-Konsole, der Amazon Location-APIs oder der SDKs erstellen.

Jede Ressource definiert den zugrunde liegenden [Datenanbieter](#), der verwendet werden soll (falls zutreffend), und bietet Zugriff auf Funktionen, die sich auf ihren Typ beziehen.

Beispielsweise:

- Mit [Amazon Location Service Maps](#) können Sie eine Karte von einem Kartenanbieter auswählen, die Sie in Ihrer Mobil- oder Webanwendung verwenden möchten.
- Mit [Amazon Location Service Places](#) können Sie einen Datenanbieter für die Suche nach Sehenswürdigkeiten, das Ausfüllen von Textteilen, die Geokodierung und die umgekehrte Geokodierung auswählen.
- Mit [Amazon Location Service Routes](#) können Sie einen Datenanbieter auswählen und Routen finden und die Reisezeit anhand von up-to-date Straßen- und Live-Verkehrsinformationen schätzen.
- Mit [Amazon Location Service Geofences](#) können Sie Interessengebiete als virtuelle Grenze definieren. Anschließend können Sie Standorte anhand dieser Daten auswerten und Benachrichtigungen über Eingangs- und Austrittsereignisse erhalten.
- [Amazon Location Service Tracker](#) erhalten Standortaktualisierungen von Ihren Geräten. Sie können Tracker mit Geofence-Sammlungen verknüpfen, sodass alle Positionsaktualisierungen automatisch mit Ihren Geofences abgeglichen werden.

Sie können IAM-Richtlinien verwenden, um den Zugriff auf Ihre Amazon-Standortressourcen zu verwalten und zu autorisieren. Sie können Ihre Ressourcen auch in Ressourcengruppen

organisieren, um Aufgaben zu verwalten und zu automatisieren, wenn Ihre Anzahl an Ressourcen wächst. Weitere Informationen zur Verwaltung von AWS Ressourcen finden Sie unter [Was sind AWS-Ressourcengruppen?](#) Im AWS-Ressourcengruppen-Benutzerhandbuch.

Der Standort wird anhand von Breiten- und Längengradkoordinaten definiert, die dem [World Geodetic System \(WGS 84\)](#) entsprechen, das üblicherweise als Standardkoordinatenreferenzsystem für GPS-Dienste (Global Positioning System) verwendet wird.

In den folgenden Abschnitten wird beschrieben, wie die Komponenten von Amazon Location funktionieren.

Zuordnungen

Die Kartenressource von Amazon Location Service bietet Ihnen Zugriff auf die zugrunde liegenden Grundkartendaten für eine Karte. Sie verwenden die Kartenressource mit einer Karten-Rendering-Bibliothek, um Ihrer Anwendung eine interaktive Karte hinzuzufügen. Sie können Ihrer Karte weitere Funktionen hinzufügen, wie Markierungen (oder Stecknadeln), Routen und Polygonflächen, je nach Bedarf für Ihre Anwendung.

Note

Informationen zur praktischen Verwendung von Kartenressourcen finden Sie unter [Verwenden von Amazon Location Maps in Ihrer Anwendung](#).

Im Folgenden finden Sie eine Übersicht über die Erstellung und Verwendung von Kartenressourcen:

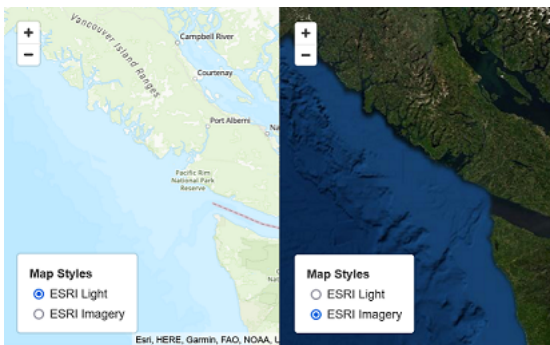


1. Sie erstellen eine Kartenressource in Ihrem AWS Konto, indem Sie einen Kartenstil aus einem Datenprovider auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das zu Ihrer Entwicklungsumgebung und Ihren Anwendungen passt. Weitere Informationen zu den verfügbaren Optionen finden Sie im Thema [Zugriff auf den Amazon-Standort](#).

- Um eine Map in Ihrer Anwendung anzuzeigen, kombinieren Sie eine Kartenressource mit einer Rendering-Bibliothek wie Amplify MapLibre, oder Tangram. Weitere Informationen finden Sie unter [Verwenden von Karten in diesem Handbuch](#).
- Anschließend können Sie die Überwachung mithilfe von Diensten wie Amazon CloudWatch und AWS CloudTrail Amazon Location integrieren. Weitere Informationen finden Sie unter [Überwachen von Amazon Location Service mit Amazon CloudWatch](#) und [Protokollierung und Überwachung mit AWS CloudTrail](#).

Kartenstile

Wenn Sie eine Kartenressource erstellen, müssen Sie einen Kartenstil für diese Ressource auswählen. Kartenstile definieren das Aussehen der gerenderten Karte. Die folgende Abbildung zeigt beispielsweise denselben Datenanbieter mit zwei verschiedenen Stilen aus verschiedenen Kartenressourcen in Amazon Location. Ein Stil ist ein typischer Straßenstil, der auf den Vektordaten in der Karte basiert. Der andere umfasst Rasterdaten, die Satellitenbilder zeigen. Der Stil kann sich ändern, wenn Sie die Karte vergrößern oder verkleinern, aber in der Regel haben Stile ein einheitliches Thema. Es ist möglich, Teile oder alle Stilinformationen zu überschreiben, bevor sie an die Karten-Rendering-Bibliothek übergeben werden.



Politische Ansichten

Bestimmte Kartenstile in Amazon Location Service unterstützen zusätzliche politische Ansichten.

Note

Die politische Ansicht muss in Übereinstimmung mit den geltenden Gesetzen verwendet werden, einschließlich der Gesetze zur Kartierung des Landes oder der Region, in dem die Karten, Bilder und andere Daten sowie Inhalte Dritter, auf die Sie über Amazon Location Service zugreifen, zur Verfügung gestellt werden.

Die folgenden Kartenstile unterstützen eine politische Sichtweise Indiens (IND).

- [Kartenstile von Esri](#):
 - Esri Navigation
 - Esri Licht
 - Esri Straßenkarte
 - Dunkelgraue Leinwand von Esri
 - Hellgraue Leinwand von Esri
- [Stile für offene Datenkarten](#):
 - Open Data Standard Light
 - Offener Datenstandard Dunkel
 - Öffnen Sie die Datenvisualisierung Light
 - Öffnen Sie Data Visualization Dark

In der Amazon Location Service Service-Konsole können Sie die angezeigten Stile filtern, um nur die Stile anzuzeigen, die die politische Sichtweise Indiens unterstützen.

Benutzerspezifische Layers

Eine benutzerdefinierte Ebene ist eine zusätzliche Ebene, die Sie für einen Kartenstil aktivieren können. Derzeit unterstützt nur der VectorEsriNavigation Kartenstil die POI benutzerdefinierte Ebene.

Wenn Sie den POI benutzerdefinierten Layer aktivieren, fügt er Ihrer Karte eine umfangreichere Auswahl an Orten hinzu, z. B. Geschäfte, Dienstleistungen, Restaurants, Attraktionen und andere Sehenswürdigkeiten. Standardmäßig ist unset die benutzerdefinierte Ebene. Weitere Informationen finden Sie [MapConfiguration](#) in der Referenz zur Location-API.

Rendern von Karten

Um eine Karte in Ihrer Anwendung zu rendern, verwenden Sie in der Regel eine Bibliothek zum Rendern von Karten. Es gibt mehrere gängige Optionen, die Bibliotheken verwenden können:

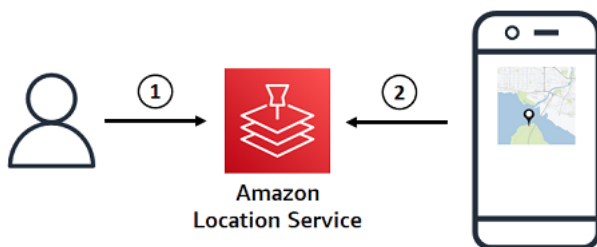
- MapLibre— MapLibre ist eine Open-Source-Bibliothek speziell für das Rendern interaktiver Karten und die bevorzugte Methode zum Rendern von Karten von Amazon Location Service. MapLibre beinhaltet die Möglichkeit, Raster- und Vektordaten aus einer Datenquelle (z. B. einer Amazon

Location-Kartenressource) zu rendern. Sie können die Karte erweitern MapLibre , um Ihre eigenen Daten auf der Karte zu zeichnen.

- **Amplify** — Amplify ist ein Open-Source-Framework zum Erstellen von Anwendungen für das Web, iOS, Android und mehr. Wenn Ihre Anwendung Amplify verwendet, können Sie sie um die Amazon Location-Funktionalität erweitern. Amplify enthält Bibliotheken speziell für die Erstellung standortbasierter Anwendungen auf Amazon, einschließlich des Renderns von Karten. Amplify verwendet, MapLibre um die Karte zu rendern, bietet jedoch zusätzliche Funktionen, die spezifisch für Amazon Location Service sind, um die Nutzung effizienter zu gestalten, und fügt auch Such- und andere Funktionen hinzu.
- **Tangram** — Tangram ist eine alternative Open-Source-Bibliothek, die interaktive Karten rendert, ähnlich wie. MapLibre

Die Karten-Rendering-Bibliothek ruft zur Laufzeit Daten von Amazon Location Service ab und rendert die Kartendaten auf der Grundlage der ausgewählten Kartenressource. Die Kartenressource definiert den Datenanbieter und den Kartenstil, die verwendet werden.

Die folgende Abbildung zeigt, wie die Kartenressource in Amazon Location Service zusammen mit einer Karten-Rendering-Bibliothek verwendet wird, um die endgültige Karte zu erstellen.



1. Sie erstellen eine Kartenressource in Amazon Location Service, indem Sie das AWS Management Console oder verwenden AWS CLI. Dadurch werden der Datenanbieter und der Kartenstil definiert, den Sie verwenden möchten.
2. Ihre Anwendung enthält eine Bibliothek zum Rendern von Karten. Sie geben der Karten-Rendering-Bibliothek den Namen der Kartenressource, die verwendet werden soll. Die Karten-Rendering-Bibliothek ruft Daten und Stilinformationen für diese Kartenressource von Amazon Location ab und rendert die Karte auf dem Bildschirm.

Terminologie der Karten

Ressource für Karten

Ermöglicht den Zugriff auf Kartendaten eines ausgewählten Anbieters. Verwenden Sie die Kartenressource, um Kartenkacheln abzurufen, die Kartendaten und einen Style-Deskriptor enthalten, um anzugeben, wie Features auf einer Karte dargestellt werden.

Grundkarte

Stellt geographischen Kontext für Ihre Karte bereit, die als Vektorkachel-Layer gespeichert wird. Kachel-Layer enthalten geografischen Kontext wie Straßennamen, Gebäude und Landnutzung als visuelle Referenz.

Vektor

Vektordaten sind Formdaten, die aus Punkten, Linien und Polygonen bestehen. Sie werden häufig zum Speichern und Anzeigen von Straßen, Orten und Gebieten auf einer Karte verwendet. Eine Vektorform kann auch als Symbole für Markierungen auf einer Karte verwendet werden.

Raster

Rasterdaten sind Bilddaten, die aus einem Raster bestehen, in der Regel aus Farben. Sie werden häufig verwendet, um eine Darstellung kontinuierlicher Daten auf Karten zu speichern und anzuzeigen, z. B. Gelände-, Satelliten- oder Heatmaps. Rasterbilder können auch als Bilder oder Symbole verwendet werden.

Kartenstil

Vektordaten enthalten grundsätzlich keine Informationen darüber, wie die Datenschichten gezeichnet werden müssen, um die endgültige Karte zu erstellen. Ein Kartenstil definiert Farb- und andere Stilinformationen für die Daten, um zu definieren, wie sie beim Rendern aussehen. Kartenressourcen enthalten Stilinformationen für die Karte.

Amazon Location Service bietet Styles, die der [Mapbox GL-Stilspezifikation entsprechen](#).

Vektor-Kachel

Ein Kachelformat, das Kartendaten mithilfe von Vektorformen speichert. Diese Daten führen zu einer Karte, die sich an die Bildschirmauflösung anpassen lässt und Features auf verschiedene Arten selektiv rendern kann, wobei für eine optimale Leistung eine geringe Dateigröße beibehalten wird.

Unterstütztes Vektor-Dateiformat: Mapbox Vector Tiles (MVT).

Glyph-Datei

Eine Binärdatei mit codierten Unicode-Zeichen. Wird von einem Karten-Renderer zur Anzeige von Beschriftungen verwendet.

Sprite-Datei

Eine PNG-Bilddatei (Portable Network Graphic), die kleine Rasterbilder mit Ortsbeschreibungen in einer JSON-Datei enthält. Wird von einem Karten-Renderer verwendet, um Symbole oder Texturen auf einer Karte zu rendern.

Suche nach Orten

Eine wichtige Funktion von Amazon Location Service ist die Möglichkeit, die Geolokalisierungsinformationen zu durchsuchen. Amazon Location stellt diese Funktionalität über die Place-Index-Ressource bereit.

Note

Informationen zur Verwendung von Ortsindex-Ressourcen für die Suche in der Praxis finden Sie unter [Suchen von Orts- und Geolocation-Daten mit Amazon Location](#).

Sie können die Ortsindex-APIs verwenden, um nach Folgendem zu suchen:

- Sehenswürdigkeiten wie Restaurants und Sehenswürdigkeiten. Suchen Sie nach Namen und optionalem Standort, um sich umzusehen, und Sie erhalten eine nach Relevanz geordnete Liste mit Optionen.

- Eine Straßenadresse, die einen Breiten- und Längengrad für diese Adresse erhält. Dies wird als Geokodierung bezeichnet.
- Eine Breitengrad- und Längengradposition, an die die zugehörige Straßenadresse oder andere Informationen über den Standort gesendet werden. Dies wird als umgekehrte Geokodierung bezeichnet.
- Eine unvollständige oder falsch geschriebene Freiform-Textabfrage, typischerweise während der Eingabe durch den Benutzer. Dies wird als automatische Vervollständigung, Autosuggestion oder Fuzzy-Matching bezeichnet.

Der Ortsindex beinhaltet, welcher Datenanbieter für die Suche verwendet werden soll.

Note

Kartendaten und andere Geolokalisierungsinformationen, einschließlich genauer Standorte, können je nach Datenanbieter variieren. Es hat sich bewährt, denselben Datenanbieter für Ihren Ortsindex, Ihre Karte und andere Amazon-Standortressourcen zu verwenden. Wenn beispielsweise die von Ihrem Ortsindex zurückgegebenen Orte nicht mit der Position derselben Orte übereinstimmen, die in Ihrer Kartenressource angegeben wurden, können Sie eine Markierung an einer Stelle platzieren, die auf der Karte falsch zu sein scheint.

Im Folgenden wird gezeigt, wie Sie Ortsindex-Ressourcen erstellen und verwenden:



1. Zunächst erstellen Sie eine Ortsindex-Ressource in Ihrem AWS Konto, indem Sie einen Datenanbieter auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das zu Ihrer Entwicklungsumgebung und Ihren Anwendungen passt. Weitere Informationen zu den verfügbaren Optionen finden Sie im Thema [Zugriff auf den Amazon-Standort](#).
3. Beginnen Sie mit der Verwendung der Amazon Location Places-APIs. Weitere Informationen finden Sie im Thema zur Verwendung der [Places-Suche](#).

4. Anschließend können Sie die Überwachung mithilfe von Diensten wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Konzepte zur Geokodierung

Ein Ortsindex von Amazon bietet eine Aktion namens [SearchPlaceIndexForText](#), mit der Sie den zu durchsuchenden Text angeben können. Sie können beispielsweise nach Folgendem suchen:

- Orte — eine Suche nach einem Ort **Paris** könnte den Standort der Stadt in Frankreich zurückgeben.
- Unternehmen, nach denen gesucht wird, **coffee shop** könnte eine Liste von Coffeeshops mit ihren Namen und Standorten zurückgeben. Sie können auch einen Ort angeben, in dem Sie suchen möchten, oder ein Begrenzungsfeld, in dem Sie suchen möchten, um die Ergebnisse relevanter zu gestalten. In diesem Fall würde die Angabe eines Standorts in der Innenstadt von Seattle, Washington, zu Coffeeshops in dieser Gegend führen.
- Adressen — eine Suche nach einer Adresse **1600 Pennsylvania Ave, Washington D.C.** könnte den Standort des Weißen Hauses in den Vereinigten Staaten (der sich an dieser Adresse befindet) zurückgeben.

Die Suche nach Text auf diese Weise wird allgemein als Geokodierung bezeichnet. Dabei wird nach einem geografischen Standort für die Adresse oder den Ort gesucht.

Amazon Location Service bietet auch eine umgekehrte Geokodierungsaktion namens [SearchPlaceIndexForPosition](#). Dabei wird ein geografischer Standort verwendet und die Adresse, das Unternehmen oder andere Informationen darüber zurückgegeben, was sich an diesem Standort befindet.

Suchergebnisse

Wenn Sie eine erfolgreiche Suchanfrage in Amazon Location Service stellen, werden ein oder mehrere Ergebnisse zurückgegeben. Jedes Ergebnis enthält eine Bezeichnung, die dem Namen oder der Beschreibung des Ergebnisses entspricht. Beispielsweise könnte eine Suche nach ein Ergebnis mit der Bezeichnung zurückgeben **coffee shop**, das Ihnen mitteilt **Hometown Cafe**, dass ein Café mit dem Namen „Hometown Cafe“ gefunden wurde. Das Suchergebnis enthält in der Regel auch eine

strukturierte Adresse (mit Eigenschaften wie Adressnummer, Wohneinheit, Straße und Postleitzahl). Je nach Datenanbieter enthält es auch andere Metadaten, z. B. das Land und die Zeitzone.

Bei einer Suche nach einem Unternehmensnamen oder einer Kategorie (z. B. **coffee shop**) möchten Sie möglicherweise alle zurückgegebenen Ergebnisse auf einer Karte anzeigen. Bei einer Adresssuche möchten Sie vielleicht nur das erste Ergebnis automatisch verwenden. Informationen zur Relevanz finden Sie im nächsten Thema.

Vielfältige Ergebnisse und Relevanz

Bei der Suche nach Text findet Amazon Location Service oft mehr als ein einziges Ergebnis. Eine Suche nach **Paris** kann beispielsweise die Stadt in Frankreich, aber auch die Stadt in Texas zurückgeben. Die Ergebnisse sind nach der Relevanz sortiert, die vom Datenanbieter festgelegt wurde.

Note

Die Ergebnisse werden von allen Anbietern in der Reihenfolge ihrer Relevanz zurückgegeben. Wenn Sie Esri oder Grab als Datenanbieter wählen, enthalten die Ergebnisse einen Relevanzwert, anhand dessen Sie die relative Relevanz zwischen den Ergebnissen einer einzelnen Anfrage ermitteln können.

Wenn Sie zusätzliche Informationen angeben, z. B. einen Ländernamen oder einen Ort, an dem gesucht werden soll, kann sich die Reihenfolge der Ergebnisse ändern, die Anzahl der Ergebnisse reduzieren oder sogar die Menge der zurückgegebenen Ergebnisse ändern. Beispielsweise ist es wahrscheinlicher, dass **Paris** bei einer Suche nach einem Standort in Texas das erste Ergebnis zurückgeben `Paris, Texas` wird als `Paris, France`.

In einer interaktiven Anwendung können Sie anhand der Relevanz entscheiden, ob Sie das beste Ergebnis akzeptieren möchten, oder einen Benutzer bitten, zwischen mehreren zurückgegebenen Ergebnissen zu unterscheiden. Wenn das erste Ergebnis eine hohe Relevanz hat, können Sie es einfach als die richtige Antwort akzeptieren. Wenn es mehrere Ergebnisse mit hoher Relevanz oder keine Ergebnisse mit hoher Relevanz gibt, sollten Sie die Ergebnisse auflisten und den Benutzer das beste Ergebnis auswählen lassen.

Ergebnisse adressieren

Mit derselben [SearchPlaceIndexForText](#)Aktion können Sie mit Amazon Location Service nach Adressen suchen. Je mehr Informationen Sie angeben, desto wahrscheinlicher ist es, dass die zurückgegebene Adresse mit der angegebenen übereinstimmt. Zum Beispiel **123 Main St** ist es weniger wahrscheinlich, ein korrektes Ergebnis zu finden als **123 Main St, Anytown, California, 90210**.

Adressen haben mehrere Attribute, wie Hausnummer, Straße, Ort, Region, Postleitzahl usw. Diese Attribute werden verwendet, um eine Adresse im Ortsindex zu finden, die so vielen Aspekten wie möglich entspricht. Je mehr Attribute gefunden werden, desto relevanter wird die Übereinstimmung angesehen und desto wahrscheinlicher ist es, dass sie zurückgegeben wird.

Note

Die Relevanz von Adressenergebnissen hängt davon ab, wie genau das Ergebnis mit der Eingabe übereinstimmt. Dabei kann es sich um die Anzahl der übereinstimmenden Attribute handeln, aber auch darum, wie genau die Ergebnisse mit der Eingabe übereinstimmen. Beispielsweise **123 Main St** hätte eine Eingabe von einer höheren Relevanz, wenn sie in den Daten gefunden wird, als wenn sie das einzige Ergebnis **Main St** ist. **Main St** wird immer noch zurückgegeben, aber wahrscheinlich mit einem niedrigeren Relevanzwert.

Die Suchergebnisse enthalten eine Bezeichnung für die vollständige Adresse (**123 Main St, Anytown, California, 90210**), aber auch die einzelnen strukturierten Attribute der zurückgegebenen Adresse. Dies ist hilfreich, da Sie damit beispielsweise Adressfelder in einer Datenbank ausfüllen oder die Ergebnisse untersuchen und die Stadt, Region oder Postleitzahl des gefundenen Standorts ermitteln können.

Interpolation

Adressen in den Ortsindexdaten enthalten exakte Adressübereinstimmungen. Nehmen wir zum Beispiel an, dass es eine Straße gibt **9th street** und ein Block zwei Häuser hat, **220** und **240**, wie in der folgenden Abbildung.



Der Datenanbieter erstellt die Geolokalisierungsdaten mit diesen beiden bekannten Adressen. Sie können nach diesen beiden Adressen suchen und sie werden gefunden. Nachdem der Datenprovider die Kartendaten erstellt hat, nehmen wir an, dass zwischen den ersten beiden Adressen ein neues Haus hinzugefügt wird. Dieses neue Haus erhält die Adresse **230 S 9th St**. Wenn Sie nach **230 S 9th St** suchen, wird der Datenanbieter trotzdem ein Ergebnis finden. Anstatt eine bekannte Adresse zu verwenden, interpoliert er zwischen den bereits bekannten Adressen und schätzt anhand dieser Adressen die Position der neuen Adresse. In diesem Fall könnte davon ausgegangen werden, dass 230 auf halber Strecke zwischen 220 und 240 liegt (und auf derselben Straßenseite), und auf dieser Grundlage eine ungefähre Position zurückgeben.

Note

Datenanbieter aktualisieren ihre Geolokalisierungsdaten regelmäßig mit neuen Adressen. In diesem Fall **230 S 9th St** würden die Daten zu den Daten des Datenanbieters hinzugefügt, aber normalerweise wird es einen Zeitraum geben, in dem eine neue Adresse erstellt, aber noch nicht zu den Daten hinzugefügt wurde.

In diesem Fall kann der Datenanbieter nicht erkennen, ob die neue Adresse in der Welt existiert, da sie noch nicht in den Daten enthalten ist, gibt aber anhand der vorhandenen Informationen die bestmögliche Antwort. Dieses Ergebnis wird als interpoliert bezeichnet und kann vom Datenanbieter

in den Ergebnissen zurückgegeben werden. Wenn `interpolated` zurückgegeben wird `false`, handelt es sich um eine bekannte Adresse. Wenn es zurückkehrt `true`, handelt es sich um eine ungefähre Adresse. Wenn sie nicht zurückgegeben wird, hat der Datenanbieter keine Information darüber bereitgestellt, ob das Ergebnis aus der Interpolation stammt.

Important

Der Datenprovider kann auch interpolierte Ergebnisse für Adressen zurückgeben, die überhaupt nicht existieren. Wenn Sie in diesem Fall beispielsweise die Adresse eingeben **232 S 9th St**, würde der Anbieter diese nicht existierende Adresse finden und eine Position in der Nähe von 230, aber auf der 240-Seite zurückgeben. Interpolierte Adressen sind nützlich, um Sie an den richtigen Ort zu bringen. Beachten Sie jedoch, dass es sich dabei nicht um bekannte Adressen handelt.

Geokodierungsergebnisse speichern

Wenn Sie eine Ortsindex-Ressource erstellen, müssen Sie eine Datenspeicheroption (`IntendedUse` in der API aufgerufen) angeben. Dabei kann es sich entweder um einmalig verwendbare oder um gespeicherte Ergebnisse handeln. Hier werden Sie nach Ihrer beabsichtigten Verwendung der Ergebnisse gefragt. Wenn Sie die Ergebnisse speichern möchten (auch für Caching-Zwecke), müssen Sie die Speicheroption wählen, nicht die Option zur einmaligen Verwendung.

Note

Wenn Sie die Option „Gespeichert“ wählen (mit Ja, Ergebnisse werden in der Konsole gespeichert, oder mit Auswahl `storage` in der `CreatePlaceIndex` API), speichert Amazon Location Service die Ergebnisse nicht für Sie. Dies ist ein Hinweis darauf, dass Sie beabsichtigen, die Ergebnisse zu speichern.

Wenn Sie sich überlegen, wie Sie die Ergebnisse Ihrer Anfragen an Amazon Location Service verwenden werden, sollten Sie stets die geltenden [AWS Servicebedingungen](#) beachten.

Terminologie von Places

Platzieren Sie die Indexressource

Ermöglicht die Auswahl einer Datenquelle zur Unterstützung von Suchabfragen. Sie können beispielsweise nach Sehenswürdigkeiten, Adressen oder Koordinaten suchen. Wenn eine Suchabfrage an eine Ortsindexressource gesendet wird, wird sie mithilfe der konfigurierten Datenquelle der Ressource ausgeführt.

Geokodierung

Geokodierung ist der Prozess, bei dem eine Texteingabe im Ortsindex danach gesucht wird und Ergebnisse mit Position zurückgegeben werden.

Umgekehrte Geokodierung

Bei umgekehrter Geokodierung wird eine Position ermittelt und Informationen über diese Position aus dem Ortsindex zurückgegeben, z. B. die Adresse, die Stadt oder das Unternehmen an diesem Standort.

Relevanz

Relevanz gibt an, wie genau ein Ergebnis mit der Eingabe übereinstimmt. Es ist kein Maß für Richtigkeit.

Interpolation

Interpolation ist der Prozess, bei dem unbekannte Adressen gefunden werden, indem bekannte Adressen als Richtwerte verwendet werden.

ISO 3166-Ländercodes

Amazon Location Service Places verwendet [die Ländercodes der International Organization for Standardization \(ISO\) 3166](#), um auf Länder oder Regionen zu verweisen.

Verwenden Sie die [ISO-Online-Browserplattform](#), um den Code für ein bestimmtes Land oder eine bestimmte Region zu finden.

Routen

Dieser Abschnitt bietet einen Überblick über die Konzepte rund um das Routing mit Amazon Location Service.

Note

Informationen zur Verwendung von Routenressourcen in der Praxis finden Sie unter [Berechnen von Routen mit Amazon Location Service](#).

Ressourcen für den Routenrechner

Mithilfe der Ressourcen für den Routenrechner können Sie Routen ermitteln und die Reisezeit anhand des up-to-date Straßennetzes und der Live-Verkehrsinformationen Ihres ausgewählten Datenanbieters abschätzen.

Sie können die Routes APIs verwenden, um Funktionen zu erstellen, mit denen Ihre Anwendung die Reisezeit, Entfernung und Geometrie der Route zwischen zwei beliebigen Orten abfragen kann. Sie können die Routes API auch verwenden, um Reisezeit und Entfernung zwischen einer Reihe von Abfahrts- und Zielorten in einer einzigen Anfrage abzufragen, um eine Matrix zu berechnen.

Im Folgenden wird gezeigt, wie Sie eine Routenberechnungsressource erstellen und verwenden:



1. Zunächst erstellen Sie eine Routenberechnungsressource in Ihrem AWS Konto, indem Sie einen Datenanbieter auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das zu Ihrer Entwicklungsumgebung und Ihren Anwendungen passt.
3. Beginnen Sie mit der Verwendung der Amazon Location Routes APIs. Weitere Informationen zur Verwendung der Routing-APIs finden Sie im Thema unter [Berechnen von Routen mit Amazon Location Service](#).

4. Anschließend können Sie die Überwachung mithilfe von Diensten wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [Überwachen von Amazon Location Service mit Amazon CloudWatch](#) und [Protokollierung und Überwachung mit AWS CloudTrail](#).

Berechnung einer Route

Eine Ressource für den Routenrechner von Amazon Location bietet eine Aktion namens `CalculateRoute`, mit der Sie eine Route zwischen zwei geografischen Standorten (dem Abflug - und dem Zielort) erstellen können. Die berechnete Route umfasst die Geometrie zum Zeichnen der Route auf einer Karte sowie die Gesamtzeit und Entfernung der Route.

Verwendung von Wegpunkten

Wenn Sie Ihre Routenanfrage erstellen, können Sie der Route zusätzliche Wegpunkte hinzufügen. Dies sind Punkte zwischen dem Start- und dem Zielort, die als Stopps entlang der Route dienen. Die Route wird über jeden der angegebenen Wegpunkte berechnet. Die Route von einem Punkt in der Anfrage zum nächsten wird a Leg genannt. Jede Etappe umfasst eine Entfernung, eine Zeit und die Geometrie für diesen Teil der Route.

Note

Die Wegpunkte werden in der Reihenfolge geroutet, die in der Anfrage angegeben ist. Sie werden nicht für den kürzesten Pfad neu angeordnet. Informationen zur Suche nach dem kürzesten Pfad finden Sie im [Routen planen](#) Abschnitt.

Du kannst bis zu 25 Wegpunkte in eine einzige Anfrage aufnehmen, um eine Route zu berechnen.

Verkehr und Abfahrtszeit

Der Amazon Location Service berücksichtigt den Verkehr bei der Berechnung einer Route. Der Datenverkehr, der berücksichtigt wird, basiert auf der von Ihnen angegebenen Zeit. Sie können angeben, dass Sie jetzt abfahren möchten, oder Sie können eine bestimmte Uhrzeit angeben, zu der Sie abfahren möchten. Dies wirkt sich auf das Routenergebnis aus, da es an den Verkehr zur angegebenen Uhrzeit angepasst wird.

Note

Sie können die Ankunftszeit anhand der Abfahrtszeit und der Reaktionszeit der Route berechnen, um beispielsweise die Ankunft eines Fahrers abzuschätzen.

Wenn Sie möchten, dass Amazon Location den Verkehr nicht berücksichtigt, geben Sie keine Abfahrtszeit und auch keine Abfahrt jetzt an. Dadurch wird eine Route berechnet, die von den besten Verkehrsbedingungen für die Route ausgeht.

Optionen für den Reisemodus

Sie können den Reisemodus festlegen, wenn Sie eine Route mit Amazon Location Service berechnen. Der Standardreisemodus ist Auto, aber Sie können auch wahlweise „LKW“ oder „Zu Fuß“ wählen.

Wenn Sie entweder den Pkw- oder den Lkw-Modus angeben, können Sie auch zusätzliche Optionen angeben.

Für den Automodus können Sie angeben, dass Sie Mautstraßen oder Fähren vermeiden möchten. Dadurch wird versucht, Fähren und mautpflichtige Straßen zu vermeiden, sie werden aber trotzdem entlangfahren, wenn sie die einzige Möglichkeit sind, zum Ziel zu gelangen.

Im Lkw-Modus können Sie auch Fähren und Mautstraßen vermeiden, aber zusätzlich können Sie die Größe und das Gewicht des Lkw angeben, um Strecken zu vermeiden, auf denen der Lkw nicht Platz findet.

Routen planen

Sie können Amazon Location Service verwenden, um Eingaben für Ihre Routenplanungs- und Optimierungssoftware zu erstellen. Sie können Routenergebnisse, einschließlich Reisezeit und Reisedistanz, für Routen zwischen einer Reihe von Abflugpositionen und einer Reihe von Zielpositionen erstellen. Dies wird als Erstellen einer Routenmatrix bezeichnet.

Note

Es gibt viele verschiedene Szenarien, die Software zur Routenplanung und -optimierung lösen kann. Planungssoftware kann beispielsweise anhand von Zeiten und Entfernungen zwischen Punkten den kürzesten Weg berechnen, der an jedem Punkt stoppt, wodurch eine effiziente Route für einen einzelnen Fahrer bereitgestellt wird. Alternativ kann eine

Planungssoftware verwendet werden, um Stopps auf mehrere Lkw aufzuteilen und so die Effizienz der gesamten Flotte zu erhöhen, oder um sicherzustellen, dass jeder Kunde innerhalb des von ihm gewünschten Zeitrahmens besucht wird. Amazon Location stellt die Routing-Funktionen auf effiziente Weise bereit, damit die Planungssoftware ihre Aufgabe erledigen kann.

Wenn beispielsweise die Abflugpositionen A und B und die Zielpositionen X und Y angegeben sind, gibt Amazon Location Service die Reisezeit und die Reisedistanz für Routen von A nach X, A nach Y, B nach X und B nach Y zurück.

Wie bei der Berechnung einer einzelnen Route können Sie die Routen mit unterschiedlichen Transportmitteln, Vermeidungsmöglichkeiten und Verkehrsbedingungen berechnen. Sie können beispielsweise angeben, dass es sich bei dem Fahrzeug um einen Lkw mit einer Länge von 35 Fuß handelt, und die berechnete Route verwendet diese Einschränkungen, um die Reisezeit und die Reisedistanz zu bestimmen. Sie können keine Wegpunkte in eine Routenmatrixberechnung einbeziehen.

Die Anzahl der zurückgegebenen Ergebnisse (und der berechneten Routen) entspricht der Anzahl der Startpositionen multipliziert mit der Anzahl der Zielpositionen. Ihnen wird jede berechnete Route in Rechnung gestellt, nicht jede Anfrage an den Service. Eine Routenmatrix mit 10 Abfahrten und 10 Zielen wird also als 100 Routen abgerechnet.

Terminologie der Route

Ressource für den Routenrechner

Eine AWS Ressource, mit der Sie Reisezeit und Entfernung schätzen und Routen auf einer Karte mit Verkehrs- und Straßennetzdaten darstellen können, die von Ihrem ausgewählten Datenanbieter stammen.

Mithilfe von Ressourcen zur Routenberechnung berechnen Sie Routen für verschiedene Verkehrsmittel, Umwege und Verkehrsbedingungen.

Route

Eine Route enthält Details, die verwendet werden, wenn Sie einen Pfad von der Startposition, den Wegpunktpositionen und der Zielposition aus entlang fahren.

Beispiele für Details in einer Route sind:

- Die Entfernung von einer Position zu einer anderen Position.
- Die Zeit, die benötigt wird, um von einer Position zur nächsten Position zu gelangen.
- Die LineString Geometrie, die den Pfad der Route darstellt.

Weitere Informationen zu Routen finden Sie in der [Antwortsyntax für den CalculateRoute Vorgang](#) in der Amazon Location Service Routes API-Referenz.

Routenmatrix

Eine Liste von Routen, von einer Reihe von Abflugpositionen bis zu einer Reihe von Zielpositionen. Nützlich als Eingabe für Software zur Routenplanung oder -optimierung.

Weitere Informationen zur Berechnung einer Routenmatrix finden Sie unter der [Syntax für den CalculateRouteMatrix Vorgang](#) in der Amazon Location Service Routes API-Referenz.

LineString Geometrie

Eine Amazon Location-Route besteht aus einer oder mehreren Etappen (einer Route von einem Wegpunkt zu einem anderen innerhalb der Gesamtroute). Die Geometrie jedes Abschnitts ist eine Polylinie, die als Polylinie dargestellt wird. LineString A LineString ist eine geordnete Anordnung von Positionen, die verwendet werden kann, um eine Route auf einer Karte darzustellen.

Das Folgende ist ein Beispiel für eine LineString mit drei Punkten:

```
[  
  [-122.7565, 49.0021],  
  [-122.3394, 47.6159],  
  [-122.1082, 45.8371]  
]
```

Zwischenposition

Wegpunkte sind Zwischenpositionen, die als Stopps auf einer Route zwischen der Startposition und der Zielposition dienen. Die Reihenfolge der Zwischenstopps auf der Route folgt der Reihenfolge, in der Sie die Wegpunktpositionen in der Anfrage angeben.

Bein

Ein einzelnes Bein ist die Reise von einer Position zur anderen. Wenn sich die Positionen nicht auf einer Straße befinden, werden sie auf die nächstgelegene Straße verschoben. Die Anzahl der Etappen auf einer Route ist um eins geringer als die Gesamtzahl der Positionen.

Eine Route ohne Wegpunkte besteht aus einer einzigen Etappe von der Startposition bis zum Ziel. Eine Route mit einem Wegpunkt besteht aus 2 Etappen, von der Startposition zum Wegpunkt und dann vom Wegpunkt zum Ziel.

Schritt

Ein Schritt ist ein Unterabschnitt einer Etappe. Jeder Schritt enthält zusammenfassende Informationen für diesen Schritt in der Etappe.

Geofences und Tracker

Dieser Abschnitt bietet einen Überblick über die Konzepte der Arbeit mit Geofences und Trackern von Amazon Location Service. Geofences sind Polygongrenzen, mit denen Sie benachrichtigt werden können, wenn sich Geräte oder Positionen in die Bereiche hinein- oder herausbewegen. Tracker-Ressourcen werden verwendet, um Positionen von Geräten zu speichern und zu aktualisieren, wenn sie sich bewegen.

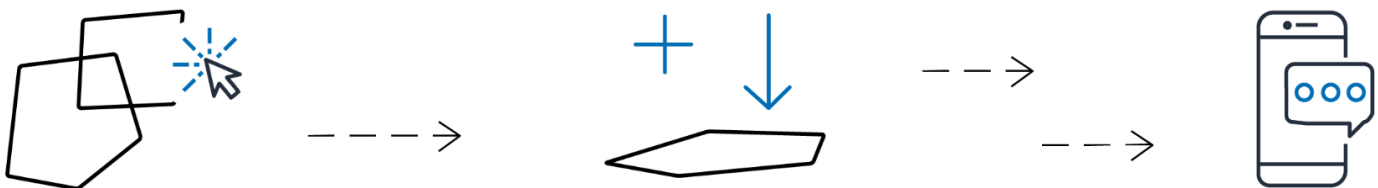
Note

Informationen zur praktischen Verwendung von Geofences und Trackern finden Sie unter [Geofencing eines Interessengebiets mithilfe von Amazon Location](#)

Geofences

Mit Ressourcen zur Erfassung von Geofences können Sie Geofences — virtuelle Grenzen — auf einer Karte speichern und verwalten. Sie können Standorte anhand einer Geofence-Sammlungsressource auswerten und Benachrichtigungen erhalten, wenn die Standortaktualisierung die Grenze eines der Geofences in der Geofence-Sammlung überschreitet.

Im Folgenden wird gezeigt, wie Sie Ressourcen zur Geofence-Erfassung erstellen und verwenden:



1. Erstellen Sie in Ihrem Konto eine Ressource zur Erfassung von Geofence. AWS

2. Fügen Sie dieser Sammlung Geofences hinzu. Sie können dies tun, indem Sie entweder das Geofence-Upload-Tool auf der Amazon Location-Konsole oder die Amazon Location Geofences-API verwenden. Weitere Informationen zu den verfügbaren Optionen finden Sie unter [Zugreifen auf den Amazon-Standort](#).

Geofences können entweder durch ein Polygon oder durch einen Kreis definiert werden. Verwenden Sie ein Polygon, um herauszufinden, wann ein Gerät einen bestimmten Bereich betritt. Verwenden Sie einen Kreis, um herauszufinden, wann sich ein Gerät innerhalb einer bestimmten Entfernung (Radius) von einem Punkt befindet.

3. Sie können damit beginnen, Standorte anhand all Ihrer Geofences zu bewerten. Wenn eine Standortaktualisierung die Grenzen eines oder mehrerer Geofences überschreitet, gibt Ihre Geofence-Erfassungsressource bei Amazon einen der folgenden Geofence-Ereignistypen aus: EventBridge
 - ENTER — Für jeden Geofence, bei dem die Standortaktualisierung durch Eingabe die Grenze überschreitet, wird ein Ereignis generiert.
 - EXIT — Ein Ereignis wird für jeden Geofence generiert, bei dem das Standort-Update seine Grenze überschreitet, indem es ihn verlässt.

Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#). Sie können die Überwachung auch mithilfe von Diensten wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Wenn Sie beispielsweise eine LKW-Flotte verfolgen und benachrichtigt werden möchten, wenn sich ein Lkw in einem bestimmten Bereich eines Ihrer Lagerhäuser befindet. Sie können einen Geofence für den Bereich rund um jedes Lager erstellen. Wenn die Lkw Ihnen dann aktualisierte Standorte senden, können Sie Amazon Location Service verwenden, um diese Positionen auszuwerten und festzustellen, ob ein Lkw eines der Geofence-Gebiete erreicht (oder verlassen) hat.

Note

Ihre Abrechnung erfolgt nach der Anzahl der Geofence-Sammlungen, anhand derer Sie die Auswertung durchführen. Ihre Rechnung wird nicht von der Anzahl der Geofences in jeder Sammlung beeinflusst. Da jede Geofence-Sammlung bis zu 50.000 Geofences enthalten kann, sollten Sie Ihre Geofences nach Möglichkeit in weniger Sammlungen zusammenfassen, um die Kosten für Geofence-Bewertungen zu senken. Die generierten

Ereignisse enthalten die ID der einzelnen Geofences in der Sammlung sowie die ID der Sammlung.

Geofence-Ereignisse

Standorte von Positionen, die Sie überwachen, werden durch eine ID namens `a` referenziert `DeviceId` (und die Positionen werden als Gerätepositionen bezeichnet). Sie können eine Liste mit Gerätepositionen zur Auswertung direkt an die Ressource zur Erfassung von Geofences senden, oder Sie können einen Tracker verwenden. Weitere Informationen zu Trackern finden Sie im nächsten Abschnitt.

Sie erhalten Ereignisse (über Amazon EventBridge) nur, wenn ein Gerät einen Geofence betritt oder verlässt, nicht bei jeder Positionsänderung. Das bedeutet, dass Sie in der Regel Ereignisse erhalten und viel seltener darauf reagieren müssen als bei jeder Aktualisierung der Geräteposition.

Note

Bei der ersten Standortauswertung für ein bestimmtes Objekt wird davon ausgegangen `DeviceID`, dass sich das Gerät zuvor nicht in Geofences befand. Das erste Update generiert also ein ENTER Ereignis, wenn es sich innerhalb eines Geofences in der Sammlung befindet, und kein Ereignis, wenn nicht.

Um zu berechnen, ob ein Gerät einen Geofence betreten oder verlassen hat, muss Amazon Location Service den vorherigen Positionsstatus für das Gerät beibehalten. Dieser Positionsstatus wird für 30 Tage gespeichert. Nach 30 Tagen ohne Aktualisierung für ein Gerät wird eine neue Standortaktualisierung als erste Positionsaktualisierung behandelt.

Tracker

Ein Tracker speichert Positionsaktualisierungen für eine Sammlung von Geräten. Der Tracker kann verwendet werden, um den aktuellen Standort oder den Standortverlauf der Geräte abzufragen. Er speichert die Updates, reduziert aber den Speicherplatz und das visuelle Rauschen, indem die Standorte vor dem Speichern gefiltert werden.

Jedes in Ihren Tracker-Ressourcen gespeicherte Positionsupdate kann ein Maß für die Positionsgenauigkeit und bis zu 3 Felder mit Metadaten über die Position oder das Gerät enthalten,

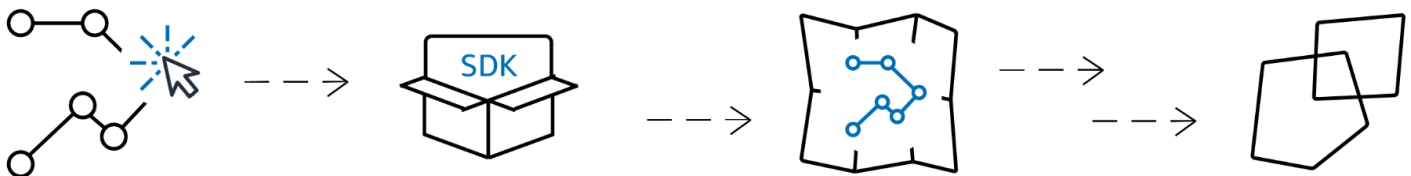
das Sie speichern möchten. Die Metadaten werden als Schlüssel-Wert-Paare gespeichert und können Informationen wie Geschwindigkeit, Richtung, Reifendruck oder Motortemperatur speichern.

Note

Der Tracker-Speicher wird automatisch mit AWS eigenen Schlüsseln verschlüsselt. Sie können mithilfe von KMS-Schlüsseln, die Sie verwalten, eine weitere Verschlüsselungsebene hinzufügen, um sicherzustellen, dass nur Sie auf Ihre Daten zugreifen können. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für Amazon Location Service](#).

Das Filtern und Speichern von Tracker-Positionen ist für sich genommen nützlich, aber Tracker sind besonders nützlich, wenn sie mit Geofences kombiniert werden. Sie können Tracker mit einer oder mehreren Ressourcen Ihrer Geofence-Erfassung verknüpfen. Positionsaktualisierungen werden dann automatisch anhand der Geofences in diesen Sammlungen ausgewertet. Durch den richtigen Einsatz von Filtern können Sie auch die Kosten für Ihre Geofence-Bewertungen erheblich senken.

Das folgende Diagramm zeigt Ihnen, wie Sie Tracker-Ressourcen erstellen und verwenden:



1. Zunächst erstellen Sie eine Tracker-Ressource in Ihrem AWS Konto.
2. Als Nächstes entscheiden Sie, wie Sie Standortaktualisierungen an Ihre Tracker-Ressourcen senden. Verwenden Sie [AWS SDKs](#), um Tracking-Funktionen in Ihre mobilen Anwendungen zu integrieren. Alternativ können Sie MQTT verwenden, indem Sie den step-by-step Anweisungen zur [Nachverfolgung](#) mit MQTT folgen.
3. Sie können jetzt Ihre Tracker-Ressource verwenden, um den Standortverlauf aufzuzeichnen und ihn auf einer Karte zu visualisieren.
4. Sie können Ihre Tracker-Ressource auch mit einer oder mehreren Geofence-Sammlungen verknüpfen, sodass jede an Ihre Tracker-Ressource gesendete Positionsaktualisierung automatisch mit allen Geofence-Daten in allen verknüpften Geofence-Sammlungen verglichen wird. Sie können Ressourcen auf der Seite mit den Tracker-Ressourcendetails der Amazon Location-Konsole oder mithilfe der Amazon Location Trackers API verknüpfen.

5. Anschließend können Sie die Überwachung mithilfe von Diensten wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Verwendung von Trackern mit Geofences

Tracker bieten zusätzliche Funktionen, wenn sie mit Geofences kombiniert werden. Sie verknüpfen einen Tracker mit einer Geofence-Erfassung, entweder über die Amazon Location Console oder die API, um Tracker-Standorte automatisch auszuwerten. Jedes Mal, wenn der Tracker einen aktualisierten Standort erhält, wird dieser Standort mit jedem Geofence in der Sammlung verglichen, und die entsprechenden ENTER EXIT Ereignisse werden in Amazon generiert. EventBridge Sie können auch Filter auf den Tracker anwenden und je nach Filterung die Kosten für Geofence-Bewertungen senken, indem Sie nur aussagekräftige Standortaktualisierungen auswerten.

Wenn Sie den Tracker einer Geofence-Sammlung zuordnen, nachdem er bereits einige Positionsaktualisierungen erhalten hat, wird das erste Positionsupdate nach der Zuordnung als erstes Update für die Geofence-Bewertungen behandelt. Wenn es sich innerhalb eines Geofence-Bereichs befindet, erhältst du ein Ereignis. ENTER Wenn es sich nicht innerhalb eines Geofences befindet, erhältst du kein EXIT Ereignis, unabhängig vom vorherigen Status.

Filterung nach Positionen

Tracker können die Positionen, die an sie gesendet werden, automatisch filtern. Es gibt mehrere Gründe, warum Sie möglicherweise einige Aktualisierungen Ihres Gerätestandorts herausfiltern möchten. Wenn Sie über ein System verfügen, das nur etwa jede Minute Berichte sendet, sollten Sie Geräte nach Zeit filtern und Positionen nur alle 30 Sekunden speichern und auswerten. Selbst wenn Sie häufiger überwachen, möchten Sie möglicherweise Positionsaktualisierungen filtern, um das Rauschen der GPS-Hardware zu beseitigen. GPS-Positionen sind von Natur aus laut. Ihre Genauigkeit ist nicht zu 100% perfekt, sodass sich selbst ein stationäres Gerät leicht zu bewegen scheint. Bei niedrigen Geschwindigkeiten verursacht dieses Zittern ein unübersichtliches Bild und kann zu falschen Ein- und Austrittsereignissen führen, wenn sich das Gerät in der Nähe des Randes eines Geofences befindet.

Die Positionsfilterung funktioniert, wenn Positionsaktualisierungen von einem Tracker empfangen werden. Dadurch wird das visuelle Rauschen auf Ihren Gerätepfaden (Jitter) reduziert, die Anzahl falscher Geofence-Ein- und Austrittsereignisse reduziert und die Kosten gesenkt, indem die Anzahl

der gespeicherten Positionsaktualisierungen und der ausgelösten Geofence-Bewertungen reduziert wird.

Tracker bieten drei Optionen zur Positionsfilterung, um die Kosten im Griff zu behalten und das Jitter bei Ihren Standortaktualisierungen zu reduzieren.

- Genauigkeitsbasiert — Kann mit jedem Gerät verwendet werden, das Genauigkeitsmessungen ermöglicht. Die meisten GPS- und Mobilgeräte liefern diese Informationen. Die Genauigkeit jeder Positionsmessung wird durch viele Umweltfaktoren beeinflusst, darunter den GPS-Satellitenempfang, die Landschaft und die Nähe von WLAN- und Bluetooth-Geräten. Die meisten Geräte, einschließlich der meisten Mobilgeräte, können zusammen mit der Messung eine Schätzung der Genauigkeit der Messung liefern. Bei der AccuracyBased Filterung ignoriert Amazon Location Standortaktualisierungen, wenn sich das Gerät weniger als die gemessene Genauigkeit bewegt hat. Wenn beispielsweise zwei aufeinanderfolgende Updates von einem Gerät einen Genauigkeitsbereich von 5 m und 10 m haben, ignoriert Amazon Location das zweite Update, wenn sich das Gerät weniger als 15 m bewegt hat. Amazon Location wertet ignorierte Updates weder anhand von Geofences aus, noch speichert sie.

Wenn keine Genauigkeit angegeben wird, wird sie als Null behandelt, und die Messung wird als absolut genau angesehen, und es wird keine Filterung auf die Aktualisierungen angewendet.

Note

Sie können die genauigkeitsbasierte Filterung verwenden, um alle Filter zu entfernen. Wenn Sie die genauigkeitsbasierte Filterung auswählen, aber alle Genauigkeitsdaten auf Null überschreiben oder die Genauigkeit ganz weglassen, filtert Amazon Location keine Aktualisierungen heraus.

In den meisten Szenarien ist die genauigkeitsbasierte Filterung eine gute Wahl für das Filtern von Positionsaktualisierungen. Sie bietet ein ausgewogenes Verhältnis zwischen der Standortverfolgung und dem Herausfiltern nicht benötigter Aktualisierungen, wodurch die Kosten gesenkt werden.

- Entfernungsbasiert — Verwenden Sie diese Option, wenn Ihre Geräte keine Genauigkeitsmessung bieten, Sie aber dennoch die Vorteile der Filterung nutzen möchten, um Jitter zu reduzieren und die Kosten zu senken. DistanceBased Beim Filtern werden Standortaktualisierungen ignoriert, bei denen sich Geräte weniger als 30 m (98,4 ft) bewegt haben. Wenn Sie die DistanceBased

Positionsfiltrierung verwenden, wertet Amazon Location diese ignorierten Aktualisierungen weder anhand von Geofences aus, noch speichert Amazon Location die Aktualisierungen.

Die Genauigkeit der meisten Mobilgeräte, einschließlich der durchschnittlichen Genauigkeit von iOS- und Android-Geräten, liegt innerhalb von 15 m. In den meisten Anwendungen können durch DistanceBased Filterung die Auswirkungen von Standortungenauigkeiten bei der Anzeige der Geräteflugbahn auf einer Karte sowie die Abprallwirkung mehrerer aufeinanderfolgender Ein- und Austrittsereignisse verringert werden, wenn sich Geräte in der Nähe der Grenze eines Geofences befinden. Es kann auch dazu beitragen, die Kosten Ihrer Anwendung zu senken, da weniger Aufrufe zur Auswertung verknüpfter Geofences oder zum Abrufen von Gerätepositionen getätigt werden.

Entfernungs-basierte Filterung ist nützlich, wenn Sie zwar filtern möchten, Ihr Gerät aber keine Genauigkeitsmessungen liefert, oder wenn Sie eine größere Anzahl von Aktualisierungen herausfiltern möchten als bei genauigkeitsbasierter Filterung.

- **Zeitbasiert** — (Standard) Verwenden Sie diese Option, wenn Ihre Geräte sehr häufig Positionsaktualisierungen senden (mehr als einmal alle 30 Sekunden) und Sie Geofence-Auswertungen nahezu in Echtzeit durchführen möchten, ohne jedes Update zu speichern. Bei der TimeBased Filterung wird jede Standortaktualisierung anhand verknüpfter Geofence-Sammlungen bewertet, aber nicht jede Standortaktualisierung wird gespeichert. Wenn Ihre Aktualisierungshäufigkeit häufiger als 30 Sekunden beträgt, wird nur ein Update pro 30 Sekunden für jede eindeutige Geräte-ID gespeichert.

Zeitbasierte Filterung ist besonders nützlich, wenn Sie weniger Positionen speichern möchten, aber möchten, dass jede Positionsaktualisierung anhand der zugehörigen Geofence-Sammlungen ausgewertet wird.

Note

Denken Sie bei der Entscheidung über Ihre Filtermethode und die Häufigkeit der Positionsaktualisierungen an die Kosten Ihrer Tracking-Anwendung. Ihnen wird jede Standortaktualisierung und einmal die Auswertung der Positionsaktualisierung anhand jeder verknüpften Geofence-Erfassung in Rechnung gestellt. Wenn Sie beispielsweise zeitbasierte Filterung verwenden und Ihr Tracker mit zwei Geofence-Sammlungen verknüpft ist, zählt jede Positionsaktualisierung als eine Anfrage zur Standortaktualisierung und als zwei Auswertungen der Geofence-Erfassung. Wenn du Positionsaktualisierungen für deine Geräte alle 5 Sekunden meldest und zeitbasierte Filterung verwendest, werden dir 720

Standortaktualisierungen und 1.440 Geofence-Bewertungen pro Stunde für jedes Gerät in Rechnung gestellt.

Geofence-Terminologie

Geofence-Sammlung

Enthält null oder mehr Geofences. Es ist in der Lage, Geofence zu überwachen, indem es auf Anfrage Eingangs- und Austrittsereignisse ausgibt, um die Position eines Geräts anhand seiner Geofences zu bewerten.

Geofence

Eine Polygon- oder Kreisgeometrie, die eine virtuelle Grenze auf einer Karte definiert.

Polygongeometrie

Ein Amazon Location Geofence ist eine virtuelle Grenze für ein geografisches Gebiet und wird als Polygongeometrie oder als Kreis dargestellt.

Ein Kreis ist ein Punkt, der von einer Entfernung umgeben ist. Verwenden Sie einen Kreis, wenn Sie benachrichtigt werden möchten, wenn sich ein Gerät in einer bestimmten Entfernung von einem Standort befindet.

Ein Polygon ist eine Anordnung, die aus einem oder mehreren linearen Ringen besteht. Verwenden Sie ein Polygon, wenn Sie eine bestimmte Grenze für Gerätebenachrichtigungen definieren möchten. Ein linearer Ring ist eine Anordnung von vier oder mehr Scheitelpunkten, wobei der erste und der letzte Scheitelpunkt identisch sind und somit eine geschlossene Grenze bilden. Jeder Eckpunkt ist ein zweidimensionaler Punkt der Form *[Längengrad, Breitengrad]*, wobei die Längen- und Breitengradeinheiten Grad sind. Die Scheitelpunkte müssen gegen den Uhrzeigersinn um das Polygon herum angeordnet sein.

Note

Amazon Location Service unterstützt keine Polygone mit mehr als einem Ring. Dazu gehören Löcher, Inseln oder Multipolygone. Amazon Location unterstützt auch keine Polygone, die im Uhrzeigersinn gewickelt sind oder den Antimeridian kreuzen.

Das Folgende ist ein Beispiel für einen einzelnen linearen externen Ring:

```
[
  [
    [-5.716667, -15.933333],
    [-14.416667, -7.933333],
    [-12.316667, -37.066667],
    [-5.716667, -15.933333]
  ]
]
```

Tracker-Terminologie

Tracker-Ressource

Eine AWS Ressource, die Standortaktualisierungen von Geräten empfängt. Die Tracker-Ressource bietet Unterstützung für Standortabfragen, z. B. für den aktuellen und historischen Gerätestandort. Beim Verknüpfen einer Tracker-Ressource mit einer Geofence-Sammlung werden Standortaktualisierungen automatisch mit allen Geofences in der verknüpften Geofence-Sammlung verglichen.

Nachverfolgte Positionsdaten

Eine Tracker-Ressource speichert Informationen über Ihre Geräte im Laufe der Zeit. Die Informationen umfassen eine Reihe von Positionsaktualisierungen, wobei jedes Update Standort, Uhrzeit und optionale Metadaten enthält. Die Metadaten können die Genauigkeit einer Position und bis zu drei Schlüsselwertpaare enthalten, mit deren Hilfe Sie wichtige Informationen zu jeder Position verfolgen können, z. B. Geschwindigkeit, Richtung, Reifendruck, verbleibenden Kraftstoff oder Motortemperatur des Fahrzeugs, das Sie verfolgen. Tracker speichern den Gerätestandortverlauf für 30 Tage.

Filterung von Positionen

Mithilfe von Positionsfiltern können Sie die Kosten kontrollieren und die Qualität Ihrer Tracking-Anwendung verbessern, indem Positionsaktualisierungen, die keine wertvollen Informationen liefern, herausgefiltert werden, bevor die Aktualisierungen gespeichert oder anhand von Geofences ausgewertet werden.

Sie können wählen `AccuracyBasedDistanceBased`, oder `TimeBased` filtern. Standardmäßig ist die Positionsfilterung auf `eingestelltTimeBased`.

Sie können die Positionsfilterung konfigurieren, wenn Sie Tracker-Ressourcen erstellen oder aktualisieren.

Zeitstempelformat nach RFC 3339

Amazon Location Service Trackers verwendet das Format [RFC 3339](#), das dem Format [8601 der International Organization for Standardization \(ISO\)](#) für Datum und Uhrzeit folgt.

Das Format lautet „yyyy-MM-DDThh:mm:ss.sssz+ 00:00“:

- YYYY-MM-DD— Stellt das Datumsformat dar.
- T— Zeigt an, dass die Zeitwerte folgen werden.
- hh:mm:ss.sss— Stellt die Uhrzeit im 24-Stunden-Format dar.
- Z— Gibt an, dass die verwendete Zeitzone UTC ist, worauf Abweichungen von der UTC-Zeitzone folgen können.
- +00:00— Geben Sie optional Abweichungen von der UTC-Zeitzone an. Zum Beispiel bedeutet +01:00 UTC + 1 Stunde.

Beispiel

Für den 2. Juli 2020 um 12:15:20 Uhr nachmittags, mit einer Anpassung um weitere 1 Stunde an die UTC-Zeitzone.

```
2020-07-02T12:15:20.000Z+01:00
```

Häufige Anwendungsfälle für die Nutzung von Amazon Location Service

Mit Amazon Location Service können Sie eine Reihe von Anwendungen erstellen, von der Inventarverfolgung bis hin zu standortbezogenem Marketing. Die folgenden Anwendungsfälle sind häufig:

Nutzerinteraktion und Geomarketing

Verwenden Sie Standortdaten, um Lösungen zu entwickeln, die die Nutzerinteraktion mit zielgerichtetem Marketing verbessern. Amazon Location kann beispielsweise ein Ereignis auslösen, das eine Benachrichtigung auslöst, wenn ein Kunde, der in seiner mobilen App einen Kaffee bestellt hat, in der Nähe ist. Darüber hinaus können Sie Geotargeting-Funktionen

einrichten, sodass Einzelhändler Rabattcodes oder digitale Flyer an Kunden senden können, die sich in der Nähe von Zielgeschäften befinden.

Nachverfolgung von Vermögenswerten

Entwickeln Sie Funktionen zur Inventarverfolgung, die Unternehmen dabei helfen, den aktuellen und historischen Standort ihrer Produkte, Mitarbeiter und Infrastruktur zu verstehen. Mit Funktionen zur Inventarverfolgung können Sie eine Reihe von Lösungen entwickeln, die die Personalausstattung an entfernten Standorten optimieren, den Versand unterwegs sichern und die Effizienz des Versands maximieren.

Lieferung

Integrieren Sie Standortfunktionen in Lieferanwendungen, um den Abflugort, die Lieferfahrzeuge und deren Zielort zu speichern, zu verfolgen und zu koordinieren. Beispielsweise verfügt eine Anwendung für die Lieferung von Lebensmitteln mit integrierten Funktionen für Amazon Location über Funktionen zur Standortverfolgung und Geofencing, mit denen ein Restaurant automatisch benachrichtigt werden kann, wenn ein Lieferfahrer in der Nähe ist. Dies reduziert die Wartezeit und trägt dazu bei, die Qualität der gelieferten Lebensmittel aufrechtzuerhalten.

Dieses Thema bietet Ihnen einen Überblick über die Architektur und die Schritte für Anwendungen, die Sie mit Amazon Location erstellen können.

Themen

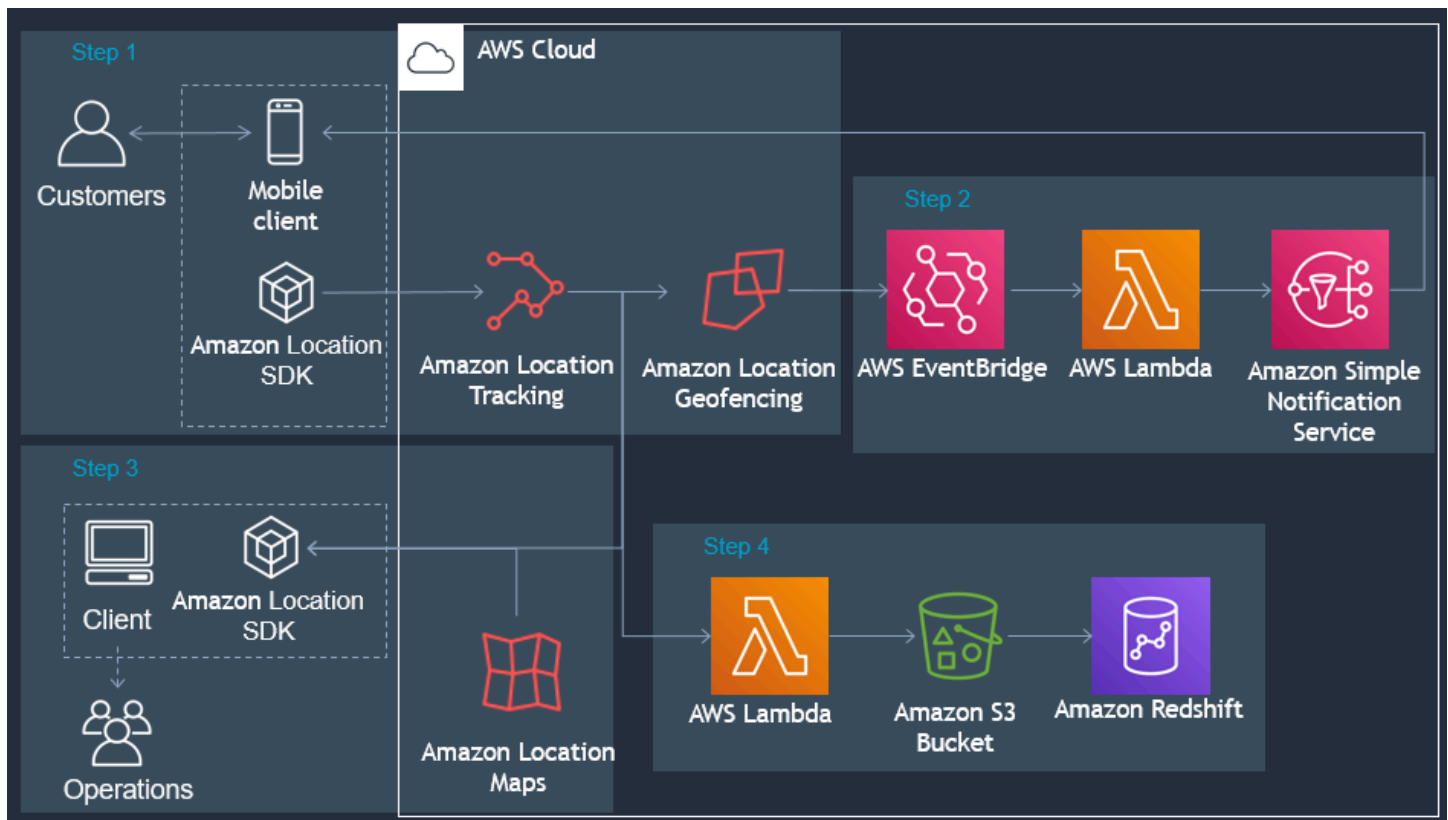
- [Anwendungen für Benutzerinteraktion und Geomarketing](#)
- [Anwendungen zur Nachverfolgung von Vermögenswerten](#)
- [Bereitstellungsanwendungen](#)

Anwendungen für Benutzerinteraktion und Geomarketing

Im Folgenden wird eine Anwendungsarchitektur für Benutzerinteraktion und Geomarketing mithilfe von Amazon Location veranschaulicht:

Mit dieser Architektur können Sie:

- Initiieren Sie Ereignisse auf der Grundlage der Nähe eines Ziels, sodass Sie Kunden an Kunden in der Nähe weiterleiten oder Personen ansprechen können, die Ihr Unternehmen kürzlich verlassen haben (sogenanntes Geotargeting).
- Visualisieren Sie die Standorte der Kundengeräte auf einer Karte, um Trends im Zeitverlauf zu verfolgen.
- Speichern Sie die Standorte der Kundengeräte, die Sie im Laufe der Zeit analysieren können.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie einen Überblick über die Schritte, die zum Erstellen einer Anwendung für Benutzerinteraktion und Geomarketing erforderlich sind:

1. Erstellen Sie Ihre Geofences in Geofence-Sammlungen und verknüpfen Sie Tracker mit ihnen. Weitere Informationen finden Sie unter [the section called “Geofencing und Tracking”](#).
2. Konfigurieren Sie Amazon so EventBridge, dass eine Benachrichtigung an Kunden gesendet wird, die ein geografisch abgegrenztes Interessengebiet betreten oder verlassen. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).

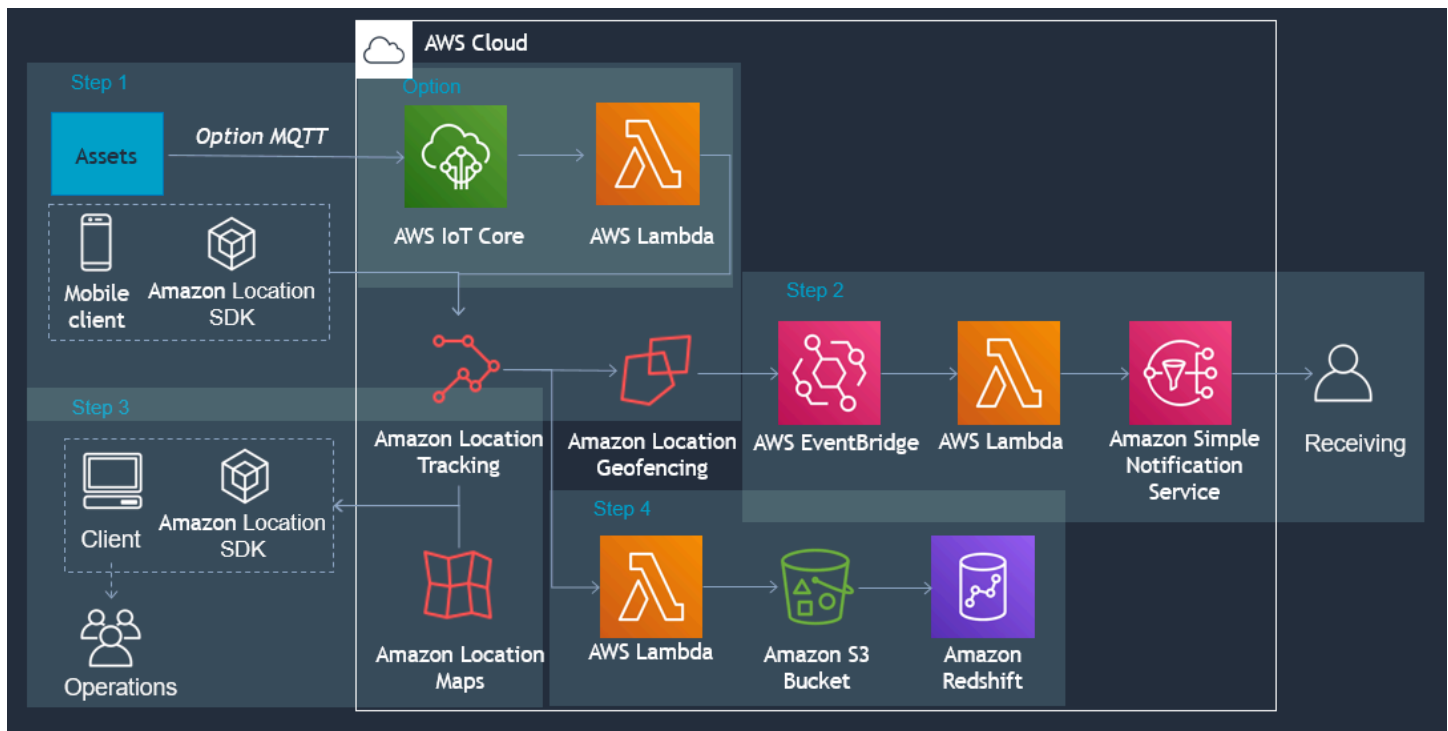
3. Zeigen Sie Kundenstandorte und Geofences auf einer Karte an. Weitere Informationen finden Sie unter Karten [verwenden](#).
4. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
5. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon verwenden CloudWatch und AWS CloudTrail Ihre Anwendung verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Anwendungen zur Nachverfolgung von Vermögenswerten

Die folgende Abbildung zeigt die Architektur einer Asset-Tracking-Anwendung, die Amazon Location verwendet:

Mit dieser Architektur können Sie:

- Zeigen Sie die Standorte von Anlagen auf einer Karte an, um das Gesamtbild zu veranschaulichen. Zum Beispiel die Anzeige einer Heatmap mit historischen Orten oder Ereignissen, um einem Betriebs- oder Planungsteam zu helfen.
- Initiieren Sie Ereignisse auf der Grundlage der Nähe von Anlagen, um eine Empfangsabteilung zu benachrichtigen, damit sie sich auf den Wareneingang vorbereiten und die Bearbeitungszeit reduzieren kann.
- Speichern Sie die Standorte von Ressourcen, um Aktionen in Ihren Backend-Anwendungen einzuleiten oder Daten im Laufe der Zeit zu analysieren.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie einen Überblick über die Schritte, die zum Erstellen einer Asset-Tracking-Anwendung erforderlich sind:

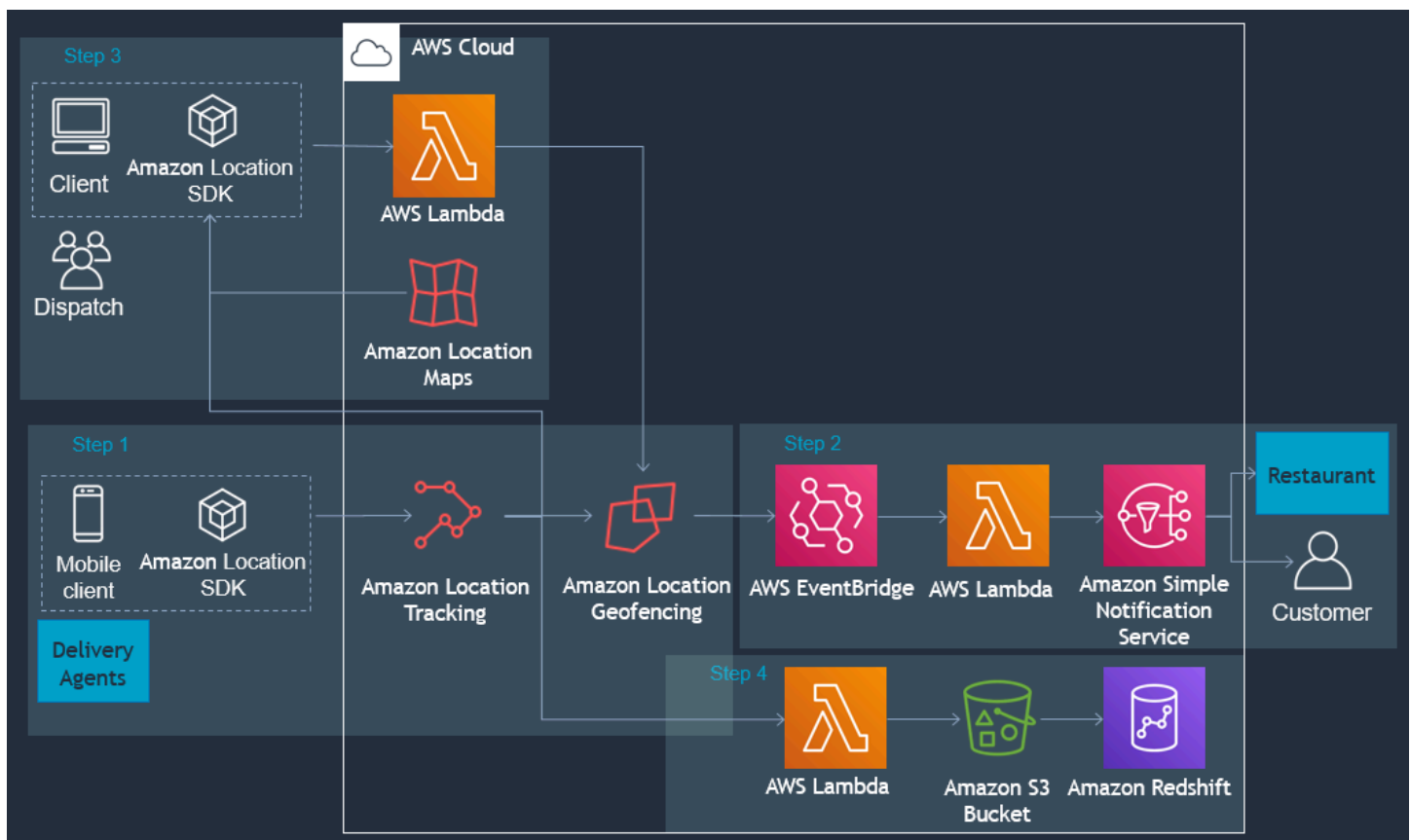
1. Erstellen Sie Ihre Geofences in Geofence-Sammlungen und verknüpfen Sie Tracker mit ihnen. Weitere Informationen finden Sie unter [the section called “Geofencing und Tracking”](#).
2. Konfigurieren Sie Amazon so EventBridge , dass es eine Benachrichtigung sendet oder einen Prozess einleitet. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).
3. Zeigen Sie Ihre verfolgten Ressourcen und Ihre aktiven Geofences auf einer Karte an. Weitere Informationen finden Sie unter [Karten verwenden](#).
4. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
5. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon verwenden CloudWatch und AWS CloudTrail Ihre Anwendung verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Bereitstellungsanwendungen

Im Folgenden finden Sie eine Abbildung einer Bereitstellungsanwendungsarchitektur, die Amazon Location verwendet.

Mit dieser Architektur können Sie:

- Initiieren Sie Ereignisse auf der Grundlage der Nähe der Zusteller, sodass die Abholung rechtzeitig erfolgt und die Kunden benachrichtigt werden können, wenn ihre Lieferung eintrifft.
- Zeigen Sie die Standorte der Fahrer sowie die Abhol- und Abgabestellen nahezu in Echtzeit auf einer Karte an, um den Versandteams das Gesamtbild zu zeigen.
- Speichern Sie die Standorte der Zusteller, sodass Sie in Ihrer Backend-Anwendung darauf reagieren oder sie im Laufe der Zeit analysieren können.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie eine Übersicht über die Schritte, die zum Erstellen einer Bereitstellungsanwendung erforderlich sind:

1. Erstellen Sie Ihre Geofence-Sammlungen und verknüpfen Sie verfolgte Geräte mit der Sammlung. Weitere Informationen finden Sie unter [the section called "Geofencing und Tracking"](#).
2. Erstellen Sie eine AWS Lambda Funktion zum automatischen Hinzufügen und Entfernen von Geofences, wenn Ihre Bestellungen gebucht werden.

3. Konfigurieren Sie Amazon so EventBridge , dass Benachrichtigungen gesendet oder Prozesse initiiert werden. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).
4. Zeigen Sie verfolgte Ressourcen und aktive Geofences auf einer Karte an. Weitere Informationen finden Sie unter Karten [verwenden](#).
5. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
6. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon verwenden CloudWatch und AWS CloudTrail Ihre Anwendung verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Was ist ein Datenanbieter?

Verwenden Sie Amazon Location Service, um über Ihr AWS Konto auf Geolocation-Ressourcen von mehreren Datenanbietern zuzugreifen, ohne dass Verträge oder Integrationen mit Drittanbietern erforderlich sind. Auf diese Weise können Sie sich auf die Erstellung Ihrer Anwendung konzentrieren, ohne Konten, Anmeldeinformationen, Lizenzen und Rechnungen von Drittanbietern verwalten zu müssen.

Die folgenden Amazon-Ortungsdienste verwenden Datenanbieter.

- Karten — Wählen Sie beim [Erstellen einer Kartenressource Stile von verschiedenen Kartenanbietern](#) aus. Sie können Kartenressourcen verwenden, um eine interaktive Karte zur Visualisierung von Daten zu erstellen.
- Orte — Wählen Sie einen Datenanbieter aus, wenn Sie [eine Ortsindexressource erstellen](#), um Abfragen für Geokodierung, umgekehrte Geokodierung und Suchen zu unterstützen.
- Routen — Wählen Sie einen Datenanbieter aus, der Abfragen für Routenberechnungen in verschiedenen Regionen und Anwendungen unterstützt, wenn Sie [eine Routenberechnungsressource erstellen](#). Mit dem von Ihnen ausgewählten Datenanbieter können Sie mit Amazon Location Service Routen auf der Grundlage von up-to-date Straßennetzdaten, Live-Verkehrsdaten, geplanten Sperrungen und historischen Verkehrsmustern berechnen.

Jeder Anbieter sammelt und kuratiert seine Daten mit unterschiedlichen Mitteln. Sie verfügen möglicherweise auch über unterschiedliche Fachkenntnisse in verschiedenen Regionen der Welt. Dieser Abschnitt enthält Einzelheiten zu unseren Datenanbietern. Sie können einen beliebigen Datenanbieter nach Ihren Wünschen auswählen.

Stellen Sie sicher, dass Sie die Allgemeinen Geschäftsbedingungen lesen, wenn Sie Datenanbieter von Amazon Location Service verwenden. Weitere Informationen finden Sie in den [AWS-Servicebedingungen](#). the section called "Datenschutz"In diesem Abschnitt finden Sie auch weitere Informationen darüber, wie Amazon Location Ihre Privatsphäre schützt.

Reichweite und Funktionen des Datenanbieters

Die folgende Tabelle zeigt die Abdeckung und die Funktionen für jeden Datenanbieter auf hohem Niveau.

Datenanbieter	Geografischer Geltungsbereich	Abdeckung der Funktionen	AWS-Region
Esri	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Schnapp dir	Südost-Asien	Karten, Orte, Routen	Nur Asien-Pazifik (Singapur). ap-southeast-1
HIER	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Daten öffnen	Global	Zuordnungen	Alle Regionen , in denen Amazon Location verfügbar ist.

Weitere Informationen zu den spezifischen Funktionen der einzelnen Datenanbieter finden Sie unter [Funktionen nach Datenanbieter](#).

Jeder Datenanbieter sammelt und produziert Daten auf unterschiedliche Weise. In den folgenden Themen erfahren Sie mehr über ihre Versorgungsgebiete:

- [Berichterstattung: Esri](#)
- [Berichterstattung: Schnappen](#)
- [Abdeckung: HIER](#)

- [Geltungsbereich: Offene Daten](#)

Wenn Sie auf ein Problem mit den Daten stoßen und dem Datenanbieter einen Fehler melden möchten, finden Sie weitere Informationen in den folgenden Themen:

- [Fehler beim Melden an Esri](#)
- [Fehlerberichterstattung für Daten GrabMaps](#)
- [Fehler bei der Meldung an HERE](#)
- [Fehler melden und zu Open Data beitragen](#)

Kartenstile

Jeder Datenanbieter bietet eine Reihe von Kartenstilen zum Rendern der von ihm bereitgestellten Kartendaten. Ein Style kann beispielsweise Satellitenbilder enthalten oder so optimiert sein, dass er die Straßen für die Navigation anzeigt. Die Liste und Beispiele der Stile für jeden Anbieter finden Sie in den folgenden Themen.

- [Kartenstile von Esri](#)
- [Schnappen Sie sich Kartenstile](#)
- [Kartenstile von HERE](#)
- [Map-Stile für Open Data](#)

Weitere Informationen zu den einzelnen Datenanbietern

Die folgenden Links bieten weitere Informationen zu den einzelnen Datenanbietern.

- [Esri](#)
- [GrabMaps](#)
- [HERE Technologies](#)
- [Daten öffnen](#)

Esri

Amazon Location Service nutzt die Ortungsdienste von Esri, um AWS Kunden dabei zu unterstützen, Karten effektiv zu nutzen, zu geokodieren und Routen zu berechnen. Die Ortungsdienste von Esri

basieren auf hochwertigen, verlässlichen ready-to-use Standortdaten, die von Expertenteams aus Kartografen, Geografen und Demografen kuratiert wurden.

Weitere Informationen zu den Funktionen finden Sie unter Datenanbietern von [Esri](#) on Amazon Location Service.

Themen

- [Kartenstile von Esri](#)
- [Berichterstattung: Esri](#)
- [Nutzungsbedingungen und Datenzuweisung: Esri](#)
- [Fehler beim Melden an Esri](#)

Kartenstile von Esri

Amazon Location Service unterstützt beim [Erstellen einer Kartenressource](#) die folgenden Kartenstile von Esri.

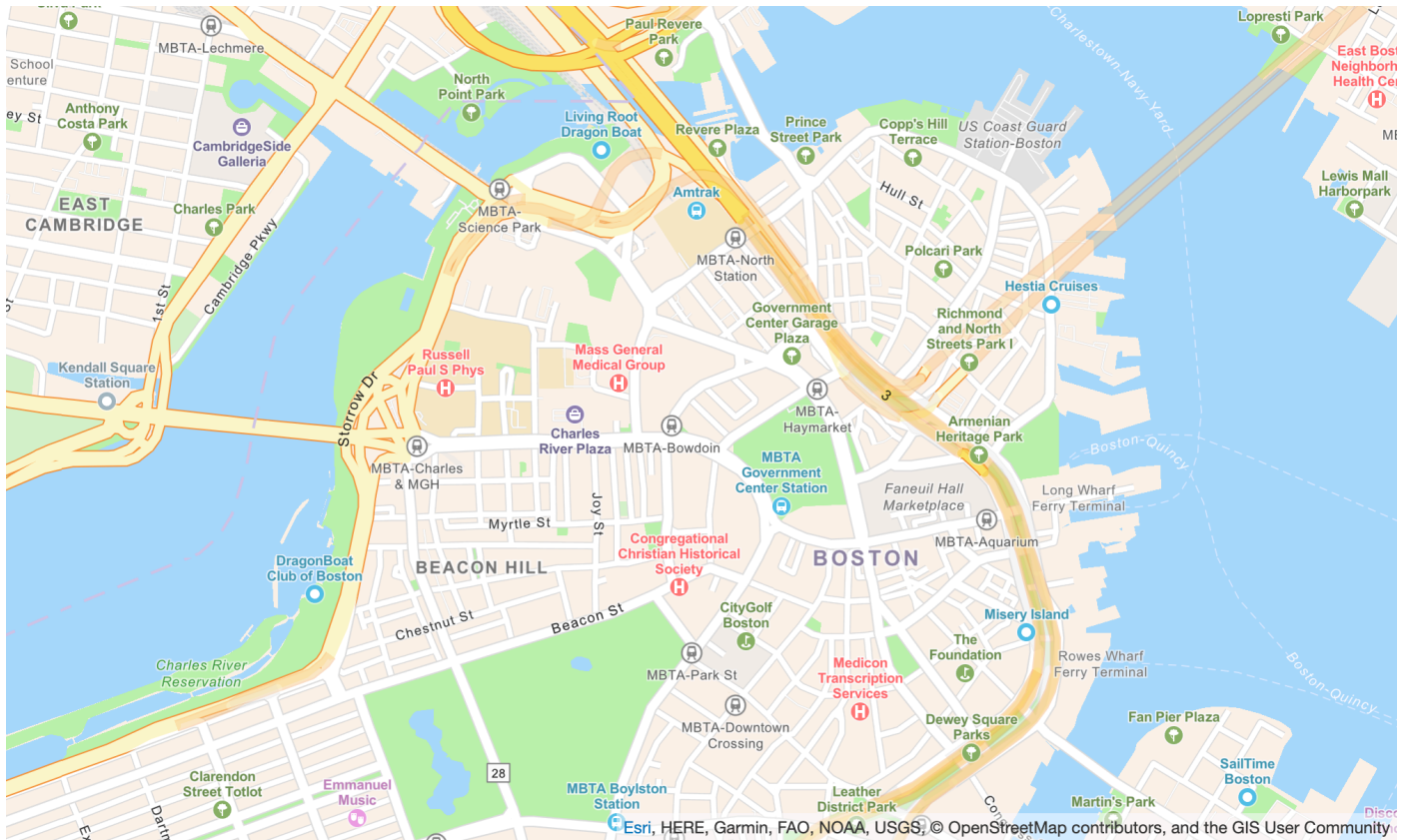
Note

Kartenstile von Esri, die in diesem Abschnitt nicht aufgeführt sind, werden nicht unterstützt.

Die Vektorstile von Esri unterstützen alternative Formate. [Politische Ansichten](#)

Esri Navigation

Esri Navigation



Name des Kartenstils: VectorEsriNavigation

Diese Karte bietet eine detaillierte Grundkarte für die Welt, symbolisiert durch einen benutzerdefinierten Navigationskartenstil, der für die Verwendung während des Tages auf Mobilgeräten konzipiert ist.

Diese umfassende Straßenkarte umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen. Der Vektorkachel-Layer in dieser Karte wurde mit denselben Datenquellen erstellt, die auch für die World Street Map und andere Esri Grundkarten verwendet wurden. Aktivieren Sie den POI Layer, indem Sie ihn so einrichten [CustomLayers](#), dass er die zusätzlichen Ortsdaten nutzt.

Weitere Informationen finden Sie unter [Esri World Navigation](#) auf der Esri Website.

Note

In der oben abgebildeten VectorEsriNavigation Karte ist der POI Layer aktiviert.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Arial Italic
- Arial Regulär
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regulär

Esri Imagery

Esri Bilddaten

Name des Kartenstils: `RasterEsriImagery`

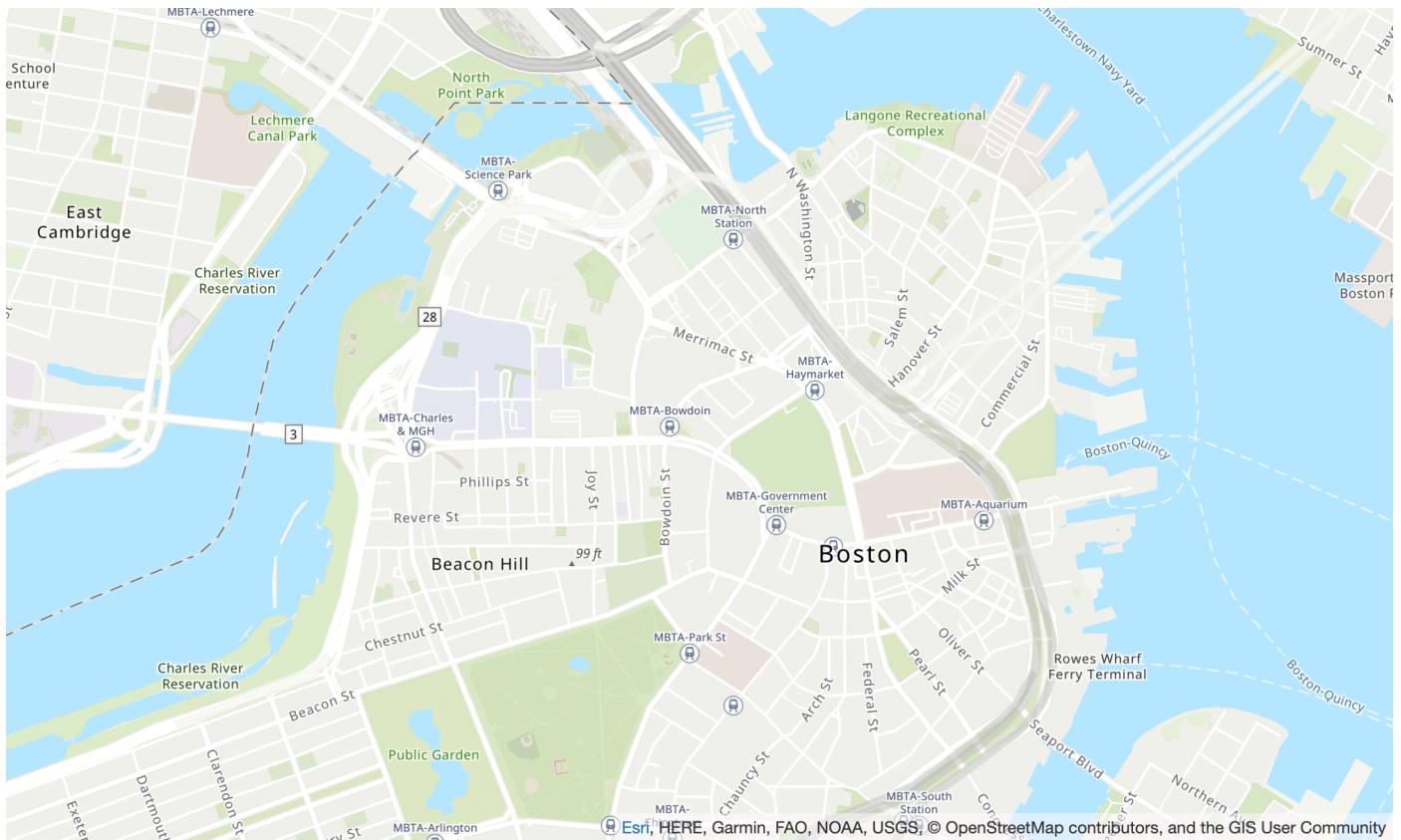
Diese Karte bietet Satelliten- und Luftbilder von einem Meter oder mehr in vielen Teilen der Welt und Satellitenbilder mit niedrigerer Auflösung weltweit.

Die Karte enthält 15 Millionen Bilder in kleinen und mittleren Maßstäben (~ 1:591 M bis ~ 1:72 k) und 2,5 m SPOT-Bilder (~ 1:288 k bis ~ 1:72 k) für die ganze Welt. Die Karte enthält Bilder von Maxar mit einer Auflösung von 0,5 m vom Festland der Vereinigten Staaten und Teilen Westeuropas. Diese Karte enthält zusätzliche Maxar-Submeter-Bilder in vielen Teilen der Welt. In anderen Teilen der Welt hat die GIS-Benutzergemeinschaft Bilddaten mit unterschiedlichen Auflösungen bereitgestellt. In ausgewählten Gemeinden sind Bilder mit sehr hoher Auflösung (bis zu 0,03 m) bis zu einem Maßstab von ~ 1:280 verfügbar.

Weitere Informationen finden Sie unter [Esri World Imagery auf der Esri Website](#).

Esri Light

Esri Light



Name des Kartenstils: VectorEsriTopographic

Dies bietet eine detaillierte Grundkarte für die Welt, symbolisiert durch einen klassischen Kartenstil von Esri. Dazu gehören Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen.

Diese Grundkarte wurde aus einer Vielzahl verlässlicher Quellen von verschiedenen Datenanbietern zusammengestellt, darunter dem US Geological Survey (USGS), der US Environmental Protection Agency (EPA), dem US National Park Service (NPS), der Ernährungs- und Landwirtschaftsorganisation der Vereinten Nationen (FAO), dem Department of Natural Resources Canada (NRCAN), HERE und Esri. Die Daten für ausgewählte Gebiete stammen von Mitwirkenden. OpenStreetMap Darüber hinaus werden Daten von der GIS-Community bereitgestellt.

Schriftarten

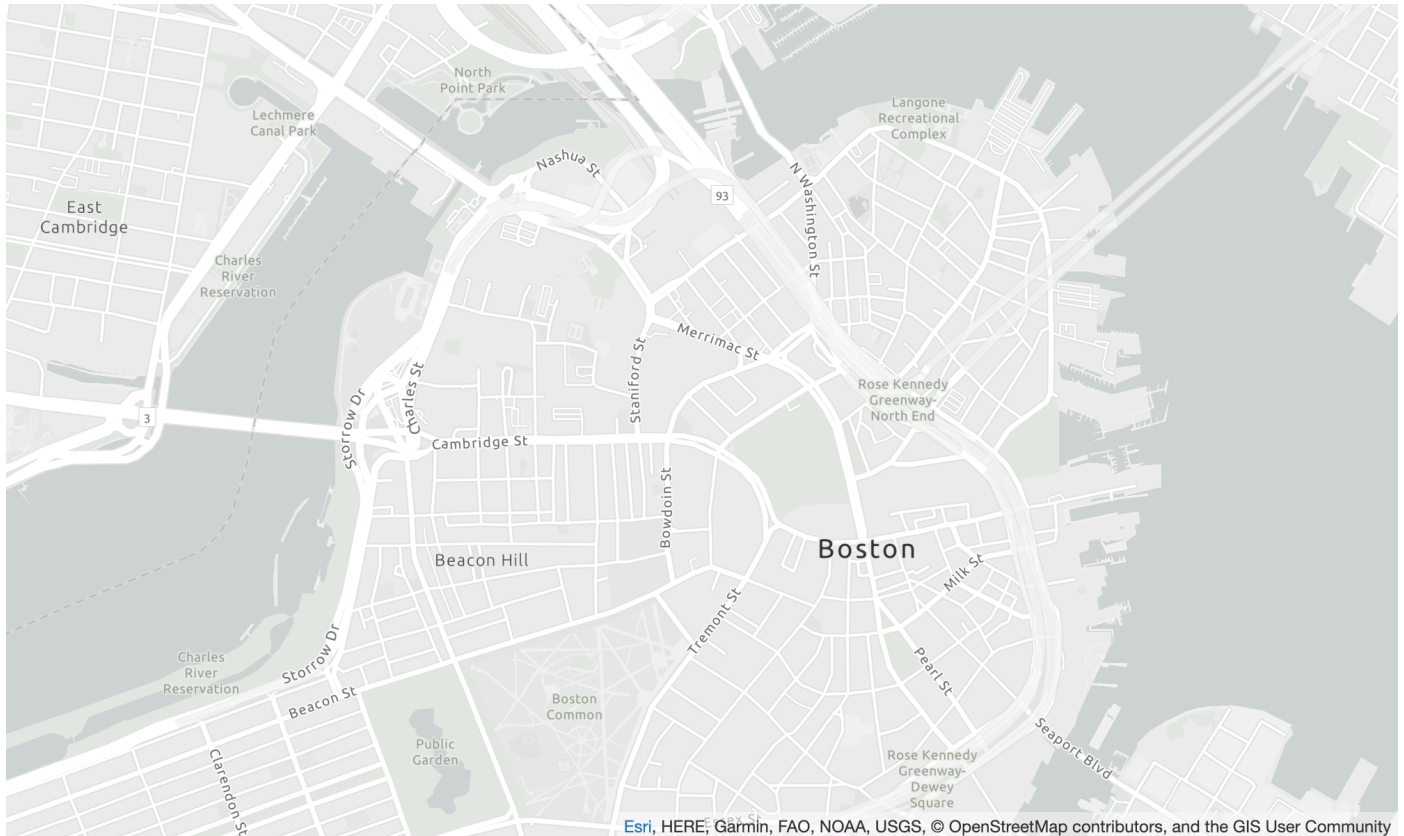
Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Noto Sans Italic

- Noto Sans Regulär
- Noto Sans Bold
- Noto Serif Regulär
- Roboto Condensed Light Kursiv

Esri Light Gray Canvas

Hellgraue Leinwand von Esri



Name des Kartenstils: `VectorEsriLightGrayCanvas`

Diese Karte bietet eine detaillierte Grundkarte für die Welt, die durch einen hellgrauen, neutralen Hintergrundstil mit minimalen Farben, Beschriftungen und Funktionen symbolisiert wird und die Aufmerksamkeit auf Ihren thematischen Inhalt lenken soll.

Dieser Vektorkachel-Layer wurde mit denselben Datenquellen erstellt, die auch für die hellgraue Leinwand und andere Esri Grundkarten verwendet wurden. Die Karte umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen.

Weitere Informationen finden Sie unter [Esri Light Gray Canvas](#) auf der Esri Website.

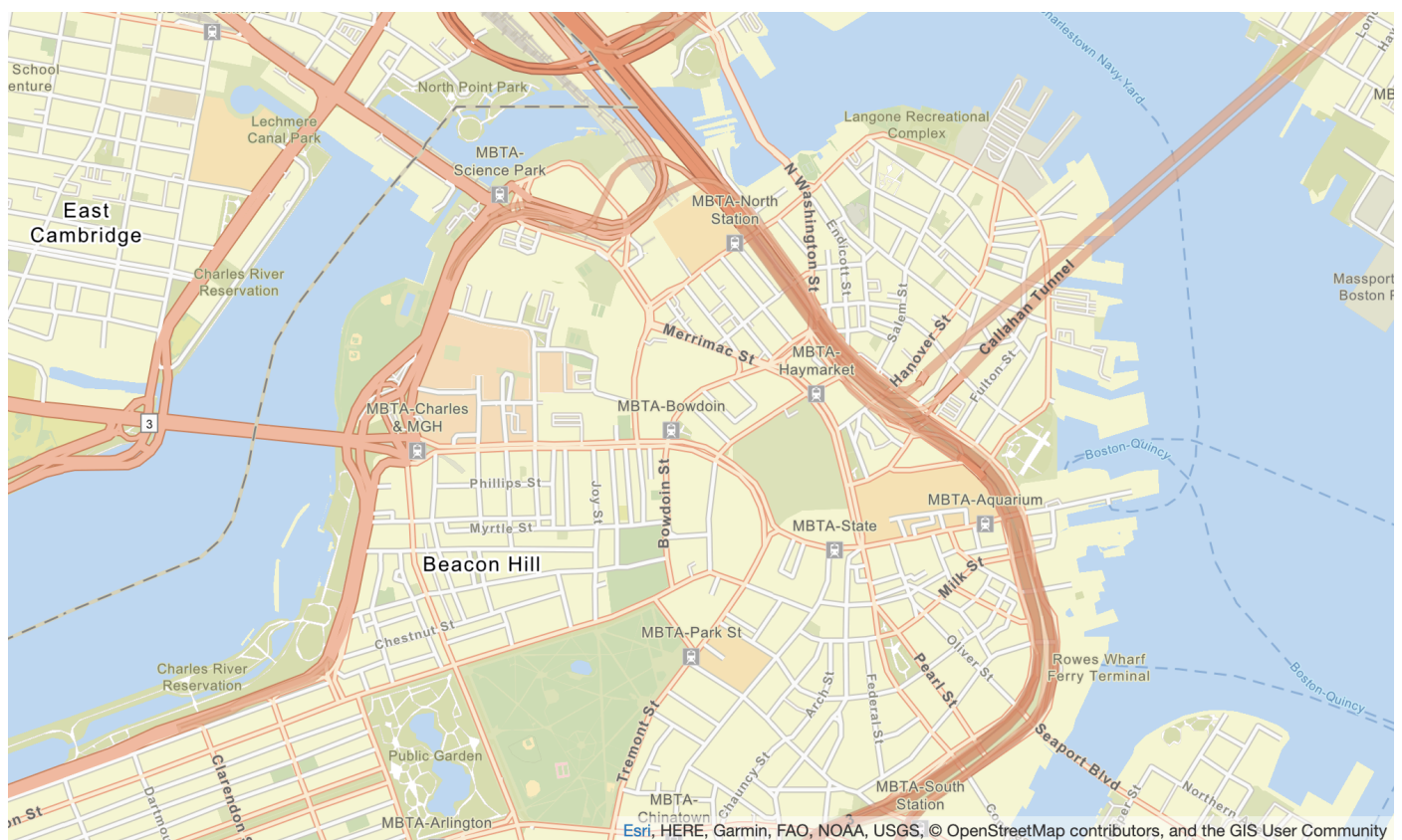
Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Ubuntu Kursiv
- Ubuntu Regulär
- Ubuntu Light
- Ubuntu fett

Esri Street Map

Esri Straßenkarte



Name des Kartenstils: `VectorEsriStreets`

Diese Karte bietet eine detaillierte Grundkarte für die Welt, symbolisiert durch einen benutzerdefinierten Navigationskartenstil, der für die Verwendung während des Tages auf Mobilgeräten konzipiert ist.

Diese umfassende Straßenkarte umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen. Sie umfasst auch eine größere Auswahl an Orten wie Geschäfte, Dienstleistungen, Restaurants, Attraktionen und andere Sehenswürdigkeiten. Der Vektorkachel-Layer in dieser Karte wurde mit denselben Datenquellen erstellt, die auch für die World Street Map und andere Esri Grundkarten verwendet wurden.

Weitere Informationen finden Sie auf der [Esri Website unter Esri World Street](#).

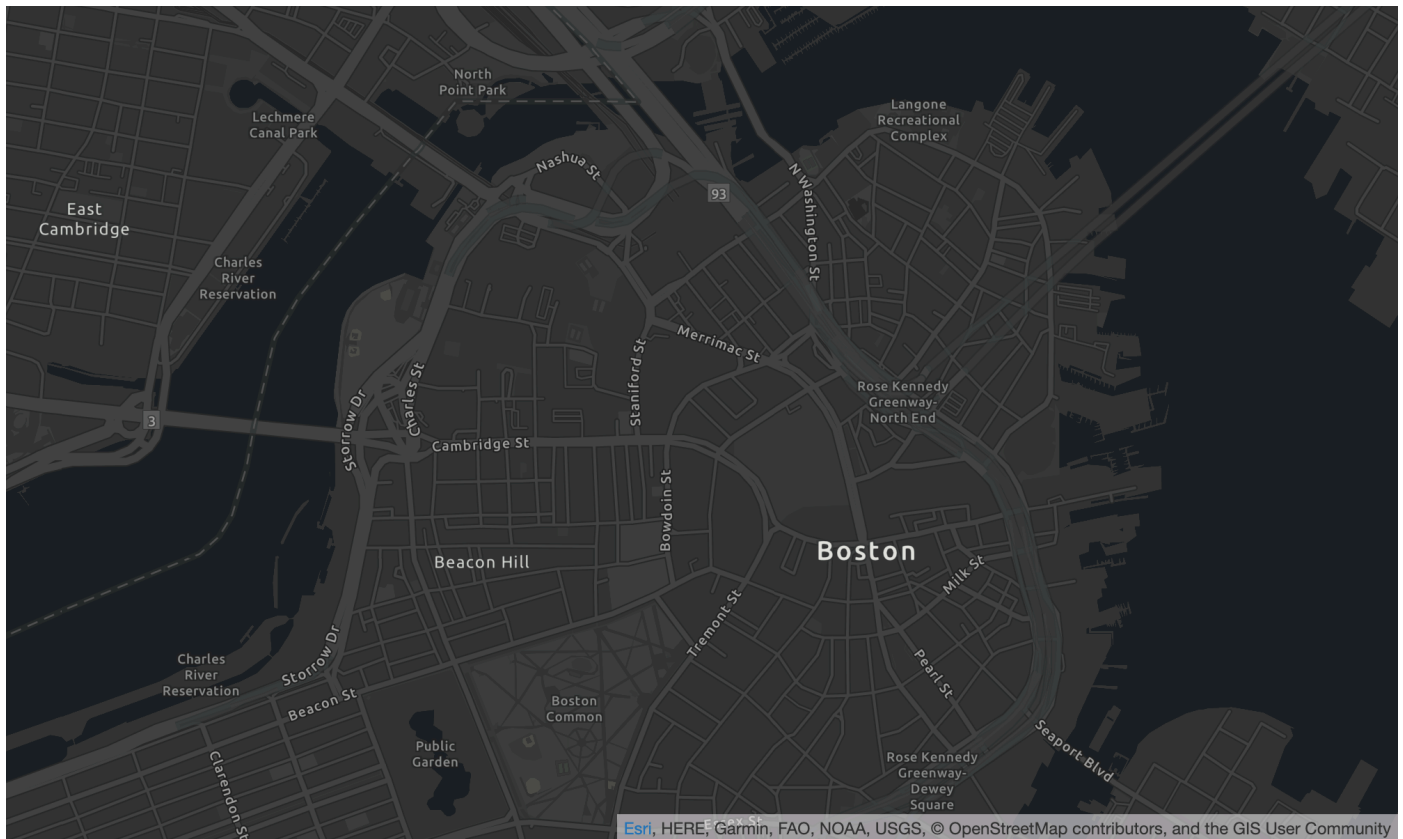
Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Arial Italic
- Arial Regulär
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regulär

Esri Dark Gray Canvas

Dunkelgraue Leinwand von Esri



Name des Kartenstils: `VectorEsriDarkGrayCanvas`

Diese Karte bietet eine detaillierte Vektor-Grundkarte für die Welt, die durch einen dunkelgrauen, neutralen Hintergrundstil mit minimalen Farben, Beschriftungen und Funktionen symbolisiert wird und die Aufmerksamkeit auf Ihren thematischen Inhalt lenken soll.

Diese Karte umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen. Die Vektorkachel-Layer in dieser Karte werden mit denselben Datenquellen erstellt, die auch für die Rasterkarte Dark Grey Canvas und andere Esri Grundkarten verwendet wurden.

Weitere Informationen finden Sie unter [Esri Dark Gray Canvas](#) auf der Esri Website.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Ubuntu Medium Italic
- Ubuntu Medium

- Ubuntu Kursiv
- Ubuntu Regulär
- Ubuntu Fett

Berichterstattung: Esri

Sie können Esri als Datenanbieter verwenden, um Abfragen für Geokodierung, umgekehrte Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindexressource erstellen](#), oder um Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie eine Routenberechnungsressource [erstellen](#).

Esri bietet unterschiedliche Datenqualitätsstufen in verschiedenen Regionen der Welt. Weitere Informationen zur Abdeckung in Ihrer Region, für die Sie sich interessieren, finden Sie unter:

- [Einzelheiten von Esri zur Geokodierungsabdeckung](#)
- [Einzelheiten zu Straßennetzen und Verkehrsabdeckung von Esri](#)

Nutzungsbedingungen und Datenzuweisung: Esri

Bevor Sie die Daten von Esri verwenden, stellen Sie sicher, dass Sie alle geltenden rechtlichen Anforderungen einhalten können, einschließlich der für Esri und geltenden Lizenzbedingungen. AWS

Weitere Informationen zu den AWS Anforderungen finden Sie in den [AWS-Servicebedingungen](#).

Informationen zu den Attributionsrichtlinien von Esri finden Sie in den [Datenattributionen und Nutzungsbedingungen](#) von Esri.

Fehler beim Melden an Esri

Wenn Sie auf ein Problem mit den Daten stoßen und Esri Fehler und Unstimmigkeiten melden möchten, folgen Sie dem technischen Support-Artikel von Esri [unter So geht's: Feedback zu Grundkarten und Geokodierung geben](#).

GrabMaps

Grab ist die größte Lieferorganisation in Südostasien mit Millionen von Fahrerpartnern und Kunden. Ihre Tochtergesellschaft [GrabMaps](#), erstellt up-to-date Kartendaten in diesen Ländern/Regionen für

den eigenen Gebrauch und für andere. Amazon Location Service nutzt GrabMaps Ortungsdienste, um AWS Kunden dabei zu unterstützen, Karten effektiv zu nutzen, zu geokodieren und Routen zu berechnen. GrabMaps' Die Ortungsdienste sind darauf ausgelegt, qualitativ hochwertige, verlässliche ready-to-use Standortdaten bereitzustellen, insbesondere für südostasiatische Länder.

Informationen zu zusätzlichen Funktionen finden Sie [GrabMaps](#) unter Datenanbieter von Amazon Location Service.

Important

Grab bietet Karten nur für Südostasien und ist nur in der Region Asien-Pazifik (Singapur) (ap-southeast-1) verfügbar. Weitere Informationen finden Sie unter [Länder/Regionen und abgedecktes Gebiet](#).

Themen

- [Schnappen Sie sich Kartenstile](#)
- [Berichterstattung: Schnappen](#)
- [Länder/Regionen und abgedecktes Gebiet](#)
- [Nutzungsbedingungen und Datenzuweisung: Grab](#)
- [Fehlerberichterstattung für Daten GrabMaps](#)

Schnappen Sie sich Kartenstile

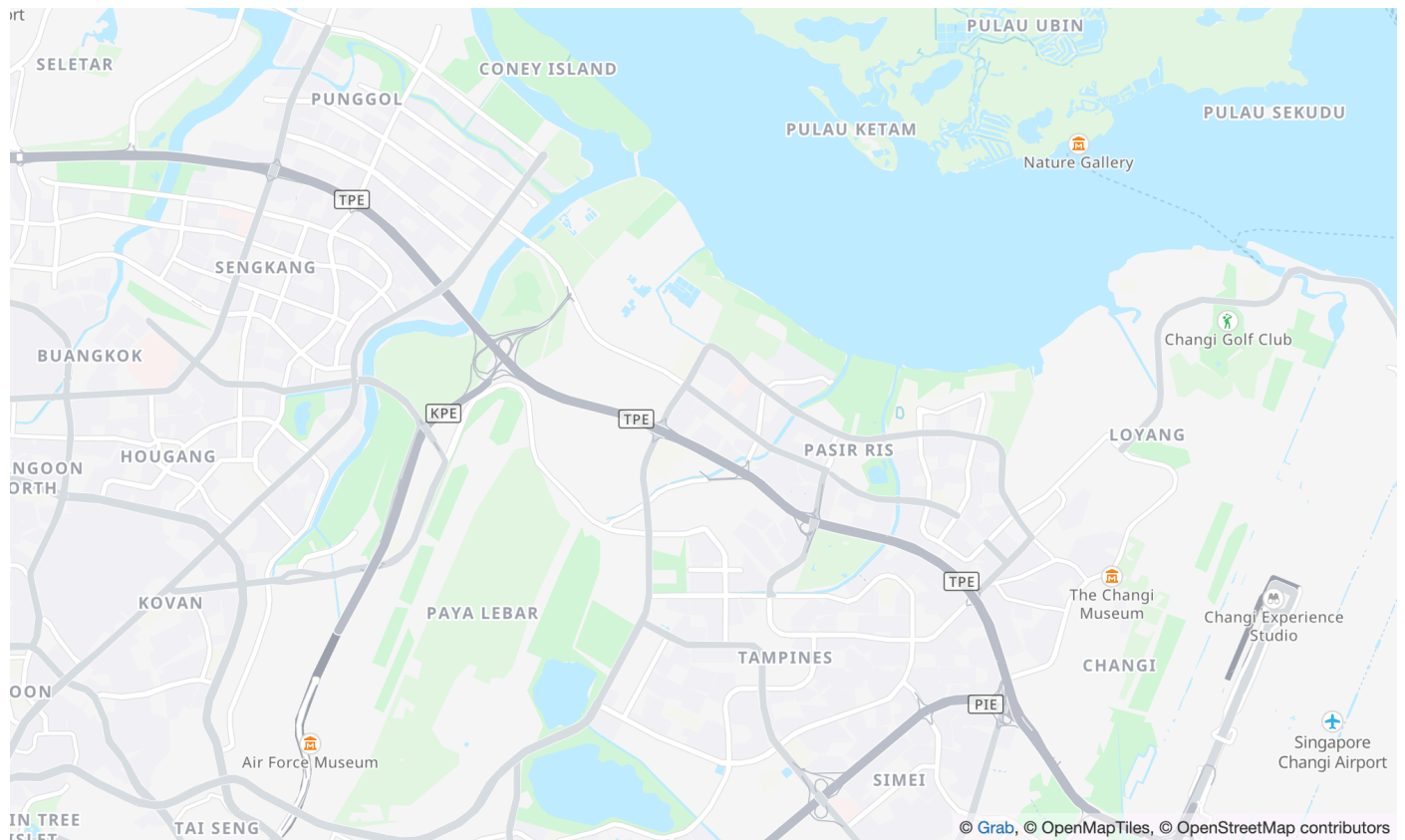
Amazon Location Service unterstützt beim [Erstellen einer Kartenressource die folgenden Grab-Kartenstile](#):

Note

Grab-Kartenstile, die in diesem Abschnitt nicht aufgeführt sind, werden derzeit nicht unterstützt.

Grab Standard Light Map

Schnappen Sie sich die Standard-Lichtkarte



Name des Kartenstils: `VectorGrabStandardLight`

Die Standard-Grundkarte von Grab mit detaillierter Farbgebung der Landnutzung, Gebietsnamen, Straßen, Sehenswürdigkeiten und Sehenswürdigkeiten für Südostasien.

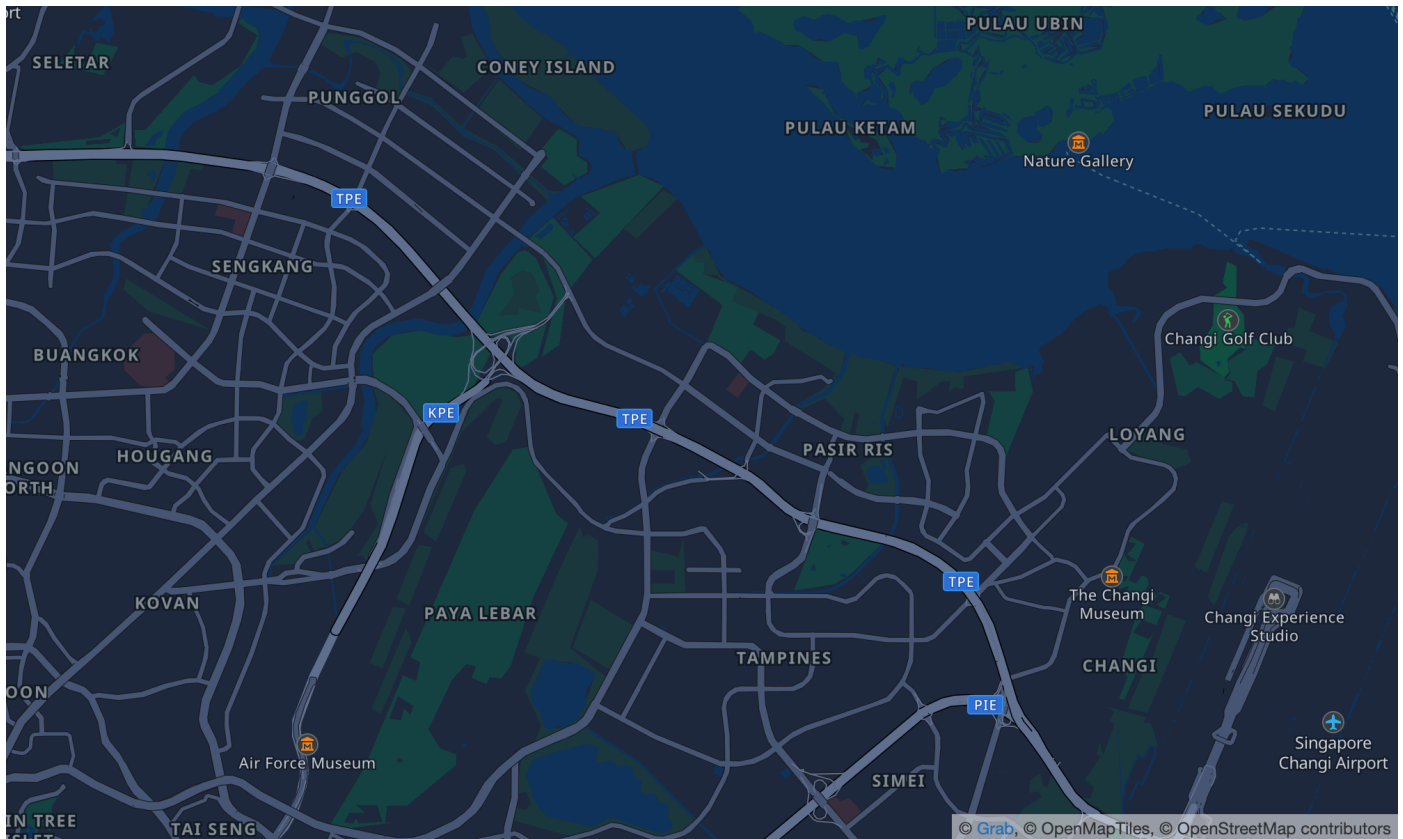
Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Noto Sans Regular
- Noto Sans Medium
- Noto Sans Bold

Grab Standard Dark Map

Schnappen Sie sich die dunkle Standardkarte



Name des Kartenstils: `VectorGrabStandardDark`

Grab's dunkle Variante der Standard-Grundkarte mit detaillierter Farbgebung der Landnutzung, Gebietsnamen, Straßen, Sehenswürdigkeiten und Sehenswürdigkeiten für Südostasien.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Noto Sans Regular
- Noto Sans Medium
- Noto Sans Bold

Berichterstattung: Schnappen

Sie können Grab als Datenanbieter verwenden, um Abfragen für Geokodierung, umgekehrte Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindexressource erstellen](#), oder um

Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie eine [Routenberechnungsressource erstellen](#).

Länder/Regionen und abgedecktes Gebiet

Grab bietet Karten nur für Südostasien und ist nur in der Region Asien-Pazifik (Singapur) verfügbar (ap-southeast-1).

Grab bietet detaillierte Daten für die folgenden Länder/Regionen:

- Malaysia
- Philippinen
- Thailand
- Singapur
- Vietnam
- Indonesien
- Myanmar
- Kambodscha

Note

Außerhalb dieser Bereiche liefern die Amazon Location Service Service-Ressourcen, die mit Grab als Datenanbieter erstellt wurden, keine Ergebnisse. Dazu gehören Suchergebnisse oder Routen.

Die Karten von Grab liegen innerhalb der folgenden Grenzen:

- Süden — Breitengrad -21,943045533438166
- West — Längengrad 90,0
- Norden — Breitengrad 31,952162238024968
- Ost — Längengrad 146,25

Für die Zoomstufen 1—4 umfasst Grab eine globale Abdeckung. Für die Zoomstufen 5 und darunter werden Kartenkacheln nur innerhalb dieses begrenzten Felds bereitgestellt.

Note

Außerhalb dieses begrenzten Bereichs geben die Kartenressourcen von Amazon Location Service, die mit Grab als Datenanbieter erstellt wurden, keine Kartenkacheln zurück. Um zu vermeiden, dass in Ihrer Anwendung 404-Fehler auftreten, können Sie die Karte mit einem Begrenzungsrahmen einschränken, wie unter beschrieben. [Festlegen von Erweiterungen für eine Zuordnung mit MapLibre](#)

Schnappen Sie sich die Routenplanungsmodi

Für die Routenplanung bietet Grab Auto - und Motorrad-Routing für alle zuvor aufgeführten Länder/Regionen.

Grab unterstützt kein Lkw-Routing.

Für Fahrrad - und Wanderwege unterstützt Grab die folgenden Städte:.

- Singapur
- Jakarta
- Manila
- Klang-Tal
- Bangkok
- Ho-Chi-Minh-Stadt
- Hanoi

Nutzungsbedingungen und Datenzuweisung: Grab

Wenn Sie die Daten von Grab verwenden, müssen Sie alle geltenden gesetzlichen Anforderungen einhalten, einschließlich der für Grab und AWS geltenden Lizenzbedingungen.

Weitere Informationen zu den AWS Anforderungen finden Sie in den [AWS-Servicebedingungen](#).

Informationen zu den GrabMaps Richtlinien für die Zuordnung finden Sie in Abschnitt 9.23 der [Datenzuordnungen und Nutzungsbedingungen](#) von Grab.

Fehlerberichterstattung für Daten GrabMaps

Wenn Sie ein Problem mit den Daten von GrabMaps haben und Fehler oder Unstimmigkeiten melden möchten, [wenden Sie sich an den AWS technischen Support](#).

HERE Technologies

Amazon Location Service nutzt die Ortungsdienste von HERE Technologies, um AWS Kunden dabei zu unterstützen, Karten effektiv zu nutzen, zu geokodieren und Routen zu berechnen. Die Standortdaten von HERE bieten eine standortbezogene Plattform, die offen, sicher und privat ist. Wenn Sie sich für HERE-Standortdaten entscheiden, wählen Sie genaue, aktuelle und robuste Daten aus, die nativ in der Cloud bereitgestellt werden. AWS

Weitere Informationen zu den Funktionen finden Sie [HIER](#) zu den Datenanbietern von Amazon Location Service.

Themen

- [Kartenstile von HERE](#)
- [Abdeckung: HIER](#)
- [Nutzungsbedingungen und Datenzuweisung: HIER](#)
- [Fehler bei der Meldung an HERE](#)

Kartenstile von HERE

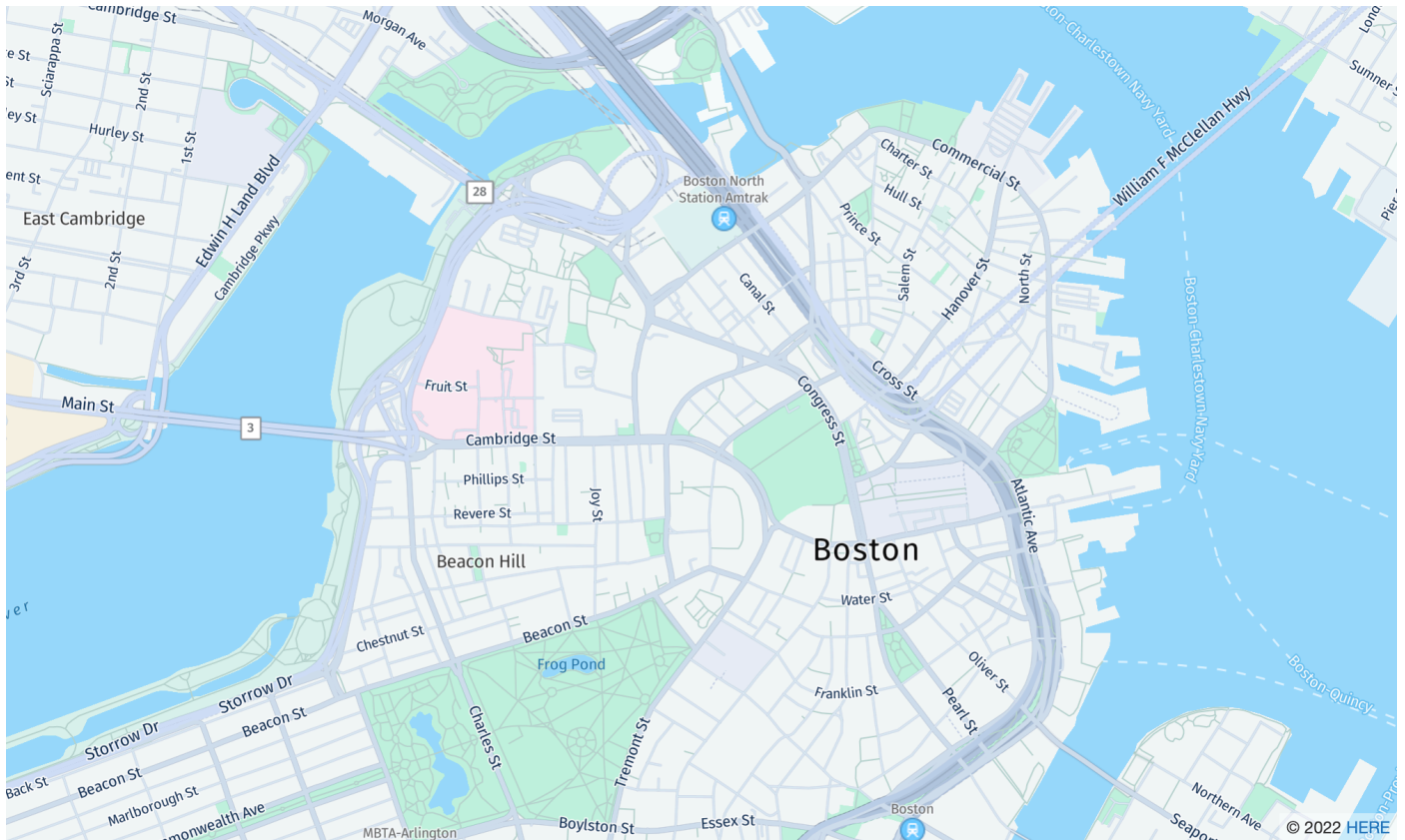
Amazon Location Service unterstützt die folgenden HERE-Kartenstile bei der [Erstellung einer Kartenressource](#):

Note

HERE-Kartenstile, die in diesem Abschnitt nicht aufgeführt sind, werden derzeit nicht unterstützt.

HERE Explore

[HIER Erkunden](#)



Name des Kartenstils: VectorHereExplore

HIER Erkunden

Eine detaillierte, neutrale Basiskarte der Welt. Die Straßenkarte umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen. Enthält eine vollständig gestaltete Karte von Japan.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Fira GO Kursiv
- Fira GO Regulär
- Fira GO Bold
- Noto ohne CJK JP Licht
- Noto Sans CJK JP Regulär
- Noto Sans CJK JP Gold

HERE Imagery

HIER Bilder



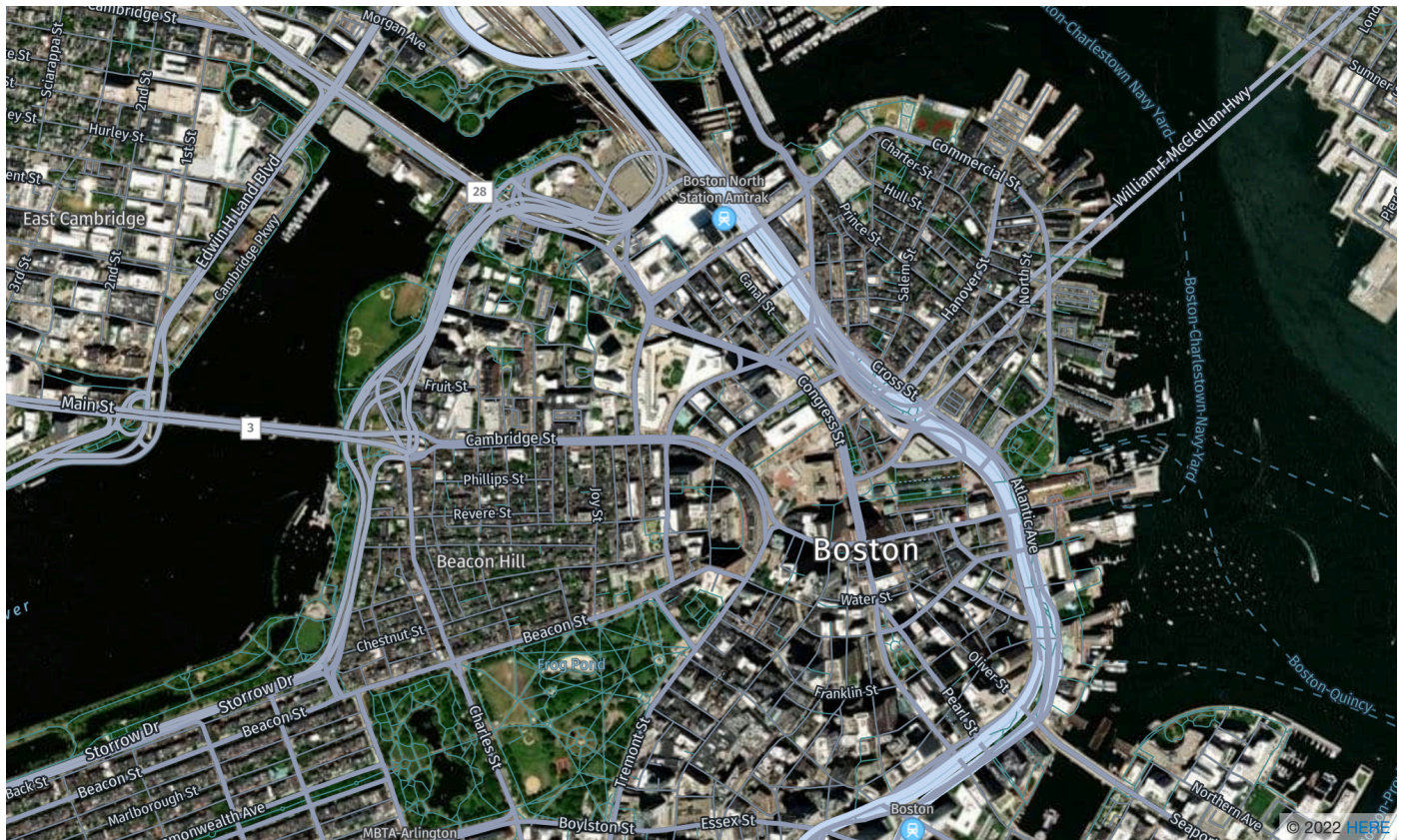
Name des Kartenstils: `RasterHereExploreSatellite`

HIER Bildmaterial

HERE Imagery bietet hochauflösende Satellitenbilder mit globaler Abdeckung.

HERE Hybrid

HIER Hybrid



Name des Kartenstils: HybridHereExploreSatellite

HIER Hybrid

Der HERE Hybrid-Stil zeigt das Straßennetz, Straßennamen und Stadtbeschriftungen auf Satellitenbildern an. Dieser Stil überlagert zwei Kartenkacheln: das Satellitenbild (Raster-Kachel) im Hintergrund und das Straßennetz und die Beschriftungen (Vektorkachel) darüber. Dieser Stil ruft automatisch sowohl die Raster- als auch die Vektorkacheln ab, die zum Rendern der Karte erforderlich sind.

Note

Hybridstile verwenden beim Rendern der angezeigten Karte sowohl Vektor- als auch Rasterkacheln. Dies bedeutet, dass mehr Kacheln abgerufen werden als bei alleiniger Verwendung von Vektor- oder Rasterkacheln. Ihre Gebühren beinhalten alle abgerufenen Kacheln.

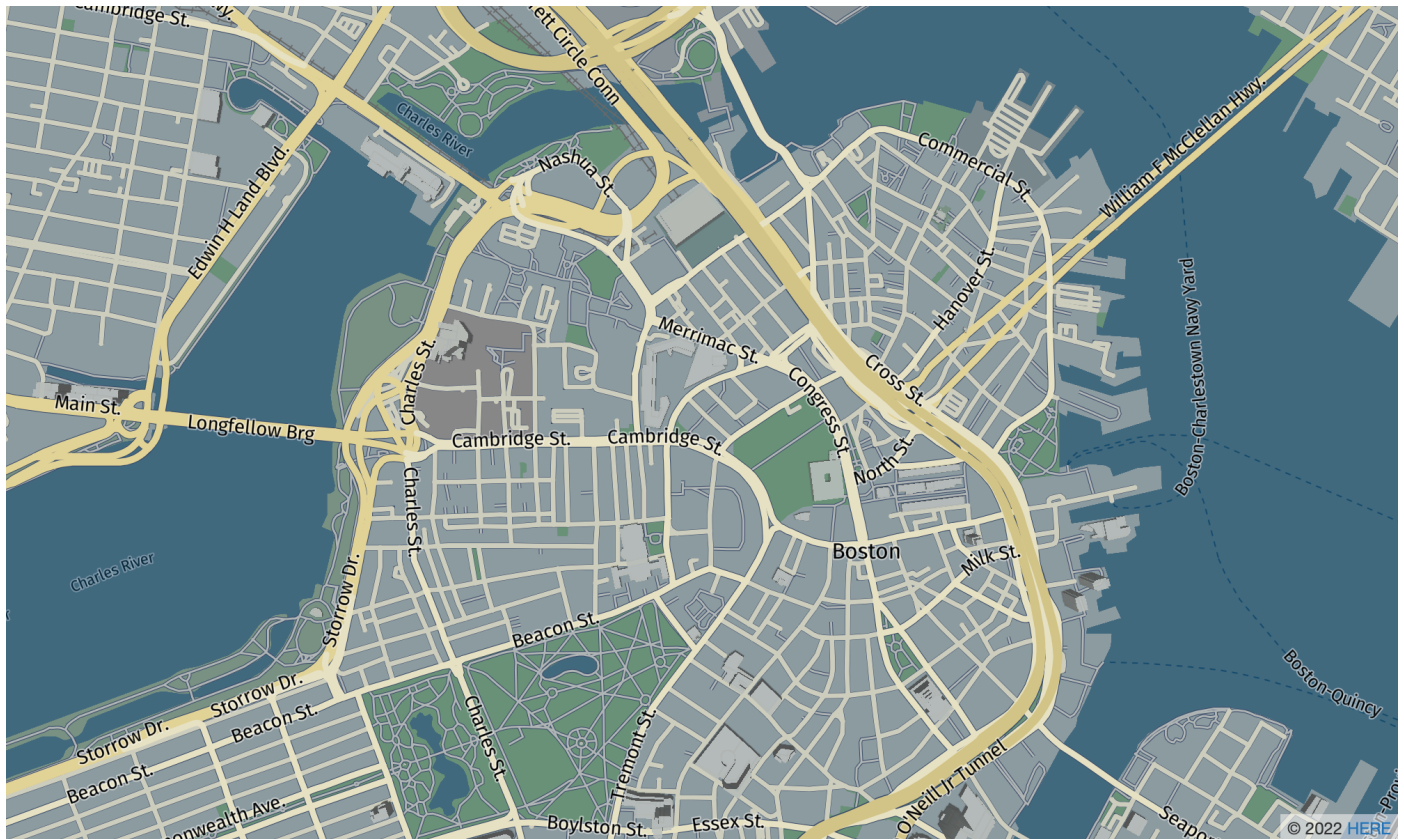
Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Fira GO Kursiv
- Fira GO Regulär
- Fira GO Bold
- Noto ohne CJK JP Licht
- Noto Sans CJK JP Regulär
- Noto Sans CJK JP Gold

HERE Contrast (Berlin)

HIER Kontrast (Berlin)



Name des Kartenstils: VectorHereContrast

HIER Contrast (Berlin)

Eine detaillierte Basiskarte der Welt, die 3D- und 2D-Rendering kombiniert. Die Straßenkarte mit hohem Kontrast umfasst Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

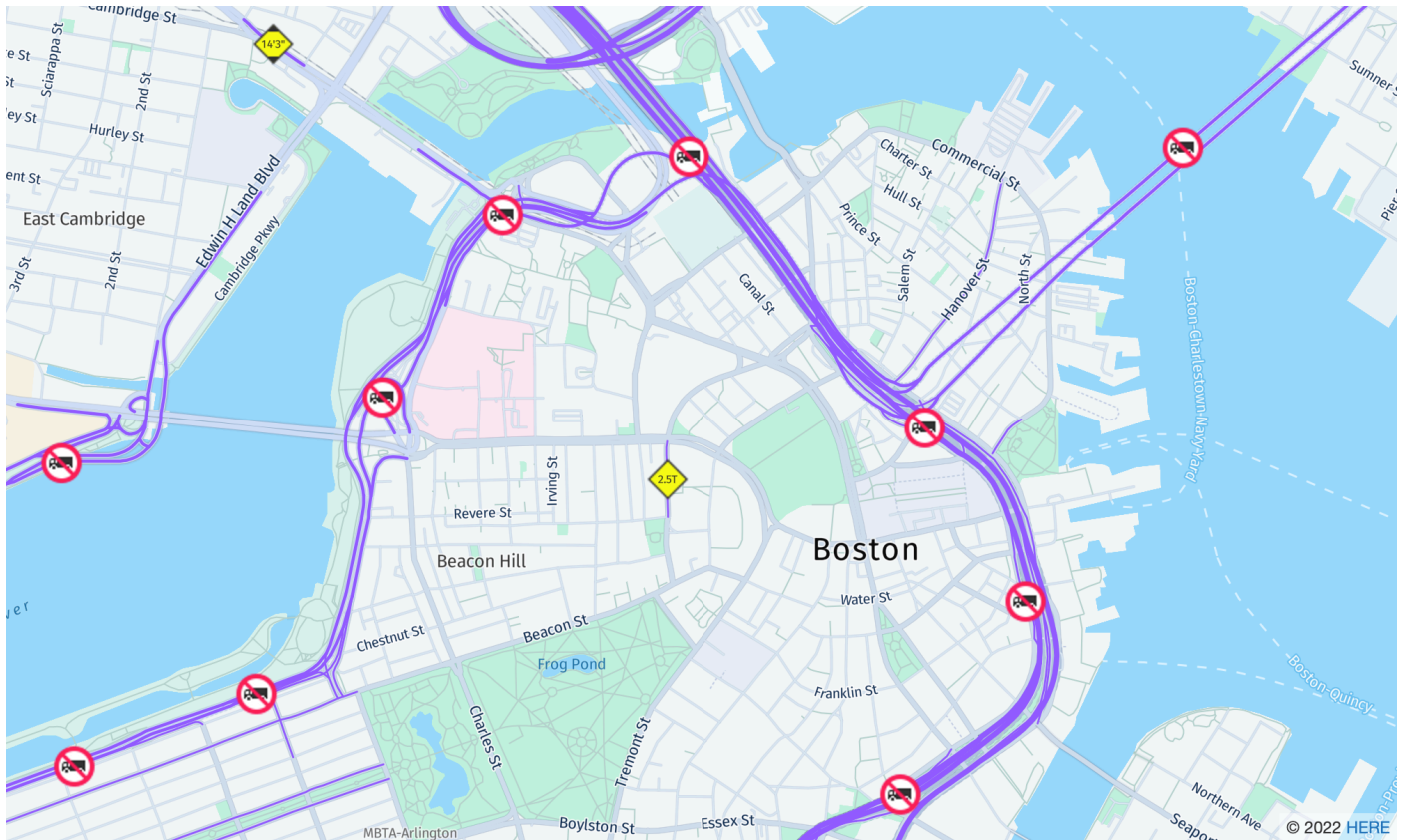
- Fira GO Regulär
- Fira GO Bold

Note

Dieser Stil wurde in `VectorHereBerlin` (HERE Berliner Karten) umbenannt. `VectorHereBerlin` ist veraltet, funktioniert aber weiterhin in Anwendungen, die es verwenden.

HERE Explore Truck

HIER Erkunden Sie Truck



Name des Kartenstils: VectorHereExploreTruck

HIER Erkunden Sie Truck

Eine detaillierte, neutrale Basiskarte der Welt. Die Straßenkarte baut auf dem HERE Explore-Stil auf und hebt Streckenbeschränkungen und Attribute (einschließlich Breite, Höhe und Gefahrgut) mit Symbolen und Symbolen hervor, um Anwendungsfälle in Transport und Logistik zu unterstützen.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Fira GO Kursiv
- Fira GO Regulär
- Fira GO Bold
- Noto ohne CJK JP Licht
- Noto Sans CJK JP Regulär

- Noto Sans CJK JP Gold

Weitere Informationen zur Qualität der Kartendaten in verschiedenen Regionen der Welt finden Sie unter [Kartenabdeckung von HERE](#).

Abdeckung: HIER

Sie können HERE als Datenanbieter verwenden, um Abfragen für Geokodierung, umgekehrte Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindex-Ressource erstellen](#), oder um Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie eine [Routenberechnungsressource erstellen](#).

HERE bietet unterschiedliche Datenqualitätsniveaus in verschiedenen Regionen der Welt. Weitere Informationen zur Abdeckung in Ihrer Interessenregion finden Sie im Folgenden:

- [Abdeckung durch Geokodierung HIER](#)
- [Abdeckung der Fahrzeugrouten von HERE](#)
- [HIER deckt das Lkw-Routing ab](#)

Nutzungsbedingungen und Datenzuweisung: HIER

Bevor Sie HERE-Daten verwenden, stellen Sie sicher, dass Sie alle geltenden rechtlichen Anforderungen einhalten können, einschließlich der für HERE und AWS geltenden Lizenzbedingungen. Aufgrund von Lizenzbeschränkungen dürfen Sie HERE nicht zum Speichern von Geokodierungsergebnissen für Standorte in Japan verwenden.

Informationen zu den AWS Anforderungen finden Sie in den [AWS-Servicebedingungen](#).

Weitere Informationen zu den Zuordnungsrichtlinien von HERE finden Sie in Abschnitt 2 der [Nutzungsbedingungen für Lieferanten von HERE Technologies, die sich auf den Standort und andere Inhalte beziehen](#).

Fehler bei der Meldung an HERE

Um Kartenfehler und Unstimmigkeiten an HERE zu melden, gehe zu <https://www.here.com/contact> und wähle Kartenfehler melden.

Daten öffnen

Amazon Location Service bietet über den Open Data-Anbieter Zugriff auf Open-Source-Kartendaten. Open Data bietet globale Grundkarten, die auf der [Daylight-Kartenverteilung](#) von [OpenStreetMap \(OSM\)](#), [Natural Earth](#) und anderen offenen Datenquellen basieren. Die bereitgestellten Karten sind so konzipiert, dass sie verschiedene Anwendungen und Anwendungsfälle unterstützen, darunter Logistik und Lieferung sowie Datenvisualisierung in Web- und Mobilumgebungen. Mit über einer Million Kartenherstellern aktualisiert die OSM-Community täglich Hunderttausende von Funktionen. Amazon Location Service nimmt diese Änderungen regelmäßig auf.

Weitere Informationen zu den Funktionen finden Sie unter [Open Data](#) on Amazon Location Service Service-Datenanbietern.

Themen

- [Map-Stile für Open Data](#)
- [Geltungsbereich: Offene Daten](#)
- [Nutzungsbedingungen und Datenzuweisung: Open Data](#)
- [Fehler melden und zu Open Data beitragen](#)

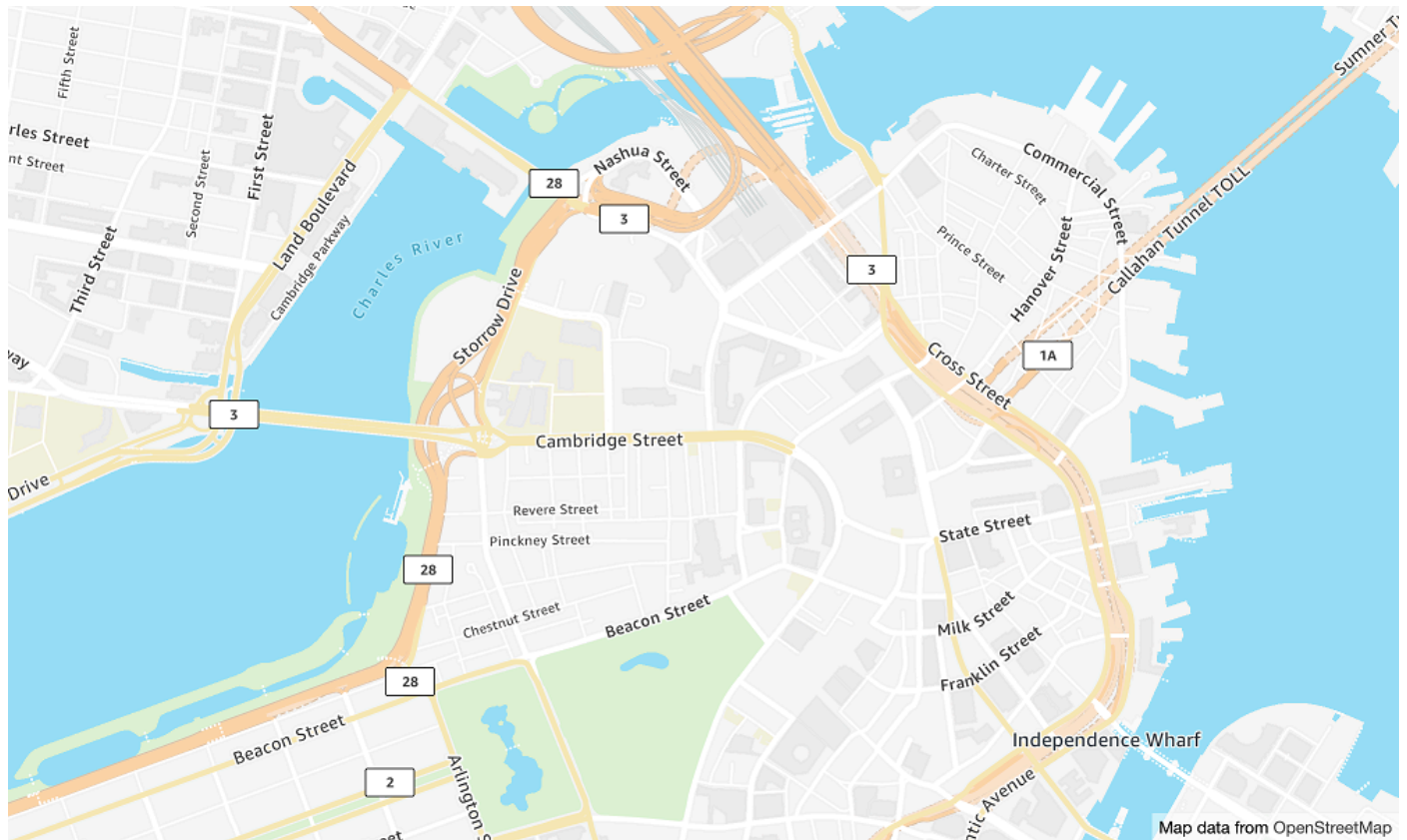
Map-Stile für Open Data

Amazon Location Service unterstützt beim [Erstellen einer Kartenressource die folgenden Kartenstile](#):

Open Data-Kartenstile unterstützen alternative Kartenstile [Politische Ansichten](#).

Open Data Standard Light

Öffnen Sie Data Standard Light



Name des Kartenstils: `VectorOpenDataStandardLight`

Dies bietet eine detaillierte Grundkarte für die Welt im Light-Map-Stil, die für die Verwendung auf Websites und mobilen Anwendungen geeignet ist. Dazu gehören Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen.


Diese Grundkarte basiert auf der [Kartenverteilung von OSM Daylight](#), die von OpenStreetMap (OSM-) Mitwirkenden zusammengestellt wurde. Die OSM-Community umfasst über 1,8 Millionen Mitwirkende, die täglich mehr als 500.000 Funktionen aktualisieren. Amazon Location Service nimmt diese Änderungen regelmäßig auf.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär

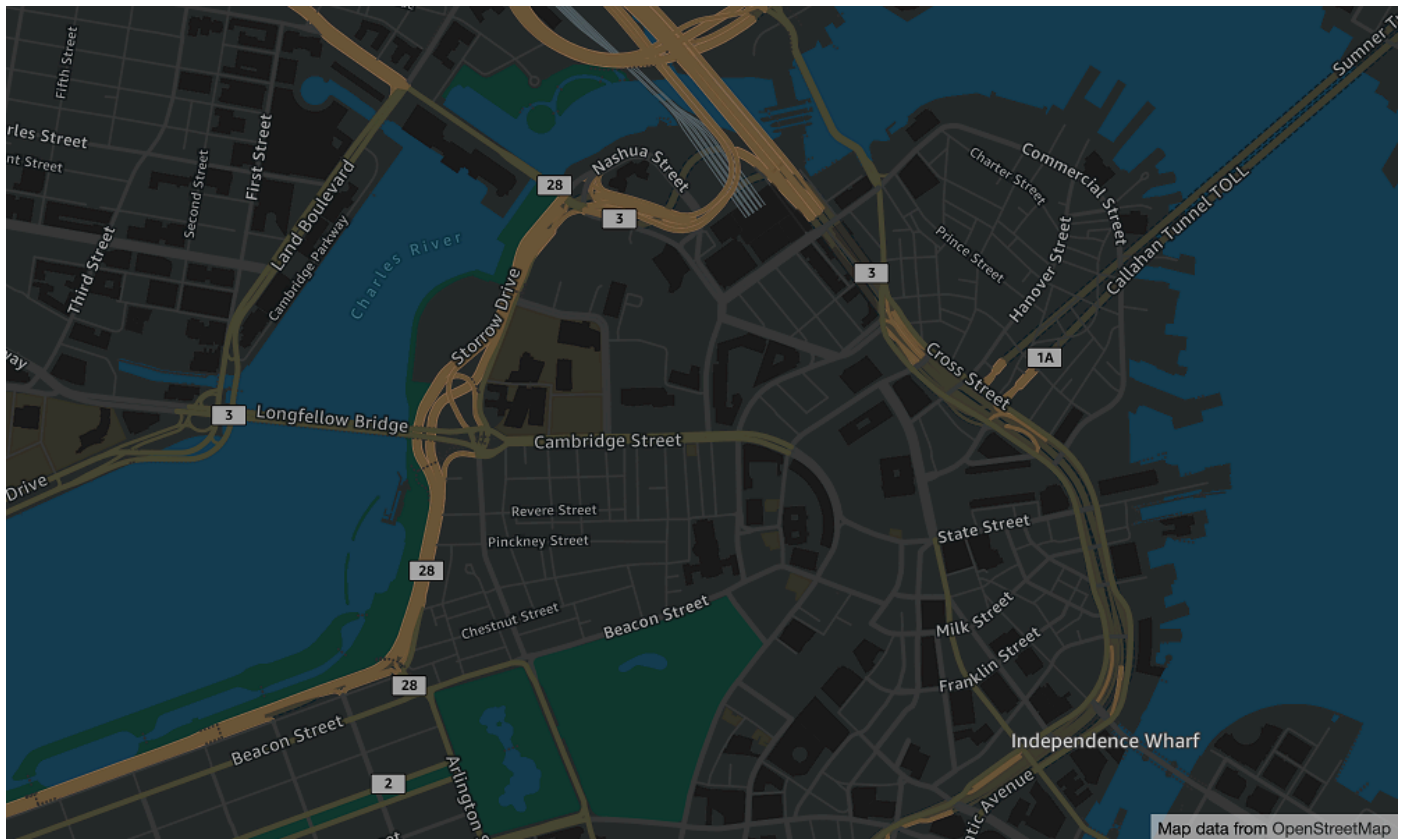
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regulär Kursiv, Noto Sans Kursiv
- Amazon Ember Regulär, Noto Sans Regulär
- Amazon Ember Regulär, Noto Sans Regulär, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär, Noto Sans Arabic Condensed Regulär
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Bei den von verwendeten Schriften VectorOpenDataStandardLight handelt es sich um kombinierte Schriften, die für die meisten Glyphen verwendet werden, aber Amazon Ember Noto Sans für Glyphen, die von nicht unterstützt werden. Amazon Ember

Open Data Standard Dark

Offener Datenstandard Dark



Name des Kartenstils: `VectorOpenDataStandardDark`

Dies ist ein Kartenstil mit dunklem Thema, der eine detaillierte Grundkarte für die Welt bietet und sich für die Verwendung auf Websites und mobilen Anwendungen eignet. Dazu gehören Autobahnen, Hauptstraßen, Nebenstraßen, Eisenbahnen, Wasserspiele, Städte, Parks, Sehenswürdigkeiten, Gebäudegrundrisse und Verwaltungsgrenzen.


Diese Grundkarte basiert auf der [Kartenverteilung von OSM Daylight](#), die von OpenStreetMap (OSM-) Mitwirkenden zusammengestellt wurde. Die OSM-Community umfasst über 1,8 Millionen Mitwirkende, die täglich mehr als 500.000 Funktionen aktualisieren. Amazon Location Service nimmt diese Änderungen regelmäßig auf.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär

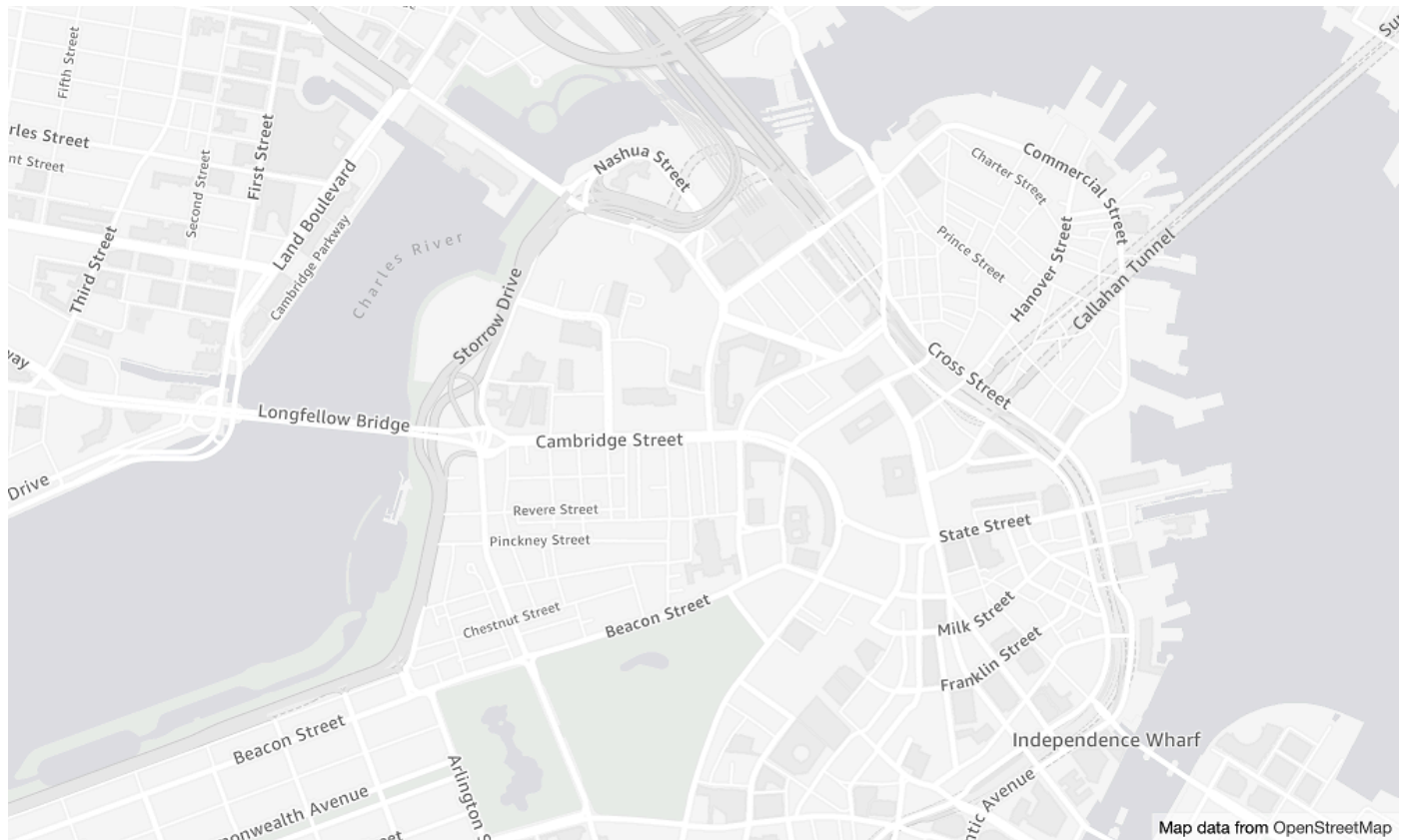
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regulär Kursiv, Noto Sans Kursiv
- Amazon Ember Regulär, Noto Sans Regulär
- Amazon Ember Regulär, Noto Sans Regulär, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär, Noto Sans Arabic Condensed Regulär
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Bei den von verwendeten Schriften VectorOpenDataStandardDark handelt es sich um kombinierte Schriften, die für die meisten Glyphen verwendet werden, aber Amazon Ember Noto Sans für Glyphen, die von nicht unterstützt werden. Amazon Ember

Open Data Visualization Light

Öffnen Sie Data Visualization Light



Name des Kartenstils: `VectorOpenDataVisualizationLight`

Dies ist ein Stil mit hellen Motiven, gedämpften Farben und weniger Funktionen, der das Verständnis überlagerter Daten erleichtert.


Diese Grundkarte basiert auf der [Kartenverteilung von OSM Daylight](#), die von OpenStreetMap Mitwirkenden (OSM) zusammengestellt wurde. Die OSM-Community umfasst über 1,8 Millionen Mitwirkende, die täglich mehr als 500.000 Funktionen aktualisieren. Amazon Location Service nimmt diese Änderungen regelmäßig auf.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär
- Amazon Ember Medium, Noto Sans Medium

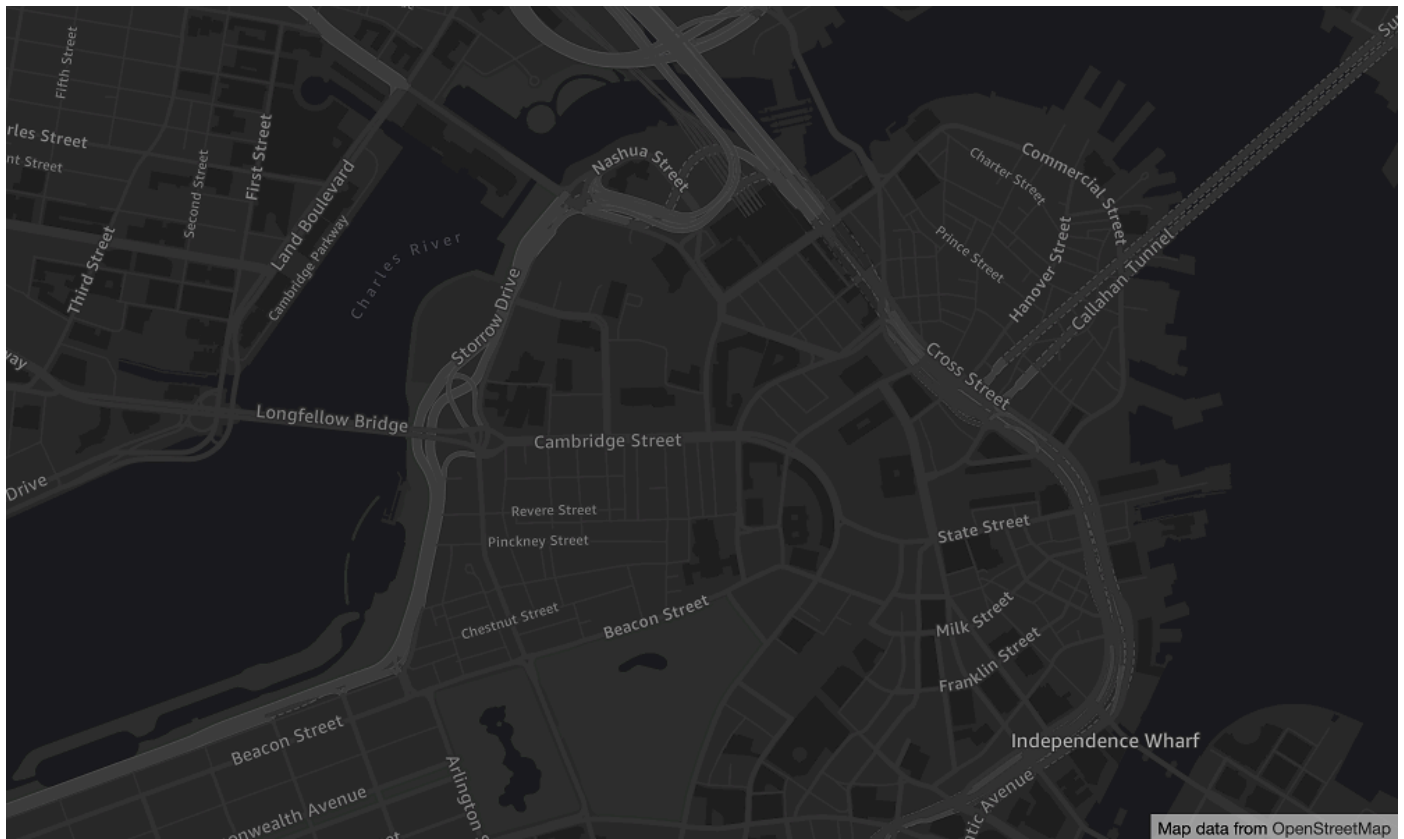
- Amazon Ember Regulär Kursiv, Noto Sans Kursiv
- Amazon Ember Regulär, Noto Sans Regulär
- Amazon Ember Regulär, Noto Sans Regulär, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär, Noto Sans Arabic Condensed Regulär
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Bei den von verwendeten Schriften `VectorOpenDataVisualizationLight` handelt es sich um kombinierte Schriften, die für die meisten Glyphen verwendet werden, aber Amazon Ember Noto Sans für Glyphen, die von nicht unterstützt werden. Amazon Ember

Open Data Visualization Dark

Öffnen Sie Data Visualization Dark



Name des Kartenstils: `VectorOpenDataVisualizationDark`

Dies ist ein Stil mit dunklen Themen, gedämpften Farben und weniger Funktionen, der das Verständnis überlagerter Daten erleichtert.

Diese Grundkarte basiert auf der [Kartenverteilung von OSM Daylight](#), die von Mitwirkenden (OSM) zusammengestellt wurde. OpenStreetMap Die OSM-Community umfasst über 1,8 Millionen Mitwirkende, die täglich mehr als 500.000 Funktionen aktualisieren. Amazon Location Service nimmt diese Änderungen regelmäßig auf.

Schriftarten

Amazon Location stellt Schriften bereit mit [GetMapGlyphs](#). Die folgenden Schriftstapel sind für diese Karte verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär
- Amazon Ember Medium, Noto Sans Medium

- Amazon Ember Regulär Kursiv, Noto Sans Kursiv
- Amazon Ember Regulär, Noto Sans Regulär
- Amazon Ember Regulär, Noto Sans Regulär, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regulär
- Amazon Ember Condensed RC Regulär, Noto Sans Regulär, Noto Sans Arabic Condensed Regulär
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

Note

Bei den von verwendeten Schriften `VectorOpenDataVisualizationDark` handelt es sich um kombinierte Schriften, die für die meisten Glyphen verwendet werden, aber `Amazon Ember` `Noto Sans` für Glyphen, die von nicht unterstützt werden. `Amazon Ember`

Geltungsbereich: Offene Daten

Open Data umfasst Karten mit globaler Abdeckung, die mit einer [Kartenressource von Amazon Location Service](#) gerendert werden können.

Note

Open Data ist nur für die Verwendung mit Kartenressourcen von Amazon Location Service bestimmt. Sie können Open Data nicht als Datenanbieter verwenden, um Abfragen für Geokodierung, umgekehrte Geokodierung und Suchen oder Abfragen zur Berechnung einer Route zu unterstützen.

Nutzungsbedingungen und Datenzuweisung: Open Data

Bevor Sie Open Data verwenden, stellen Sie sicher, dass Sie alle geltenden rechtlichen Anforderungen einhalten können, einschließlich der für Open Data geltenden Lizenzbedingungen und AWS.

Weitere Informationen zu den AWS Anforderungen finden Sie in den [AWS-Servicebedingungen](#).

[Informationen zu den Richtlinien zur Zuordnung von Open Data finden Sie OpenStreetMap in den Abschnitten Copyright und Lizenz und OpenStreetMap Lizenz-/Attributionsrichtlinien.](#)

Fehler melden und zu Open Data beitragen

OpenStreetMap (OSM) und Natural Earth sind von der Community betriebene Open-Data-Projekte. Wenn Sie auf ein Problem mit den Daten stoßen, können Sie die Fehler melden oder direkt Korrekturen oder Vorschläge einbringen.

- Um einen Fehler zu melden oder einen Vorschlag in OSM zu machen, kannst du eine Notiz auf der Karte erstellen. Dies ist ein Kommentar zur Map, der Mitwirkenden dabei hilft, Korrekturen an der Map vorzunehmen. Sie erstellen Notizen über die [OpenStreetMap Website](#). Weitere Informationen zu Notizen finden Sie unter [Notizen](#) im OpenStreetMap Wiki.
- Weitere Informationen dazu OpenStreetMap, wie Sie direkt dazu beitragen können, Orte hinzuzufügen und Fehler zu beheben, finden Sie im OpenStreetMap Wiki unter [Contribute-Kartendaten](#).
- Um eine Korrekturanfrage für Daten in Natural Earth einzureichen, können Sie ein Problem über die [Natural Earth-Website](#) einreichen.

Note

Die Korrektur von Fehlern in OpenStreetMap kann schnell erfolgen, es kann jedoch einige Zeit dauern, bis Korrekturen in der Tageslichtkartenverteilung der OSM-Daten erscheinen, die vom Open Data-Anbieter verwendet wird. Die [Daylight Map Distribution-Website](#) bietet weitere Informationen über den Prozess. Darüber hinaus aktualisiert Amazon Location Service die in Amazon Location Service verwendeten Kartendaten ungefähr monatlich.

Funktionen nach Datenanbieter

In diesem Abschnitt werden die in Amazon Location Service verfügbaren Funktionen beschrieben, kategorisiert nach Datenanbietern.

Die folgende Tabelle bietet einen allgemeinen Überblick über die Funktionen.

Datenanbieter	Geografischer Geltungsbereich	Abdeckung der Funktionen	AWS-Region
Esri	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Schnapp dir	Südost-Asien	Karten, Orte, Routen	Nur Asien-Pazifik (Singapur). ap-southeast-1
HIER	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Daten öffnen	Global	Zuordnungen	Alle Regionen , in denen Amazon Location verfügbar ist.


Auf den folgenden Registerkarten werden Details zu den einzelnen Funktionsbereichen angezeigt.

Map Features

Die folgende Tabelle zeigt die Kartenfunktionen nach Datenanbietern. Weitere Informationen zu Kartenkonzepten finden Sie unter [Zuordnungen](#).

Datenanbieter	Unterstützte Kartentypen	Vektor-Zoomstufen	Raster-Zoomstufen
Esri	Vektor Raster (Bilddaten) Weitere Informationen finden Sie unter Kartenstile von Esri .	0-15	0-23
Schnapp	Vektor	0-14	Keine

Datenanbieter	Unterstützte Kartentypen	Vektor-Zoomstufen	Raster-Zoomstufen
	(Nur Südostasien) Weitere Informationen finden Sie unter Schnappen Sie sich Kartenstile .		
HIER	Vektor Raster (Bilder) Hybrid Weitere Informationen finden Sie unter Kartenstile von HERE .	1-17	0-19
Daten öffnen	Vektor Weitere Informationen finden Sie unter Map-Stile für Open Data .	0-15	Keine

 Note

Die Zoomstufen stellen die Höchst- und Mindesteinstellungen dar, wie sie in den APIs der einzelnen Anbieter definiert sind. Verschiedene Bereiche der Karte können unterschiedliche Höchstwerte haben. Beispielsweise haben Meereskacheln möglicherweise weniger detaillierte Zoomstufen als Bereiche in Großstädten. MapLibre (und andere Karten-Rendering-Engines) ermöglichen es Ihnen, minimale und maximale Zoomstufen festzulegen. Außerdem werden die Zoomstufen des Datenanbieters in einem Bereich berücksichtigt, sodass Sie keinen Code schreiben müssen, um diese Abweichungen zu beheben.

Places and Search

In der folgenden Tabelle sind die Orts- und Suchfunktionen nach Datenanbietern aufgeführt. Weitere Informationen zu Ortskonzepten finden Sie unter [Suche nach Orten](#).

Datenanbieter	Geokodierung	Umgekehrte Geokodierung	AutoVervollständigen	GetPlace
Esri	Alle Funktionen, außer: PlaceId	Alle Funktionen, außer: TimeZone PlaceId	Alle Funktionen	Alle Funktionen
Schnapp	Alle Funktionen, außer: Art der Einheit Kategorien werden nicht unterstützt	Alle Funktionen	Alle Funktionen	Alle Funktionen, außer: Art der Einheit SubMunicipality
HIER	Alle Funktionen, außer: Nummer der Einheit Art der Einheit Relevanz Zusätzliche Einschränkungen beim Filtern	Alle Funktionen	Alle Funktionen	Alle Funktionen, außer: Nummer der Einheit Art der Einheit SubMunicipality
Daten öffnen	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt	Unterstützt nur:

Datenanbieter	Geokodierung	Umgekehrte Geokodierung	AutoVervollständigen	GetPlace
				SubMunicipality

Route features

Die folgende Tabelle zeigt die Routenfunktionen nach Datenanbietern. Weitere Informationen zu Routenkonzepten finden Sie unter [Routen](#). Ausführlichere Beschreibungen der Einschränkungen der Routenmatrix finden Sie unter [Einschränkungen bei der Ankunft und bei Zielpositionen](#).

Datenanbieter	Reisemodi	Route berechnen	Routenmatrix
Esri	Auto, LKW, zu Fuß	Abflug und Ziel müssen nicht weiter als 400 km voneinander entfernt sein. Die gesamte Reisezeit darf nicht mehr als 400 Minuten betragen. ArrivalTime wird nicht unterstützt.	Bis zu 10 Start- und Zielpositionen. In Korea nicht unterstützt. Alle Abflug- und Zielpaare müssen nicht weiter als 400 km voneinander entfernt sein.
Schnapp	Auto, Motorrad. Wandern und Radfahren in ausgewählten Städten .	Keine Entfernungsbegrenzungen.	Bis zu 350 Start- und Zielpositionen.
HIER	Auto, LKW, zu Fuß	Keine Entfernungsbegrenzung. Routen, die mehr als 10 km außerhalb eines Kreises um die	Bis zu 350 Start- und Zielpositionen. Alle Start- und Zielpositionen

Datenanbieter	Reisemodi	Route berechnen	Routenmatrix
		Start- und Zielposition verlaufen, werden nicht berechnet.	müssen in einem Umkreis von 180 km liegen. Längere Strecken werden mit zusätzlichen Einschränkungen unterstützt.
Daten öffnen	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt

Nutzungsbedingungen und Datenzuweisung für Datenanbieter

Bevor Sie einen Datenanbieter verwenden, stellen Sie sicher, dass Sie alle geltenden gesetzlichen Anforderungen einhalten können, einschließlich der Lizenzbedingungen, die für die Nutzung des Anbieters gelten.

Weitere Informationen zu den AWS Anforderungen finden Sie in den [AWS-Servicebedingungen](#).

Wenn Sie einen Datenanbieter mit Ihren Amazon-Standortressourcen für Ihre Anwendung oder Dokumentation verwenden, stellen Sie sicher, dass Sie für jeden Datenanbieter, den Sie verwenden, Quellenangaben angeben.

Weitere Informationen zur Einhaltung der Vorschriften und zur Zuordnung der einzelnen Datenanbieter finden Sie in den folgenden Themen.

- Esri — [Nutzungsbedingungen und Datenzuweisung: Esri](#)
- Schnapp dir — [Nutzungsbedingungen und Datenzuweisung: Grab](#)
- HIER — [Nutzungsbedingungen und Datenzuweisung: HIER](#)
- Daten öffnen — [Nutzungsbedingungen und Datenzuweisung: Open Data](#)

Amazon-Standortregionen und Endpunkte

Amazon Location ist in den folgenden AWS Regionen verfügbar:

Regionen

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS
USA Ost (Nord-Virginia)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS
Kanada (Zentral)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Frankfurt)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Europa (Irland)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Europa (London)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS
Europa (Stockholm)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
Südamerika (São Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (USA West)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

Note

Weitere Informationen zur Verwendung der Endpunkte in dieser Tabelle finden Sie im folgenden Abschnitt.

Endpunkte

Die allgemeine Syntax eines Amazon-Location-Regionalendpunkts lautet wie folgt:

```
protocol://service-code.geo.region-code.amazonaws.com
```

Innerhalb dieser Syntax verwendet Amazon Location die folgenden Service-Codes:

Service	Service-Code
Amazon Location Maps	Maps
Amazon Location Places	Orte
Amazon Location Geofences	Geofencing
Amazon Location Trackers	Nachverfolgung
Amazon Location Routes	Routen

Der regionale Endpunkt für Amazon Location Maps for US East (Nord-Virginia) wäre beispielsweise: `https://maps.geo.us-east-1.amazonaws.com`.

Endpunkte für den API-Betrieb

Die Syntax für einen Endpunkt der Amazon Location Service Service-Kontrollebene lautet wie folgt:

```
protocol://cp.service-code.geo.region-code.amazonaws.com
```

Die Aktionen auf der Kontrollebene für Amazon Location Service sind:

Service	Endpunkt	API-Operation
Amazon Location Maps	<code>https://cp.maps.geo. <i>region</i>.amazonaws.com</code>	CreateMap DeleteMap DescribeMap ListMaps UpdateMap

Service	Endpoint	API-Operation
Amazon Location Places	<code>https://cp.places.geo. <i>region</i> .amazonaws.com</code>	CreatePlaceIndex DeletePlaceIndex DescribePlaceIndex ListPlaceIndexes UpdatePlaceIndex
Amazon Location Geofences	<code>https://cp.geofencing.geo. <i>region</i> .amazonaws.com</code>	CreateGeofenceCollection DeleteGeofenceCollection DescribeGeofenceCollection ListGeofenceCollections UpdateGeofenceCollection
Amazon Location Trackers	<code>https://cp.tracking.geo. <i>region</i> .amazonaws.com</code>	CreateTracker DeleteTracker DescribeTracker UpdateTracker ListTrackers AssociateTrackerConsumer DisassociateTrackerConsumer ListTrackerConsumers

Service	Endpoint	API-Operation
Amazon Location Routes	<code>https://cp.routes.geo. <i>region</i> .amazonaws.com</code>	CreateRouteCalculator DeleteRouteCalculator DescribeRouteCalculator ListRouteCalculators UpdateRouteCalculator
Amazon-Standort-Metadaten	<code>https://cp.metadata.geo. <i>region</i> .amazonaws.com</code>	CreateKey DeleteKey DescribeKey ListKeys UpdateKey

Kontingente für Amazon Location Service

Dieses Thema bietet eine Zusammenfassung der Ratenlimits und Kontingente für Amazon Location Service.

Note

Wenn Sie ein höheres Kontingent benötigen, können Sie die Konsole Service Quotas verwenden, um [Kontingenterhöhungen für anpassbare Kontingente anzufordern](#). Wenn Sie eine Kontingenterhöhung beantragen, wählen Sie die Region aus, in der Sie die Kontingenterhöhung wünschen, da die AWS meisten Kontingente regionsspezifisch sind.

Service Quotas sind die maximale Anzahl von Ressourcen, die Sie pro AWS Konto und AWS Region haben können. Amazon Location Service lehnt zusätzliche Anfragen ab, die das Servicekontingent überschreiten.

Ratenlimits (Kontingente, die mit einer Rate von... beginnen)) sind die maximale Anzahl von Anfragen pro Sekunde mit einer Burst-Rate von 80 Prozent des Limits innerhalb eines beliebigen Teils der Sekunde, die für jeden API-Vorgang definiert wird. Amazon Location Service drosselt Anfragen, die das Ratenlimit des Vorgangs überschreiten.

Name	Standard	Anpas	Beschreibung
API-Schlüsselressourcen pro Konto	Jede unterstützte Region: 500	Nein	Die maximale Anzahl von API-Schlüsselressourcen (aktiv oder abgelaufen), die Sie pro Konto haben können.
Geofence-Sammlungsressourcen pro Konto	Jede unterstützte Region: 1 500	Ja	Die maximale Anzahl von Geofence-Sammlungsressourcen, die Sie pro Konto erstellen können.
Geofences pro Geofence-Sammlung	Jede unterstützte Region: 50 000	Nein	Die maximale Anzahl von Geofences, die Sie pro Geofence-Sammlung erstellen können.
Kartenressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Kartenressourcen, die Sie pro Konto erstellen können.
Ortsindexressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Ortsindexressourcen, die Sie pro Konto erstellen können.
Rate der AssociateTrackerConsumer API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von AssociateTrackerConsumer Anfragen, die Sie pro Sekunde stellen können. Darüber

Name	Standard	Anpas	Beschreibung
			hinausgehende Anfragen werden gedrosselt.
Rate der BatchDeleteDevicePositionHistory API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchDeleteDevicePositionHistory Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchDeleteGeofence API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchDeleteGeofence Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchEvaluateGeofences API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchEvaluateGeofences Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchGetDevicePosition API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchGetDevicePosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der BatchPutGeofence API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchPutGeofence Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchUpdateDevicePosition API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchUpdateDevicePosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CalculateRoute API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CalculateRoute Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CalculateRouteMatrix API-Anfragen	Jede unterstützte Region: 5 pro Sekunde	Ja	Die maximale Anzahl von CalculateRouteMatrix Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der CreateGeofenceCollection API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateGeofenceCollection Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateKey API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateKey Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateMap API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateMap Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreatePlaceIndex API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreatePlaceIndex Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der CreateRouteCalculator API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateRouteCalculator Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateTracker API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateTracker Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteGeofenceCollection API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteGeofenceCollection Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteKey API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteKey Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DeleteMap API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteMap Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeletePlaceIndex API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeletePlaceIndex Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteRouteCalculator API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteRouteCalculator Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteTracker API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteTracker Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DescribeGeofenceCollection API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeGeofenceCollection Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeKey API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeKey Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeMap API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeMap Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribePlaceIndex API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribePlaceIndex Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DescribeRouteCalculator API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeRouteCalculator Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeTracker API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeTracker Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DisassociateTrackerConsumer API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DisassociateTrackerConsumer Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ForecastGeofenceEvents API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von ForecastGeofenceEvents Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der GetDevicePosition API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetDevicePosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetDevicePositionHistory API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetDevicePositionHistory Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetGeofence API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetGeofence Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapGlyphs API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetMapGlyphs Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der GetMapSprites API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von GetMapSprites Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapStyleDescriptor API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von GetMapStyleDescriptor Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapTile API-Anfragen	Jede unterstützte Region: 500 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von GetMapTile Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetPlace API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von GetPlace Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListDevicePositions API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von ListDevicePositions Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListGeofenceCollections API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListGeofenceCollections Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListGeofences API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von ListGeofences Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListKeys API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListKeys Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListMaps API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListMaps Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListPlaceIndexes API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListPlaceIndexes Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListRouteCalculators API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListRouteCalculators Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListTagsForResource API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTagsForResource Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListTrackerConsumers API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTrackerConsumers Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListTrackers API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTrackers Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der PutGeofence API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von PutGeofence Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der SearchPlaceIndexForPosition API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForPosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der SearchPlaceIndexForSuggestions API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForSuggestions Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der SearchPlaceIndexForText API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForText Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der TagResource API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von TagResource Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UntagResource API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UntagResource Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der UpdateGeofenceCollection API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateGeofenceCollection Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateKey API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateKey Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateMap API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateMap Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdatePlaceIndex API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdatePlaceIndex Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der UpdateRouteCalculator API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateRouteCalculator Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateTracker API-Anfragen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateTracker Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der VerifyDevicePosition API-Anfragen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von VerifyDevicePosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Routenrechner-Ressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Routenrechner-Ressourcen, die Sie pro Konto erstellen können.
Tracker-Verbraucher pro Tracker	Jede unterstützte Region: 5	Nein	Die maximale Anzahl der Geofence-Sammlungen, mit der die Tracker-Ressource verknüpft werden können.

Name	Standard	Anpas	Beschreibung
Tracker-Ressourcen pro Konto	Jede unterstützte Region: 500	Ja	Die maximale Anzahl von Tracker-Ressourcen, die Sie pro Konto erstellen können.

Note

Mit Cloudwatch können Sie Ihre Nutzung anhand Ihrer Kontingente überwachen. Weitere Informationen finden Sie unter [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#).

Verwaltung Ihrer Amazon Location Service-Kontingente

Amazon Location Service ist in Service Quotas integriert, einen AWS Service, mit dem Sie Ihre Kontingente von einem zentralen Ort aus einsehen und verwalten können. Weitere Informationen zu Service Quotas finden Sie unter [Was sind Service Quotas](#) im Benutzerhandbuch für Service Quotas.

Mit Service Quotas können Sie ganz einfach den Wert Ihrer Amazon Location Service-Kontingente nachschlagen.

AWS Management Console

So zeigen Sie Amazon Location Service-Kontingente mit der Konsole an

1. Öffnen Sie die Service-Quotas-Konsole unter <https://console.aws.amazon.com/servicequotas/>.
2. Wählen Sie im Navigationsbereich AWS -Services.
3. Suchen Sie in der AWS Serviceliste nach Amazon Location und wählen Sie diesen aus.

In der Liste Service Quotas wird der Name der Service Quota, der angewendete Wert (falls verfügbar) und das AWS -Standardkontingent angezeigt. Zudem wird angezeigt, ob der Kontingentwert anpassbar ist.

4. Wählen Sie den Kontingentnamen, um zusätzliche Informationen zu einem Service Quota anzuzeigen, z. B. seine Beschreibung.

5. (Optional) Um eine Kontingenterhöhung zu beantragen, wählen Sie das Kontingent, das Sie erhöhen möchten, und dann Request quota increase (Kontingenterhöhung beantragen) aus, geben Sie die erforderlichen Informationen ein, und wählen Sie dann Request (Beantragen) aus.

Weitere Informationen zur Verwendung der Konsole mit Service Quotas finden Sie im [Benutzerhandbuch zu Service Quotas](#). Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch zu Service Quotas.

AWS CLI

Um die Kontingente für den Amazon Location Service einzusehen, verwenden Sie den AWS CLI

Führen Sie den folgenden Befehl aus, um die standardmäßigen Amazon-Standortkontingente anzuzeigen.

```
aws service-quotas list-aws-default-service-quotas \
  --query 'Quotas[*]'.
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
  --service-code geo \
  --output table
```

Weitere Informationen zur Verwendung von Service Quotas finden Sie in der [AWS CLI](#) [Befehlsreferenz für Dienstkontingente](#). Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter dem [request-service-quota-increase](#)-Befehl in der [AWS CLI - Befehlsreferenz](#).

Erste Schritte als Entwickler mit Amazon Location Service

Sie können Amazon Location Service verwenden, um geografisch relevante Funktionen für Apps in vielen verschiedenen Formfaktoren und Systemen bereitzustellen, darunter Backend-Webdienste, Webanwendungen und mobile Anwendungen. Es stehen zahlreiche Tools zur Verfügung, die Sie bei der Erstellung Ihrer Anwendungen unterstützen, darunter SDKs, Bibliotheken und Beispielcode.

Dieser Abschnitt enthält Informationen und Links, die Ihnen die ersten Schritte mit Amazon Location erleichtern. Insbesondere die folgenden Themen enthalten Informationen, die für Sie am hilfreichsten sein können:

- [Szenarien und Anwendungsfälle](#)— Eine Liste von Entwicklungsszenarien und wie Amazon Location Service Ihnen bei deren Umsetzung helfen kann.
- [Amazon Location SDKs und Tools](#) — Die Software Development Kits (SDKs) und Bibliotheken, die Sie bei der Programmierung mit Amazon Location unterstützen.
- [Amazon Location Service API-Referenz](#) — Ein Verweis auf die wichtigsten Amazon Location APIs, die im Lieferumfang des AWS SDK enthalten sind.
- [Codebeispiele](#) — Dieser Abschnitt enthält Beispiele, die Ihnen beim Einstieg oder beim Hinzufügen von Funktionen zu Ihrer bestehenden Anwendung helfen.
- [Schnellstart-Tutorial](#) — Dieses Tutorial zeigt Ihnen, wie Sie Ihre erste Anwendung erstellen. Es gibt Versionen des Tutorials zum Erstellen einer Webanwendung oder einer Android-basierten mobilen Anwendung.
- [Konzepte von Amazon Location Service](#)— In diesem Abschnitt dieses Handbuchs werden die grundlegenden Konzepte von Amazon Location beschrieben, darunter Abschnitte zu Karten, Ortssuche, Routen sowie Geofences und Trackern.
- [Amplify](#) — Amplify ist eine Komplettlösung, die einen Großteil der Funktionen umfasst, die für die Erstellung von Web- und Mobilanwendungen mithilfe von erforderlich sind. AWS Cloud Wenn Sie Amplify bereits verwenden oder sich für Amplify entscheiden, ist eine Geobibliothek mit Amazon Location Service integriert, die Sie verwenden können. Informationen zu den ersten Schritten mit Amplify Geo finden Sie in der Dokumentation [hier](#).

Szenarien und Anwendungsfälle

Amazon Location Service ist ein Service, der in der ausgeführt wird AWS Cloud. Sie können es von Ihren eigenen Amazon EC2 EC2-Instances in der Cloud aufrufen, aber viele Mapping-Anwendungen

laufen auf Geräten oder einer Kombination von Geräten und der Cloud. Im Folgenden sind nur einige typische Szenarien aufgeführt und wie Sie bei deren Entwicklung vorgehen könnten.

- Eine Backend-Anwendung, mit der Sie die Routen für Fahrer in Ihrer Flotte optimieren können.

Sie können eine Anwendung schreiben, die auf [Amazon EC2](#) läuft und den Amazon Location Service verwendet AWS Cloud , um [Routenmatrizen als Eingabe für einen Routenoptimierer für Ihre Flotte zu berechnen](#). Verwenden Sie das [AWS SDK](#), um Amazon Location aufzurufen.

- Eine Webanwendung, mit der Ihre Kunden die Standorte Ihres Unternehmens finden können.

Sie können eine Website erstellen, die auf Amazon EC2 EC2-Instances läuft, einschließlich einer standortbasierten Anwendung. Verwenden Sie das [AWS SDK JavaScript](#), um eine Webanwendung zu entwickeln, mit der Standorte mithilfe der [Ortssuche gesucht](#) und Ergebnisse auf einer [Karte](#) angezeigt werden können. MapLibre Verwenden Sie das Amazon Location SDK, um die Programmierung mit Standort zu vereinfachen.

- Fügen Sie Ortungsfunktionen zu einer vorhandenen iOS- oder Android-Anwendung hinzu.

Sie können das AWS SDK für Swift (iOS) oder [Kotlin](#) (Android) verwenden, um Amazon Location aufzurufen und Ihrer Anwendung Funktionen für die [Suche nach Orten](#) und [Karten](#) hinzuzufügen. Wird MapLibre zum Rendern von Karten verwendet. Für andere Sprachen sind zusätzliche [AWS SDKs](#) verfügbar.

- Verfolgen Sie Ressourcen (Geräte oder Fahrzeuge) und lassen Sie sich auf dem Laufenden halten, wenn sie Bereiche betreten oder verlassen, die Sie definieren.

Eine Anwendung zur Geräteverfolgung besteht aus mehreren Teilen.

- Für jedes Gerät, das Sie verfolgen, muss eine [Tracker-Ressource](#) erstellt werden, um es zu verfolgen. Es muss Positionsaktualisierungen beispielsweise mithilfe von [MQTT](#) an Amazon Location Service senden.
- Erstellen Sie [Geofences](#), um Bereiche zu definieren, in denen Sie Eingangs- und Austrittsereignisse für Ihre Anlagen erhalten möchten.
- Sie können [Amazon EC2](#) verwenden oder [AWS Lambda](#) um auf Ihre Ereignisse zu reagieren, wenn Ressourcen die Geofence-Bereiche betreten oder verlassen.
- Sie können dies erweitern, um Web- oder Geräteanwendungen zu erstellen, mit denen Sie Ihre Asset-Standorte verfolgen und auf Karten anzeigen können.

Der folgende Abschnitt enthält Einzelheiten zu den Tools und Bibliotheken, die für jeden Aspekt des Amazon Location Service zur Verfügung stehen.

SDKs und Tools für die Nutzung von Amazon Location Service

Es gibt verschiedene Tools, die Ihnen bei der Nutzung von Amazon Location Service helfen.

- **AWS SDKs** — Die AWS Software Development Kits (SDKs) sind in vielen gängigen Programmiersprachen verfügbar und bieten eine API, Codebeispiele und Dokumentation, die es einfacher machen, Anwendungen in Ihrer bevorzugten Sprache zu erstellen. Die AWS SDKs enthalten die wichtigsten APIs und Funktionen von Amazon Location, einschließlich Zugriff auf Maps, Places Search, Routes, Geofence und Tracker. Weitere Informationen zu den SDKs, die mit Amazon Location Service für verschiedene Anwendungen und Sprachen verwendet werden können, finden Sie unter [SDKs nach Sprache](#).
- **MapLibre**— Amazon Location Service empfiehlt, Karten mit der [MapLibre](#) Rendering-Engine zu rendern. MapLibre ist eine Engine zur Anzeige von Karten in Web- oder Mobilanwendungen. MapLibre hat auch ein Plugin-Modell und unterstützt die Benutzeroberfläche für Suchen und Routen in einigen Sprachen und Plattformen. Weitere Informationen zur Verwendung MapLibre und zu den Funktionen, die es bietet, finden Sie unter [MapLibre](#).
- **Amazon Location SDK** — Das Amazon Location SDK besteht aus einer Reihe von Open-Source-Bibliotheken, die die Entwicklung von Anwendungen mit Amazon Location Service erleichtern. Die Bibliotheken bieten Funktionen zur Unterstützung der Authentifizierung für Mobil- und Webanwendungen, der Standortverfolgung für mobile Anwendungen, der Konvertierung zwischen Amazon Location-Datentypen und [GeoJSON](#) sowie ein gehostetes Paket des Amazon Location-Clients für das AWS SDK v3. Weitere Informationen zum Amazon Location SDK finden Sie unter [Amazon-Standort-SDK](#).

SDKs nach Sprache

Die folgenden Tabellen enthalten Informationen zu AWS SDKs und MapLibre Versionen für Sprachen und Frameworks, sortiert nach Anwendungstyp: Web-, Mobil- oder Backend-Anwendung.

SDK-Versionen

Wir empfehlen, die neueste Version des AWS SDK und alle anderen SDKs, die Sie in Ihren Projekten verwenden, zu verwenden und die SDKs auf dem neuesten Stand zu halten. Das

AWS SDK bietet Ihnen die neuesten Features und Funktionen sowie Sicherheitsupdates. Die neueste Version des AWS SDK für finden Sie JavaScript beispielsweise in der JavaScript Dokumentation zum Thema [Browserinstallation](#) im AWS SDK.

Web frontend

Die folgenden AWS SDKs und MapLibre Versionen sind für die Entwicklung von Web-Frontend-Anwendungen verfügbar.

Sprache//Framework	AWS SDK	Framework zum Rendern
Vollständig unterstützt		
JavaScript	https://aws.amazon.com/sdk-for-javascript/	https://maplibre.org/projects/maplibre-gl-js/
ReactJS	https://aws.amazon.com/sdk-for-javascript	https://github.com/maplibre/maplibre-react-native
TypeScript	https://aws.amazon.com/sdk-for-javascript/	https://maplibre.org/projects/maplibre-gl-js/
Teilweise unterstützt		
Flutter	https://docs.amplify.aws/start/q/integration/flutter/ Flutter wird noch nicht vollständig von Amplify unterstützt AWS, aber über Amplify wird nur begrenzter Support angeboten.	https://github.com/maplibre/flutter-maplibre-gl Die MapLibre Flutter-Bibliothek gilt als experimentell.
Node.js	https://aws.amazon.com/sdk-for-javascript	Es gibt keine MapLibre Unterstützung für Node.js.
PHP	https://aws.amazon.com/sdk-for-php/	Es gibt keine MapLibre Unterstützung für PHP.

Mobile frontend

Die folgenden AWS SDKs und MapLibre Versionen sind für die Entwicklung mobiler Frontend-Anwendungen verfügbar.

Sprache//Framework	AWS SDK	Framework zum Rendern
Vollständig unterstützt		
Java	https://aws.amazon.com/sdk-for-java/	https://maplibre.org/projects/maplibre-native/
Kotlin	<p>https://aws.amazon.com/sdk-for-kotlin/</p> <p>Amazon Location Service SDK für mobile Authentifizierung für Android: https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-android</p> <p>Amazon Location Service SDK für mobiles Tracking für Android: https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-android</p>	<p>https://maplibre.org/projects/maplibre-native/</p> <p>Erfordert benutzerdefinierte Bindungen, da MapLibre es auf Java basiert.</p>
ObjectiveC	https://github.com/aws-amplify/aws-sdk-ios	https://maplibre.org/projects/maplibre-native/
ReactNative	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-react-native
Swift	<p>https://aws.amazon.com/sdk-for-swift/</p> <p>Amazon Location Service SDK für mobile Authentifizierung</p>	https://maplibre.org/projects/maplibre-native/

Sprache//Framework	AWS SDK	Framework zum Rendern
	<p>isierung für iOS: https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios</p> <p>Amazon Location Service SDK für mobiles Tracking für iOS: https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios</p>	
Teilweise unterstützt		
Flutter	<p>https://docs.amplify.aws/start/q/integration/flutter/</p> <p>Flutter wird noch nicht vollständig von Amplify unterstützt AWS, aber über Amplify wird nur begrenzter Support angeboten.</p>	<p>https://github.com/maplibre/flutter-maplibre-gl</p> <p>Die MapLibre Flutter-Bibliothek gilt als experimentell.</p>

Backend application

Die folgenden AWS SDKs sind für die Entwicklung von Backend-Anwendungen verfügbar. MapLibre ist hier nicht aufgeführt, da das Rendern von Karten für Backend-Anwendungen normalerweise nicht erforderlich ist.

Sprache	AWS SDK
.NET	https://aws.amazon.com/sdk-for-net
C++	https://aws.amazon.com/sdk-for-cpp/
Go	https://aws.amazon.com/sdk-for-go/
Java	https://aws.amazon.com/sdk-for-java/

Sprache	AWS SDK
JavaScript	https://aws.amazon.com/sdk-for-javascript/
Node.js	https://aws.amazon.com/sdk-for-javascript/
TypeScript	https://aws.amazon.com/sdk-for-javascript/
Kotlin	https://aws.amazon.com/sdk-for-kotlin/
PHP	https://aws.amazon.com/sdk-for-php/
Python	https://aws.amazon.com/sdk-for-python/
Ruby	https://aws.amazon.com/sdk-for-ruby/
Rust	https://aws.amazon.com/sdk-for-rust/ Das AWS SDK für Rust befindet sich in der Entwickler-Vorschauversion.

MapLibre Tools und Bibliotheken mit Amazon Location verwenden

Eines der wichtigsten Tools für die Erstellung interaktiver Anwendungen mit Amazon Location ist MapLibre. [MapLibre](#) ist in erster Linie eine Rendering-Engine zum Anzeigen von Karten in einer Web- oder Mobilanwendung. Es beinhaltet jedoch auch Unterstützung für Plug-ins und bietet Funktionen für die Arbeit mit anderen Aspekten von Amazon Location. Im Folgenden werden Tools beschrieben, die Sie je nach dem Standort, mit dem Sie arbeiten möchten, verwenden können.

Note

Um einen beliebigen Aspekt von Amazon Location zu verwenden, installieren Sie das [AWS SDK für die Sprache, die Sie verwenden möchten](#).

- Landkarten

Um Karten in Ihrer Anwendung anzuzeigen, benötigen Sie eine Karten-Rendering-Engine, die die von Amazon Location bereitgestellten Daten verwendet und auf den Bildschirm zeichnet. Karten-

Rendering-Engines bieten auch Funktionen zum Schwenken und Zoomen der Karte oder zum Hinzufügen von Markierungen, Stecknadeln und anderen Anmerkungen zur Karte.

Amazon Location Service empfiehlt, Karten mit der [MapLibre](#) Rendering-Engine zu rendern. MapLibre GL JS ist eine Engine für die Anzeige von Karten in JavaScript, während MapLibre Native Karten entweder für iOS oder Android bereitstellt.

MapLibre hat auch ein Plug-in-Ökosystem zur Erweiterung der Kernfunktionalität. Weitere Informationen finden Sie unter <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

- Suche nach Orten

Um die Erstellung einer Benutzeroberfläche für die Suche zu vereinfachen, können Sie den [MapLibre Geocoder](#) für das Web verwenden (Android-Anwendungen können das [Android Places-Plug-In](#) verwenden).

Verwenden Sie die [Geocoder-Bibliothek Amazon Location for Maplibre](#), um die Verwendung von Amazon Location in Applications zu vereinfachen. `amazon-location-for-maplibre-gl-geocoder` JavaScript

- Routen

Verwenden Sie [MapLibreWegbeschreibungen](#), um Routen auf der Karte anzuzeigen.

- Geofences und Tracker

MapLibre bietet keine speziellen Rendering-Funktionen oder Tools für Geofences und Tracking, aber Sie können die Rendering-Funktionen und [Plug-ins](#) verwenden, um die Geofences und getrackten Geräte auf der Karte anzuzeigen.

Die Geräte, die verfolgt werden, können [MQTT](#) verwenden oder Updates manuell an Amazon Location Service senden. Auf Geofence-Ereignisse kann mit reagiert werden. [AWS Lambda](#)

Viele Open-Source-Bibliotheken sind verfügbar, um zusätzliche Funktionen für Amazon Location Service bereitzustellen, zum Beispiel [Turf](#), das Funktionen zur räumlichen Analyse bereitstellt.

Viele Bibliotheken verwenden den offenen Standard [GeoJSON-formatierte Daten](#). Amazon Location Service bietet eine Bibliothek zur Unterstützung der Verwendung von GeoJSON in JavaScript Anwendungen. Weitere Informationen finden Sie im folgenden Abschnitt, [Amazon Location SDK und Bibliotheken](#).

Amazon Location MapLibre Geocoder-Plugin

Das Amazon Location MapLibre Geocoder-Plugin wurde entwickelt, um Ihnen die Integration von Amazon Location-Funktionen in Ihre JavaScript Anwendungen zu erleichtern, wenn Sie mit dem Rendern und Geokodieren von Karten mithilfe der Bibliothek arbeiten. [maplibre-gl-geocoder](#)

Installation

Sie können das Amazon Location MapLibre Geocoder-Plugin von NPM zur Verwendung mit Modulen mit dem folgenden Befehl installieren:

```
npm install @aws/amazon-location-for-maplibre-gl-geocoder
```

Sie können mit einem Skript in eine HTML-Datei importieren, um sie direkt im Browser zu verwenden:

```
<script src="https://www.unpkg.com/@aws/amazon-location-for-maplibre-gl-geocoder@1"/>/script<
```

Verwendung mit Modul

Dieser Code richtet eine Maplibre JavaScript GL-Karte mit Geokodierungsfunktionen von Amazon Location ein. Es verwendet die Authentifizierung über Amazon Cognito Identity Pool, um auf Amazon Location-Ressourcen zuzugreifen. Die Karte wird mit einem bestimmten Stil und bestimmten Mittelpunktkoordinaten gerendert und ermöglicht die Suche nach Orten auf der Karte.

```
// Import MapLibre GL JS
import maplibregl from "maplibre-gl";
// Import from the AWS JavaScript SDK V3
import { LocationClient } from "@aws-sdk/client-location";
// Import the utility functions
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";
// Import the AmazonLocationWithMaplibreGeocoder
import { buildAmazonLocationMaplibreGeocoder, AmazonLocationMaplibreGeocoder } from
  "@aws/amazon-location-for-maplibre-gl-geocoder"

const identityPoolId = "Identity Pool ID";
const mapName = "Map Name";
const region = "Region"; // region containing the Amazon Location resource
const placeIndex = "PlaceIndexName" // Name of your places resource in your AWS
Account.
```

```
// Create an authentication helper instance using credentials from Amazon Cognito
const authHelper = await withIdentityPoolId("Identity Pool ID");

const client = new LocationClient({
  region: "Region", // Region containing Amazon Location resources
  ...authHelper.getLocationClientConfig(), // Configures the client to use
  credentials obtained via Amazon Cognito
});

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor`,
  ...authHelper.getMapAuthenticationOptions(),
});

// Gets an instance of the AmazonLocationMaplibreGeocoder Object.
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
placeIndex, {enableAll: true});

// Now we can add the Geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoder.getPlacesGeocoder());
```

Verwendung mit einem Browser

In diesem Beispiel wird der Amazon Location Client verwendet, um eine Anfrage zu stellen, die sich mit Amazon Cognito authentifiziert.

Note

Einige dieser Beispiele verwenden den Amazon Location Client. Der Amazon Location Client basiert auf dem [AWS SDK für JavaScript V3](#) und ermöglicht das Aufrufen von Amazon Location über ein Skript, auf das in einer HTML-Datei verwiesen wird.

Fügen Sie Folgendes in eine HTML-Datei ein:

```
< Import the Amazon Location With Maplibre Geocoder >
```

```
<script src="https://www.unpkg.com/@aws/amazon-location-with-maplibre-geocoder@1"></script>
<Import the Amazon Location Client>
<script src="https://www.unpkg.com/@aws/amazon-location-client@1"></script>
<!Import the utility library>
<script src="https://www.unpkg.com/@aws/amazon-location-utilities-auth-helper@1"></script>
```

Nehmen Sie Folgendes in eine JavaScript Datei auf:

```
const identityPoolId = "Identity Pool ID";
const mapName = "Map Name";
const region = "Region"; // region containing Amazon Location resource

// Create an authentication helper instance using credentials from Amazon Cognito
const authHelper = await
  amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-descriptor`,
  ...authHelper.getMapAuthenticationOptions(),
});

// Initialize the AmazonLocationMaplibreGeocoder object
const amazonLocationMaplibreGeocoderObject =
  amazonLocationMaplibreGeocoder.buildAmazonLocationMaplibreGeocoder(client,
  placesName, {enableAll: true});

// Use the AmazonLocationWithMaplibreGeocoder object to add a geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoderObject.getPlacesGeocoder());
```

Im Folgenden sind die Funktionen und Befehle aufgeführt, die im Amazon Location MapLibre Geocoder-Plugin verwendet werden:

- **buildAmazonLocationMaplibreGeocoder**

Diese Klasse erstellt eine Instanz von `AmazonLocationMaplibreGeocoder`, die der Einstiegspunkt für alle anderen Aufrufe ist:

```
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
  placesIndex, {enableAll: true});
```

- **getPlacesGeocoder**

Gibt ein gebrauchsfertiges `IControl`-Objekt zurück, das direkt zu einer Map hinzugefügt werden kann.

```
const geocoder = getPlacesGeocoder();

// Initialize map
let map = await initializeMap();

// Add the geocoder to the map.
map.addControl(geocoder);
```

Amazon Location SDK und Bibliotheken

Das Amazon Location SDK besteht aus einer Reihe von Open-Source-Bibliotheken, die nützliche Funktionen für die Entwicklung von Amazon Location-Anwendungen bieten. Die folgenden Funktionen sind enthalten:

- **Amazon Location Client** — Die Amazon Location-Objekte im AWS SDK v3 sind gebündelt und verpackt, um die Verwendung bei der Webentwicklung zu vereinfachen.
- **Authentifizierung** — Das Authentifizierungsprogramm vereinfacht die Authentifizierung (mithilfe von Amazon Cognito oder API-Schlüsseln) beim Erstellen einer Webseite [JavaScript](#), [iOS](#)- oder [Android-Anwendung](#) für Amazon Location Service.
- **Tracking** — Die mobilen Tracking-SDKs sind für [iOS](#) und [Android](#) verfügbar. Dieses SDK erleichtert mobilen Anwendungen die Interaktion mit Amazon Location Trackers.
- **GeoJSON-Funktionen von Amazon Location** — Die [GeoJSON-Konvertierungsprogramme](#) erleichtern die Konvertierung zwischen den branchenüblichen [GeoJSON-formatierten](#) Daten und den Amazon Location API-Formaten.

Themen

- [Wie fange ich an, das Amazon Location SDK zu verwenden](#)
- [Kunde mit Amazon-Standort](#)
- [JavaScript Authentifizierungshelfer](#)
- [GeoJSON-Konvertierungshilfen](#)
- [SDK für die mobile Authentifizierung von Android](#)
- [SDK für die mobile Authentifizierung von iOS](#)
- [Tracking-SDK für Android für Mobilgeräte](#)
- [iOS SDK für mobiles Tracking](#)

Wie fange ich an, das Amazon Location SDK zu verwenden

Das Amazon Location SDK besteht aus einer Reihe von Funktionen, die die Verwendung des Amazon Location Service in einer Anwendung vereinfachen können. Sie können diese Funktionen installieren und in Ihre JavaScript Anwendung importieren. In den folgenden Abschnitten werden der Amazon Location-Client sowie die Authentifizierungs- und GeoJSON-Hilfsbibliotheken beschrieben.

Kunde mit Amazon-Standort

Bei AWS SDK v3 ist das SDK nach Diensten getrennt. Sie können nur die Teile installieren, die Sie benötigen. Verwenden Sie beispielsweise die folgenden Befehle, um den Amazon Location-Client und den Anmeldeinformationsanbieter für Amazon Cognito zu installieren.

```
npm install @aws-sdk/client-location
npm install @aws-sdk/credential-providers
```

Um die Nutzung von Amazon Location Service in JavaScript Web-Frontend-Anwendungen zu erleichtern, AWS stellt es ein gehostetes Paket aus der Amazon Location-Bibliothek und dem Anbieter für Anmeldeinformationen bereit. Um den mitgelieferten Client zu verwenden, fügen Sie ihn wie folgt zu Ihrem HTML-Code in einem Skript-Tag hinzu:

```
<script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/amazonLocationClient.js"></script>
```

Note

Das Paket wird auf dem neuesten Stand gehalten und ist aus Gründen der Benutzerfreundlichkeit abwärtskompatibel. Wenn Sie dieses Skript-Tag oder die NPM-Installation verwenden, wird immer die neueste Version abgerufen.

JavaScript Authentifizierungshelfer

Der Amazon JavaScript Location-Authentifizierungshelfer erleichtert die Authentifizierung, wenn Sie Amazon Location API-Aufrufe von Ihrer JavaScript Anwendung aus tätigen. Dieser Authentifizierungshelfer hilft Ihnen speziell bei der Verwendung von [Amazon Cognito](#) oder [API-Schlüsseln](#) als Authentifizierungsmethode. Dies ist eine Open-Source-Bibliothek, die hier verfügbar GitHub ist: <https://github.com/aws-geospatial/amazon-location-utilities-auth-helper-js>.

Note

Die Amazon Cognito-Unterstützung im Authentifizierungshelfer unterstützt die Funktion für föderierte Identitäten von Amazon Cognito nicht.

Installation

Sie können die Bibliotheken mit einer lokalen Installation verwenden, wenn Sie ein Build-System wie Webpack verwenden, oder indem Sie vorgefertigte JavaScript Bundles mit `<script>` Tags in Ihrer HTML aufnehmen.

- Verwenden Sie den folgenden Befehl, um die Bibliothek mit NPM zu installieren:

```
npm install @aws/amazon-location-utilities-auth-helper
```

- Verwenden Sie den folgenden Befehl in Ihrer HTML-Datei, um das Skript zu laden:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/dist/amazonLocationAuthHelper.js"></script>
```

Import

Um eine bestimmte Funktion in Ihrer JavaScript Anwendung zu verwenden, müssen Sie diese Funktion importieren. Der folgende Code wird verwendet, um die Funktion `withIdentityPoolId` in Ihre Anwendung zu importieren.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';
```

Authentifizierungsfunktionen

Die Amazon Location-Authentifizierungshelfer umfassen die folgenden Funktionen, die ein `AuthHelper` Objekt zurückgeben:

- `async withIdentityPoolId(identityPoolId: string): AuthHelper`— Diese Funktion gibt ein `AuthHelper` Objekt zurück, das für die Verwendung mit Amazon Cognito initialisiert wurde
- `async withAPIKey(API_KEY: string): AuthHelper`— Diese Funktion gibt ein `AuthHelper` Objekt zurück, das für die Verwendung mit API-Schlüsseln initialisiert wurde.

Das `AuthHelper` Objekt bietet die folgenden Funktionen:

- `AuthHelper.getMapAuthenticationOptions()`— Diese Funktion des `AuthHelper` Objekts gibt ein JavaScript Objekt mit dem `zurücktransformRequest`, das mit den Map-Optionen in MapLibre JS verwendet werden kann. Wird nur bereitgestellt, wenn es mit einem Identitätspool initialisiert wurde.
- `AuthHelper.getLocationClientConfig()`— Diese Funktion des `AuthHelper` Objekts gibt ein JavaScript Objekt mit dem `zurückcredentials`, das zur Initialisierung eines verwendet werden kann. `LocationClient`
- `AuthHelper.getCredentials()`— Diese Funktion des `AuthHelper` Objekts gibt die internen Anmeldeinformationen von Amazon Cognito zurück. Wird nur bereitgestellt, wenn es mit einem Identitätspool initialisiert wurde.

Beispiel: Initialisieren eines MapLibre Kartenobjekts mit Amazon Cognito unter Verwendung eines `AuthHelper`

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials
```



```
const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/mapName/style-
descriptor', // Defines the appearance of the map
  ...authHelper.getMapAuthenticationOptions(), // Provides credential options
  required for requests to Amazon Location
});
```

Beispiel: Initialisierung des MapLibre Kartenobjekts mit einem API-Schlüssel (**AuthHelper** ist in diesem Fall nicht erforderlich)

```
const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=api-key-id',
});
```

Beispiel: Initialisieren Sie den Location-Client über das AWS SDK für JS mithilfe von Amazon Cognito und AuthHelper

In diesem Beispiel wird AWS SDK für JavaScript v3 verwendet.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  Cognito pool defined above
});

//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
```

```
    Position: [-123.1187, 49.2819], // position to search near
    MaxResults: 10 // number of results to return
  });
```

Beispiel: Initialisieren Sie den Location-Client über das AWS SDK für JS mithilfe eines API-Schlüssels und AuthHelper

In diesem Beispiel wird AWS SDK für JavaScript v3 verwendet.

```
import { withAPIKey } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withAPIKey("api-key-id"); // use API Key id for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  API Key defined above
});

//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
  Position: [-123.1187, 49.2819], // position to search near
  MaxResults: 10 // number of results to return
}));
```

GeoJSON-Konvertierungshilfen

Die Amazon Location GeoJSON-Konvertierungshilfen bieten Tools zur Konvertierung von Amazon Location Service Service-Datentypen in und aus dem branchenüblichen [GeoJSON-Format](#). GeoJSON wird beispielsweise verwendet, um geografische Daten auf der Karte MapLibre zu rendern. Dies ist eine Open-Source-Bibliothek, die hier verfügbar GitHub ist: <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

Installation

Sie können die Bibliotheken mit einer lokalen Installation wie Webpack verwenden oder indem Sie vorgefertigte JavaScript Bundles mit `<script>` Tags in Ihren HTML-Code einfügen.

- Verwenden Sie den folgenden Befehl, um die Bibliothek mithilfe von NPM zu installieren.

```
npm install @aws/amazon-location-utilities-datatypes
```

- Verwenden Sie den folgenden Befehl in Ihrer HTML-Datei, um das Skript zu laden:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-datatypes@1.x/dist/amazonLocationDataConverter.js"></script>
```

Import

Um eine bestimmte Funktion in Ihrer JavaScript Anwendung zu verwenden, müssen Sie diese Funktion importieren. Der folgende Code wird verwendet, um die Funktion `placeToFeatureCollection` in Ihre Anwendung zu importieren.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utilities-datatypes';
```

GeoJSON-Konvertierungsfunktionen

Die Amazon Location GeoJSON-Konvertierungshilfen umfassen die folgenden Funktionen:

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse, keepNull: boolean): FeatureCollection`— Diese Funktion konvertiert Antworten aus den Ortssuchfunktionen in eine GeoJSON-Datei `FeatureCollection` mit einem oder mehreren `Point-Features`.
- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)`— Diese Funktion konvertiert Antworten aus den Positionsfunktionen des Tracker-Geräts in eine GeoJSON `FeatureCollection` mit einem oder mehreren `Punktmerkmalen`.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection`— Diese Funktion konvertiert Antworten aus der Funktion „Route berechnen“ in eine GeoJSON-Datei `FeatureCollection` mit einem einzigen `MultiStringLine Feature`. Jeder Abschnitt der Route wird durch einen `LineString` Eintrag in der `MultiStringLine` dargestellt.
- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection`— Diese Funktion konvertiert die Anfrage oder Antwort von Geofence-

Funktionen in eine GeoJSON FeatureCollection mit Polygon-Funktionen. Sie kann Geofences sowohl in der Antwort als auch in der Anfrage konvertieren, sodass Sie Geofences auf einer Karte anzeigen können, bevor Sie sie mit oder hochladen. `PutGeofence` `BatchPutGeofence`

Diese Funktion wandelt einen Kreisgeofence in ein Feature mit einem angenäherten Polygon um, verfügt aber auch über die Eigenschaften „Mittelpunkt“ und „Radius“, um bei Bedarf die Kreisgeofence nachzubilden (siehe die nächste Funktion).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]`— Diese Funktion konvertiert ein GeoJSON FeatureCollection mit Polygon-Features in ein Array von `BatchPutGeofenceRequestEntry` Objekten, sodass das Ergebnis verwendet werden kann, um eine Anfrage zu erstellen. `BatchPutGeofence`

Wenn ein Feature in der über die Eigenschaften „Mittelpunkt“ und „Radius“ FeatureCollection verfügt, wird es in einen Kreisgeofence-Anforderungseintrag umgewandelt, wobei die Geometrie des Polygons ignoriert wird.

Beispiel: Konvertiert Suchergebnisse in eine Punktebene in MapLibre

In diesem Beispiel wird AWS SDK für JavaScript Version 3 verwendet.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utility-datatypes';

...

let map; // map here is an initialized MapLibre instance

const client = new LocationClient(config);
const input = { your_input };
const command = new searchPlaceIndexForTextCommand(input);
const response = await client.send(command);

// calling utility function to convert the response to GeoJSON
const featureCollection = placeToFeatureCollection(response);
map.addSource("search-result", featureCollection);
map.addLayer({
  id: "search-result",
  type: "circle",
  source: "search-result",
  paint: {
```

```
        "circle-radius": 6,  
        "circle-color": "#B42222",  
    },  
});
```

SDK für die mobile Authentifizierung von Android

Diese Dienstprogramme helfen Ihnen bei der Authentifizierung, wenn Sie Amazon Location Service API-Aufrufe von Ihren Android-Anwendungen aus tätigen. Dies hilft insbesondere bei der Verwendung von [Amazon Cognito](#) oder [API-Schlüsseln](#) als Authentifizierungsmethode.

Das Android Mobile Authentication SDK ist auf Github verfügbar: [Amazon Location Service Mobile Authentication SDK for Android](#). Darüber hinaus sind sowohl das SDK für die mobile Authentifizierung als auch das AWS SDK im [AWS Maven-Repository](#) verfügbar.

Installation

Um das SDK für die mobile Authentifizierung zu verwenden, fügen Sie Ihrer `build.gradle` Datei in Android Studio die folgenden Importanweisungen hinzu.

```
implementation("software.amazon.location:auth:0.0.1")  
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

Authentifizierungsfunktionen

Das Authentication Helper SDK hat die folgenden Funktionen:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Diese Funktion gibt einen API-Schlüssel zurück, der `LocationCredentialsProvider` initialisiert wurde, um mit ihm zu arbeiten.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Diese Funktion gibt einen Identitätspool zurück, der für die Arbeit mit einem Amazon Cognito Cognito-Identitätspool `LocationCredentialsProvider` initialisiert wurde.

Verwendung

Um das SDK in Ihrem Code zu verwenden, importieren Sie die folgenden Klassen:

```
import com.amazonaws.services.geo.AmazonLocationClient
```

```
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

Beim Erstellen der Instanzen Authentication Helper und Location Client Provider stehen Ihnen zwei Optionen zur Verfügung. Sie können eine Instance mithilfe von [Amazon Location API-Schlüsseln](#) oder [Amazon Cognito](#) erstellen.

- Um eine Authentifizierungshelfer-Instance mit einem Amazon Location API-Schlüssel zu erstellen, deklarieren Sie die Helper-Klasse wie folgt:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")
```

- Um eine Authentifizierungshelferinstanz mit Amazon Cognito zu erstellen, deklarieren Sie die Helper-Klasse wie folgt:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

Sie können mithilfe des Anbieters für Standortinformationen eine Amazon Location-Client-Instance erstellen und den Amazon Location-Service aufrufen. Im folgenden Beispiel wird nach Orten in der Nähe eines bestimmten Breiten- und Längengrades gesucht.

```
var locationClient =
    authHelper.getLocationClient(locationCredentialsProvider.getCredentialsProvider())
var searchPlaceIndexForPositionRequest =
    SearchPlaceIndexForPositionRequest().withIndexName("My-Place-Index-
    Name").withPosition(arrayListOf(30.405423, -97.718833))
var nearbyPlaces =
    locationClient.searchPlaceIndexForPosition(searchPlaceIndexForPositionRequest)
```

SDK für die mobile Authentifizierung von iOS

Diese Dienstprogramme helfen Ihnen bei der Authentifizierung, wenn Sie Amazon Location Service API-Aufrufe von Ihren iOS-Anwendungen aus tätigen. Dies hilft insbesondere bei der Verwendung von [Amazon Cognito](#) oder [API-Schlüsseln](#) als Authentifizierungsmethode.

Das iOS Mobile Authentication SDK ist auf Github verfügbar: [Amazon Location Service Mobile Authentication SDK for iOS](https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios).

Installation

Installieren Sie das SDK in einem Xcode-Projekt:

1. Gehen Sie zu Datei und wählen Sie dann Paketabhängigkeiten hinzufügen in Ihrem XCode-Projekt aus.
2. Geben Sie die Paket-URL: [https://github.com/aws-geospatial/ amazon-location-mobile-auth -sdk-ios/](https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/) in die Suchleiste ein und drücken Sie die Eingabetaste.
3. Wählen Sie das amazon-location-mobile-auth-sdk-ios Package aus und klicken Sie auf Paket hinzufügen.
4. Wählen Sie das AmazonLocationiOSAuthSDK Paketprodukt aus und klicken Sie auf Package hinzufügen.

Authentifizierungsfunktionen

Das Authentication Helper SDK hat die folgenden Funktionen:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: Diese Funktion gibt einen API-Schlüssel zurück, der `LocationCredentialsProvider` initialisiert wurde, um mit ihm zu arbeiten.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Diese Funktion gibt einen Identitätspool zurück, der für die Arbeit mit einem Amazon Cognito Cognito-Identitätspool `LocationCredentialsProvider` initialisiert wurde.

Verwendung

Um das SDK für die mobile Authentifizierung zu verwenden, fügen Sie Ihrer Aktivität die folgenden Anweisungen hinzu:

```
import AmazonLocationiOSAuthSDK
import AWSLocationXCF
```

Sie haben zwei Optionen, wenn Sie die Instanzen Authentication Helper und Location Client Provider erstellen. Sie können eine Instance mithilfe von [Amazon Location API-Schlüsseln](#) oder [Amazon Cognito](#) erstellen.

- Um eine Authentifizierungshelfer-Instance mit einem Amazon Location API-Schlüssel zu erstellen, deklarieren Sie die Helper-Klasse wie folgt:

```
let authHelper = AuthHelper()
let locationCredentialsProvider = authHelper.authenticateWithAPIKey(apiKey: "My-
Amazon-Location-API-Key", region: "account-region")
```

- Um eine Authentifizierungshelferinstanz mit Amazon Cognito zu erstellen, deklarieren Sie die Helper-Klasse wie folgt:

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
  authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Amazon-Location-API-
Key", region: "account-region")
```

Sie können mithilfe des Anbieters für Standortinformationen eine Amazon Location-Client-Instance erstellen und den Amazon Location-Service aufrufen. Im folgenden Beispiel wird nach Orten in der Nähe eines bestimmten Breiten- und Längengrades gesucht.

```
let locationClient = AWSLocation.default()
let searchPlaceIndexForPositionRequest =
  AWSLocationSearchPlaceIndexForPositionRequest()!
searchPlaceIndexForPositionRequest.indexName = "My-Place-Index-Name"
searchPlaceIndexForPositionRequest.position = [30.405423, -97.718833]
let nearbyPlaces = locationClient.searchPlaceIndex(forPosition:
  searchPlaceIndexForPositionRequest)
```

Tracking-SDK für Android für Mobilgeräte

Das Amazon Location Mobile Tracking SDK bietet Dienstprogramme, mit denen Sie sich auf einfache Weise authentifizieren, Gerätepositionen erfassen und Positionsaktualisierungen an Amazon Location Trackers senden können. Das SDK unterstützt das lokale Filtern von Standortaktualisierungen mit konfigurierbaren Aktualisierungsintervallen. Dies reduziert die Datenkosten und optimiert die intermittierende Konnektivität für Ihre Android-Anwendungen.

Das Android Tracking SDK ist verfügbar auf GitHub: [Amazon Location Mobile Tracking SDK for Android](#). Darüber hinaus sind sowohl das SDK für die mobile Authentifizierung als auch das AWS SDK im [AWS Maven-Repository](#) verfügbar. Das Android-Tracking-SDK ist so konzipiert, dass es mit dem allgemeinen AWS SDK funktioniert.

In diesem Abschnitt werden die folgenden Themen für das Amazon Location Mobile Tracking Android SDK behandelt:

Themen

- [Installation](#)
- [Verwendung](#)
- [Filter](#)
- [Tracking-Funktionen des Android Mobile SDK](#)
- [Beispiele](#)

Installation

Um das SDK zu installieren, fügen Sie dem Abschnitt dependencies Ihrer build.gradle-Datei in Android Studio die folgenden Zeilen hinzu:

```
implementation("software.amazon.location:tracking:0.0.1")
implementation("software.amazon.location:auth:0.0.1")
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

Verwendung

Dieses Verfahren zeigt Ihnen, wie Sie das SDK verwenden, um das Objekt zu authentifizieren und zu erstellen: `LocationTracker`

Note

Bei diesem Verfahren wird davon ausgegangen, dass Sie die im [Installation](#) Abschnitt erwähnte Bibliothek importiert haben.

1. Importieren Sie die folgenden Klassen in Ihren Code:

```
import software.amazon.location.tracking.LocationTracker
```

```
import software.amazon.location.tracking.config.LocationTrackerConfig
import software.amazon.location.tracking.util.TrackingSdkLogLevel
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

- Erstellen Sie als Nächstes eine `AuthHelper`, da der `LocationCredentialsProvider` Parameter für die Erstellung eines `LocationTracker` Objekts erforderlich ist:

```
// Create an authentication helper using credentials from Cognito
val authHelper = AuthHelper(applicationContext)
val locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

- Verwenden Sie nun das `LocationCredentialsProvider` und, `LocationTrackerConfig` um ein `LocationTracker` Objekt zu erstellen:

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

Filter

Das Android SDK für mobiles Tracking von Amazon Location verfügt über drei integrierte Standortfilter.

- TimeLocationFilter:** Filtert den aktuellen Standort, der hochgeladen werden soll, auf der Grundlage eines definierten Zeitintervalls.
- DistanceLocationFilter:** Filtert Standortaktualisierungen auf der Grundlage eines angegebenen Entfernungsschwellenwerts.

- `AccuracyLocationFilter`: Filtert Standortaktualisierungen, indem die seit der letzten Aktualisierung zurückgelegte Entfernung mit der Genauigkeit des aktuellen Standorts verglichen wird.

In diesem Beispiel werden Filter zum `LocationTracker` Zeitpunkt der Erstellung hinzugefügt:

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),
    AccuracyLocationFilter())
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

In diesem Beispiel wird der Filter zur Laufzeit aktiviert und deaktiviert mit `LocationTracker`:

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

Tracking-Funktionen des Android Mobile SDK

Das Amazon Location Mobile Tracking SDK for Android umfasst die folgenden Funktionen:

- Klasse: `LocationTracker`

```
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, trackerName: String), oder
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, clientConfig: LocationTrackerConfig)
```

Dies ist eine Initialisierungsfunktion zum Erstellen eines `LocationTracker` Objekts. Sie benötigt Instanzen von `LocationCredentialsProvider` `trackerName` und optional eine Instanz von `LocationTrackingConfig`. Wenn die Konfiguration nicht bereitgestellt wird, wird sie mit Standardwerten initialisiert.

- Klasse: `LocationTracker`

```
start(locationTrackingCallback: LocationTrackingCallback)
```

Startet den Prozess, auf den Standort des Benutzers zuzugreifen und ihn an einen Amazon Location Tracker zu senden.

- Klasse: `LocationTracker`

```
isTrackingInForeground()
```

Überprüft, ob die Standortverfolgung gerade läuft.

- Klasse: `LocationTracker`

```
stop()
```

Stoppt das Nachverfolgen des Standorts des Benutzers.

- Klasse: `LocationTracker`

```
startTracking()
```

Startet den Prozess, auf den Standort des Benutzers zuzugreifen und ihn an den AWS Tracker zu senden.

- Klasse: `LocationTracker`

```
startBackground(mode: BackgroundTrackingMode, serviceCallback:  
ServiceCallback)
```

Startet den Prozess, bei dem auf den Standort des Benutzers zugegriffen und dieser an den AWS Tracker gesendet wird, während sich die Anwendung im Hintergrund befindet. `BackgroundTrackingMode` hat die folgenden Optionen:

- `ACTIVE_TRACKING`: Diese Option verfolgt aktiv die Standortaktualisierungen eines Benutzers.
- `BATTERY_SAVER_TRACKING`: Mit dieser Option werden die Standortaktualisierungen des Benutzers alle 15 Minuten nachverfolgt.

- Klasse: `LocationTracker`

```
stopBackgroundService()
```

Stoppt den Zugriff auf den Standort des Benutzers und das Senden an den AWS Tracker, während sich die Anwendung im Hintergrund befindet.

- Klasse: `LocationTracker`

```
getTrackerDeviceLocation()
```

Ruft den Gerätestandort von den Amazon-Ortungsdiensten ab.

- Klasse: `LocationTracker`

```
getDeviceLocation(locationTrackingCallback: LocationTrackingCallback?)
```

Ruft den aktuellen Gerätestandort vom Fused Location Provider-Client ab und lädt ihn auf Amazon Location Tracker hoch.

- Klasse: `LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback: LocationTrackingCallback?)
```

Lädt den Gerätestandort nach der Filterung anhand der konfigurierten Standortfilter in die Amazon Location Services hoch.

- Klasse: `LocationTracker`

```
enableFilter(filter: LocationFilter)
```

Aktiviert einen bestimmten Standortfilter.

- Klasse: `LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```

Deaktivieren Sie einen bestimmten Standortfilter.

- Klasse: `LocationTrackerConfig`

```
LocationTrackerConfig( // Required var trackerName: String, // Optional var locationFilters: MutableList =
```

```
mutableListOf( TimeLocationFilter(), DistanceLocationFilter(), ), var
```

```
logLevel: TrackingSdkLogLevel = TrackingSdkLogLevel.DEBUG, var accuracy:
Int = Priority.PRIORITY_HIGH_ACCURACY, var latency: Long = 1000, var
frequency: Long = 1500, var waitForAccurateLocation: Boolean = false, var
minUpdateIntervalMillis: Long = 1000, var persistentNotificationConfig:
NotificationConfig = NotificationConfig())
```

Dadurch wird der `LocationTrackerConfig` mit benutzerdefinierten Parameterwerten initialisiert. Wenn kein Parameterwert angegeben wird, wird er auf einen Standardwert gesetzt.

- Klasse: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation:
LocationEntry?): Boolean
```

Das `LocationFilter` ist ein Protokoll, das Benutzer für ihre benutzerdefinierte Filter-Implementierung implementieren können. Sie müssen die `shouldUpload` Funktion implementieren, um den vorherigen und den aktuellen Standort zu vergleichen und herauszufinden, ob der aktuelle Standort hochgeladen werden soll.

Beispiele

Das folgende Codebeispiel zeigt die Funktionalität des Mobile Tracking SDK.

In diesem Beispiel wird das `usesLocationTracker`, um das Tracking im Hintergrund zu starten und zu beenden:

```
// For starting the location tracking
locationTracker?.startBackground(
BackgroundTrackingMode.ACTIVE_TRACKING,
object : ServiceCallback {
    override fun serviceStopped() {
        if (selectedTrackingMode == BackgroundTrackingMode.ACTIVE_TRACKING) {
            isLocationTrackingBackgroundActive = false
        } else {
            isLocationTrackingBatteryOptimizeActive = false
        }
    }
},
)
```

```
// For stopping the location tracking
```

```
locationTracker?.stopBackgroundService()
```

iOS SDK für mobiles Tracking

Das Amazon Location Mobile Tracking SDK bietet Dienstprogramme, mit denen Sie sich auf einfache Weise authentifizieren, Gerätepositionen erfassen und Positionsaktualisierungen an Amazon Location Trackers senden können. Das SDK unterstützt das lokale Filtern von Standortaktualisierungen mit konfigurierbaren Aktualisierungsintervallen. Dies reduziert die Datenkosten und optimiert die intermittierende Konnektivität für Ihre iOS-Anwendungen.

Das iOS-Tracking-SDK ist verfügbar auf GitHub: [Amazon Location Mobile Tracking SDK für iOS](#).

In diesem Abschnitt werden die folgenden Themen für das Amazon Location Mobile Tracking iOS SDK behandelt:

Themen

- [Installation](#)
- [Verwendung](#)
- [Filter](#)
- [Tracking-Funktionen des iOS Mobile SDK](#)
- [Beispiele](#)

Installation

Gehen Sie wie folgt vor, um das Mobile Tracking SDK for iOS zu installieren:

1. Gehen Sie in Ihrem Xcode-Projekt zu Datei und wählen Sie Paketabhängigkeiten hinzufügen.
2. Geben Sie die folgende URL: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/> in die Suchleiste ein und drücken Sie die Eingabetaste.
3. Wählen Sie das amazon-location-mobile-tracking-sdk-ios Package aus und klicken Sie auf Paket hinzufügen.
4. Wählen Sie das AmazonLocationiOSTrackingSDK Paketprodukt aus und klicken Sie auf Package hinzufügen.

Verwendung

Das folgende Verfahren zeigt Ihnen, wie Sie mithilfe von Anmeldeinformationen von Cognito einen Authentifizierungshelfer erstellen.

1. Nach der Installation der Bibliothek müssen Sie Ihrer `info.plist` Datei eine oder beide Beschreibungen hinzufügen:

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

2. Als Nächstes importiere das `AuthHelper` in deine Klasse:

```
import AmazonLocationiOSAuthSDKimport AmazonLocationiOSTrackingSDK
```

3. Anschließend erstellen Sie ein `AuthHelper` Objekt und verwenden es mit dem AWS SDK, indem Sie einen Authentifizierungshelfer mit Anmeldeinformationen von Amazon Cognito erstellen.

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
  authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Cognito-Identity-
  Pool-Id", region: "My-region") //example: us-east-1
let locationTracker = LocationTracker(provider: locationCredentialsProvider,
  trackerName: "My-tracker-name")

// Optionally you can set ClientConfig with your own values in either initialize or
  in a separate function
// let trackerConfig = LocationTrackerConfig(locationFilters:
  [TimeLocationFilter(), DistanceLocationFilter()],
  trackingDistanceInterval: 30,
  trackingTimeInterval: 30,
  logLevel: .debug)

// locationTracker = LocationTracker(provider: credentialsProvider, trackerName:
  "My-tracker-name",config: trackerConfig)
// locationTracker.setConfig(config: trackerConfig)
```


Filter

Das iOS-SDK für mobiles Tracking von Amazon Location verfügt über drei integrierte Standortfilter.

- **TimeLocationFilter**: Filtert den aktuellen Standort, der hochgeladen werden soll, auf der Grundlage eines definierten Zeitintervalls.
- **DistanceLocationFilter**: Filtert Standortaktualisierungen auf der Grundlage eines angegebenen Entfernungsschwellenwerts.
- **AccuracyLocationFilter**: Filtert Standortaktualisierungen, indem die seit der letzten Aktualisierung zurückgelegte Entfernung mit der Genauigkeit des aktuellen Standorts verglichen wird.

In diesem Beispiel werden Filter zum `LocationTracker` Zeitpunkt der Erstellung hinzugefügt:

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),
    AccuracyLocationFilter())
)

locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

In diesem Beispiel wird der Filter zur Laufzeit aktiviert und deaktiviert mit `LocationTracker`:

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

Tracking-Funktionen des iOS Mobile SDK

Das Amazon Location Mobile Tracking SDK for iOS umfasst die folgenden Funktionen:

- Klasse: `LocationTracker`

```
init(provider: LocationCredentialsProvider, trackerName: String, config: LocationTrackerConfig? = nil)
```

Dies ist eine Initialisierungsfunktion zum Erstellen eines `LocationTracker` Objekts. Sie benötigt Instanzen von `LocationCredentialsProvider` `trackerName` und optional eine Instanz von `LocationTrackingConfig`. Wenn die Konfiguration nicht bereitgestellt wird, wird sie mit Standardwerten initialisiert.

- Klasse: `LocationTracker`

```
setTrackerConfig(config: LocationTrackerConfig)
```

Dadurch wird die Konfiguration von Tracker so eingestellt, dass sie zu jedem Zeitpunkt nach der Initialisierung des Location Trackers wirksam wird.

- Klasse: `LocationTracker`

```
getTrackerConfig()
```

Dadurch wird die Konfiguration der Standortverfolgung abgerufen, die Sie in Ihrer App verwenden oder ändern können.

Gibt zurück: `LocationTrackerConfig`

- Klasse: `LocationTracker`

```
getDeviceId()
```

Ruft die generierte Geräte-ID des Location Trackers ab.

Gibt zurück: `String?`

- Klasse: `LocationTracker`

```
startTracking()
```

Startet den Prozess, auf den Standort des Benutzers zuzugreifen und ihn an den AWS Tracker zu senden.

- Klasse: `LocationTracker`

```
resumeTracking()
```

Setzt den Vorgang fort, auf den Standort des Benutzers zuzugreifen und ihn an den AWS Tracker zu senden.

- Klasse: `LocationTracker`

```
stopTracking()
```

Stoppt das Nachverfolgen des Standorts des Benutzers.

- Klasse: `LocationTracker`

```
startBackgroundTracking(mode: BackgroundTrackingMode)
```

Startet den Prozess, bei dem auf den Standort des Benutzers zugegriffen und dieser an den AWS Tracker gesendet wird, während sich die Anwendung im Hintergrund befindet. `BackgroundTrackingMode` hat die folgenden Optionen:

- `Active`: Diese Option unterbricht Standortaktualisierungen nicht automatisch.
- `BatterySaving`: Diese Option unterbricht automatisch Standortaktualisierungen
- `None`: Mit dieser Option werden Standortaktualisierungen im Hintergrund insgesamt deaktiviert

- Klasse: `LocationTracker`

```
resumeBackgroundTracking(mode: BackgroundTrackingMode)
```

Setzt den Vorgang fort, auf den Standort des Benutzers zuzugreifen und ihn an den AWS Tracker zu senden, während sich die Anwendung im Hintergrund befindet.

- Klasse: `LocationTracker`

```
stopBackgroundTracking()
```

Stoppt den Zugriff auf den Standort des Benutzers und das Senden an den AWS Tracker, während sich die Anwendung im Hintergrund befindet.

- Klasse: `LocationTracker`

```
getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil,  
endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse,
```

```
Error>)
```

Ruft die hochgeladenen Tracking-Standorte für das Gerät des Benutzers zwischen Start- und Enddatum und Uhrzeit ab.

Gibt zurück: Void

- Klasse: `LocationTrackerConfig`

```
init()
```

Dies initialisiert die `LocationTrackerConfig` mit Standardwerten.

- Klasse: `LocationTrackerConfig`

```
init(locationFilters: [LocationFilter]? = nil, trackingDistanceInterval: Double? = nil, trackingTimeInterval: Double? = nil, trackingAccuracyLevel: Double? = nil, uploadFrequency: Double? = nil, desiredAccuracy: CLLocationAccuracy? = nil, activityType: CLActivityType? = nil, logLevel: LogLevel? = nil)
```

Dadurch wird der `LocationTrackerConfig` mit benutzerdefinierten Parameterwerten initialisiert. Wenn kein Parameterwert angegeben wird, wird er auf einen Standardwert gesetzt.

- Klasse: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntity, previousLocation: LocationEntity?, trackerConfig: LocationTrackerConfig)
```

Das `LocationFilter` ist ein Protokoll, das Benutzer für ihre benutzerdefinierte Filter-Implementierung implementieren können. Ein Benutzer müsste eine `shouldUpload` Funktion implementieren, um den vorherigen und den aktuellen Standort zu vergleichen und zurückzukehren, wenn der aktuelle Standort hochgeladen werden soll.

Beispiele

In diesem Abschnitt werden Beispiele für die Verwendung des Amazon Location Mobile Tracking SDK for iOS beschrieben.

Note

Stellen Sie sicher, dass die erforderlichen Berechtigungen in der `info.plist` Datei festgelegt sind. Dies sind dieselben Berechtigungen, die im [Verwendung](#) Abschnitt aufgeführt sind.

Das folgende Beispiel zeigt die Funktionen zum Verfolgen des Gerätestandorts und zum Abrufen nachverfolgter Standorte:

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

Fangen Sie an, den Standort zu verfolgen:

```
do {
    try locationTracker.startTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app
    settings
    }
}
```

Setze die Verfolgung des Standorts fort:

```
do {
    try locationTracker.resumeTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
}
```

Beenden Sie die Verfolgung des Standorts:

```
locationTracker.stopTracking()
```

Hintergrundverfolgung starten:

```
do {
```

```

locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None
}
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
}

```

Hintergrundverfolgung fortsetzen:

```

do {
    locationTracker.resumeBackgroundTracking(mode: .Active)
}
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
}

```

Um die Hintergrundverfolgung zu beenden:

```
locationTracker.stopBackgroundTracking()
```

Rufen Sie die verfolgten Standorte des Geräts vom Tracker ab:

```

func getTrackingPoints(nextToken: String? = nil) {
    let startTime: Date = Date().addingTimeInterval(-86400) // Yesterday's day date and
    time
    let endTime: Date = Date()
    locationTracker.getTrackerDeviceLocation(nextToken: nextToken, startTime: startTime,
    endTime: endTime, completion: { [weak self] result in
        switch result {
            case .success(let response):

                let positions = response.devicePositions
                // You can draw positions on map or use it further as per your requirement

                // If nextToken is available, recursively call to get more data
                if let nextToken = response.nextToken {
                    self?.getTrackingPoints(nextToken: nextToken)
                }
            case .failure(let error):
                print(error)
            }
        }
    })
}

```

}

Amazon-Standort-APIs

Amazon Location Service bietet API-Operationen für den programmgesteuerten Zugriff auf die Standortfunktion. Dazu gehören APIs für Karten, Orte, Routen, Tracker, Geofences und das Taggen Ihrer Ressourcen. Informationen zu den verfügbaren API-Aktionen finden Sie in der [Amazon Location Service API-Referenz](#).

Beispiele finden Sie im [Codebeispiele](#) Kapitel dieses Handbuchs.

Amazon Location mit einem AWS SDK verwenden

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, AWS Anwendungen in ihrer bevorzugten Sprache zu erstellen.

Weitere Informationen zu den SDKs, die für die Verwendung mit Amazon Location Service nach Sprachen verfügbar sind, finden Sie [SDKs nach Sprache](#) in diesem Handbuch.

SDK-Versionen

Wir empfehlen, die neueste Version des AWS SDK und alle anderen SDKs, die Sie in Ihren Projekten verwenden, zu verwenden und die SDKs auf dem neuesten Stand zu halten. Das AWS SDK bietet Ihnen die neuesten Features und Funktionen sowie Sicherheitsupdates. Die neueste Version des AWS SDK für finden Sie JavaScript beispielsweise in der JavaScript Dokumentation zum Thema [Browserinstallation](#) im AWS SDK.

Aktualisierungen der Amazon Location API-Fehlermeldungen

Ab dem 1. August 2023 ändert das Amazon Location-Team API-Fehlermeldungen wie in den folgenden Tabellen beschrieben. Fehlercodes werden nicht geändert. Wenn Ihre Anwendungen von exakten Fehlermeldungszeichenfolgen abhängen, müssen Sie Ihre Anwendungen mit den neuen Zeichenfolgen aktualisieren. Wenn Sie Hilfe bei Fragen oder Problemen benötigen, wenden Sie sich an AWS Support.

Themen

- [Orte](#)

- [Zuordnungen](#)
- [Tracker](#)
- [Routen](#)
- [Metadaten](#)
- [Geofences](#)

Orte

Orte

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	resource <PlaceIndexName> not found, reason: <Reason> Resource '<PlaceIndexName>' not found placeIdx<PlaceIndexName> not found, reason: <Reason> no place index with name '%s' found	Place index not found: <PlaceIndexName>.
404	ResourceNotFoundException	place not found	Place not found: <PlaceId>.
400	ValidationException	PlaceIndex <PlaceIndexName> cannot be used for SearchPlaceIndexForSuggestions because it	A place index with 'IntendedUse' set to Storage does not support 'SearchPlaceIndexForSuggestions' operation.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		has IntendedUse <IntendedUse>	
400	ValidationException	only one of 'BiasPosition' or 'FilterBBox' may be set	Only one of 'BiasPosition' or 'FilterBBox' may be set.
400	ValidationException	BiasPosition must have exactly 2 entries	'BiasPosition' must have exactly 2 entries.
400	ValidationException	BiasPosition[0] must be between -180 and 180	'BiasPosition[0]' must be between -180 and 180.
400	ValidationException	BiasPosition[1] must be between -90 and 90	'BiasPosition[1]' must be between -90 and 90.
400	ValidationException	FilterBBox must have exactly 4 entries	'FilterBBox' must have exactly 4 entries.
400	ValidationException	FilterBBox[0] must be between -180 and 180	'FilterBBox[0]' must be between -180 and 180.
400	ValidationException	FilterBBox[1] must be between -90 and 90	'FilterBBox[1]' must be between -90 and 90.
400	ValidationException	FilterBBox[2] must be between -180 and 180	'FilterBBox[2]' must be between -180 and 180.
400	ValidationException	FilterBBox[3] must be between -90 and 90	'FilterBBox[3]' must be between -90 and 90.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	FilterBBox must have more southwesterly point before more northeasterly point	'FilterBBox' must have more southwesterly position before more northeasterly position.
400	ValidationException	Position must have exactly 2 entries	'Position' must have exactly 2 entries.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	Language is not a valid BCP 47 language tag	'Language' must comply with the BCP 47 Language Tag standard, but was set to <GivenValue>. For more information, see https://wikipedia.org/wiki/IETF_language_tag .
400	ValidationException	'placeID' is invalid	'Placeld' must be a valid ID.
400	ValidationException	no customer account ID parameter found	'RequesterAccountID' is a required field.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' Grab must only be used in following regions: ap-southeast-1.
400	ValidationException	'IntendedUse' and 'PricingPlan' must both be provided to update either property	'IntendedUse' and 'PricingPlan' must both be provided to update either attribute
402	ServiceQuotaExceededException	Place resources per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Place index resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Place index already exists: <PlaceIndexName>.

Zuordnungen

Zuordnungen

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception unable to find style template Error fetching style was not able to serialize the map style file	Internal server error. Try again later.
404	ResourceNotFoundException	Map not found	Map not found: <MapName>.
404	ResourceNotFoundException	Sprites are not supported for this resource	Sprite not found: <SpriteName>.
400	ValidationException	Resource name should be set	'MapName' is a required field.
400	ValidationException	Must provide a valid number for start and end of Range	Font Unicode range start and end numbers must both be provided.
400	ValidationException	Start of range is an invalid number: <StartValue>	Start of font Unicode range must be a valid number.
400	ValidationException	End of range is an invalid number: <StartValue>	End of font Unicode range must be a valid number.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	End of range must be exactly 255 higher from start of range, difference found: <Difference>	The difference between the start and end of the font Unicode range must be exactly 255. Difference found: <Difference>.
400	ValidationException	Start of range must be a multiple of 256, found <StartValue>	Start of font Unicode range must be a multiple of 256, but was set to: <StartValue>.
400	ValidationException	Request font is empty	'FontStack' is a required field.
400	ValidationException	Request font is not valid for the datasource <DataSource>	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Request font is not valid	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .
400	ValidationException	DataSource is invalid: <DataSource>	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Request filename is empty	'FileName' is a required field.
400	ValidationException	Request filename is not valid	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Filename is invalid: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
400	ValidationException	Filename is an invalid content type: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
400	ValidationException	Filename is invalid: <FileName>	'Filename' must not be empty.
400	ValidationException	y-coordinate part of 'Y' must be a valid integer	y- coordinate part of 'Y' must be an integer.
400	ValidationException	tile resolution part of 'Y' must be a valid integer followed by 'x'	Tile resolution part of 'Y' must be an integer followed by 'X'.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	file type extension part of 'Y' must not be empty if a '.' is present	File type extension part of 'Y' must not be empty if a '.' is present.
400	ValidationException	'Z' must be a valid integer	'Z' must be an integer.
400	ValidationException	'X' must be a valid integer	'X' must be an integer.
400	ValidationException	'Z' must not be less than minimum zoom of style '<Style>' (<Minimum Value>)	'Z' must not be less than minimum zoom of style <Style> (<MinimumValue>).
400	ValidationException	'Z' must not be greater than maximum zoom of style '<Style>' (<Maximum Value>)	'Z' must not be greater than maximum zoom of style Style (<MaximumValue>).
400	ValidationException	'Z' value not supported	'Z' must be between 0 and 63.
400	ValidationException	tile resolution part of 'Y' must be omitted because '<Style>' is a vector style	Tile resolution part of 'Y' must be omitted for style <Style>.
400	ValidationException	tile resolution part of 'Y' must be at least 1	Tile resolution part of 'Y' must be at least 1.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	tile resolution part of 'Y' must not be greater than max resolution of style '<Style>' (<Maximum Resolution>)	Tile resolution part of 'Y' must not be greater than maximum resolution of style <Style> (max <MaxResolution>).
400	ValidationException	file type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style '<Style>'	File type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style <Style>.
400	ValidationException	file type extension part of 'Y' must be omitted for style '<Style>'	File type extension part of 'Y' must be omitted for style <Style>.
400	ValidationException	y-coordinate part of 'Y' must be an integer in the range $0..2^{\text{Zoom}} - 1$ ($0..<MaxTileCoordinate>$)	y-coordinate part of 'Y' must be an integer in the range $0..2^{\text{Zoom}} - 1$ ($0..<MaxTileCoordinate>$).
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	Invalid token	'NextToken' must be a valid token.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Unsupported Map Style: <Style>	<Style> is not a supported map style. For more information about list of supported map styles, see https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html .
402	ServiceQuotaExceededException	Map resources per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Map resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Map already exists: <MapName>.

Tracker

Tracker

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		<p>internal server error</p> <p>unable to retrieve point from the storage</p> <p>unable to verify tracker</p> <p>Error processing List request</p>	
404	ResourceNotFoundException	<p>tracker not found: <TrackerName></p> <p>Tracker with name <TrackerName> was not found</p>	Tracker not found: <TrackerName>.
404	ResourceNotFoundException	association not found: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn >	Association between tracker <TrackerName> and consumer <ConsumerArn> is not found.
400	ValidationException	'ConsumerArn' must refer to a geofence collection resource	'ConsumerArn' must refer to a geofence collection resource.
400	ValidationException	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated to	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated with.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker is it associated to	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is associated with.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Nothing to update.	At least one of the following fields must be set: 'Description', 'PositionFiltering'
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	request.TrackerName not found on request	'TrackerName ' is a required field.
400	ValidationException	no deviceId parameter found	'DeviceId' is a required field.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	provided start time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ“	'StartTimeInclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	provided end time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ	'EndTimeExclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	end time must be after start time	'EndTimeExclusive' must be after 'StartTimeInclusive'.
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state found. For more information about how key state affects the use of a KMS key, see https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.
402	ServiceQuotaExceededException	Tracker <TrackerName> may not have more than <Max> consumer associations	Tracker resource may not have more than <Max> consumer associations. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
402	ServiceQuotaExceededException	Trackers per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Tracking resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
409	ConflictException	association already exists: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn>	An association already exists between tracker <TrackerName> and consumer <Consumer Arn>.
409	ConflictException	Tracker already exists: <TrackerName>	Tracker already exists: <TrackerName>.

Routen

Routen

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	Resource not found	Route calculator not found: <RouteCalculatorName>.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	'DataSource' must be one of: Here, Esri, Grab	'DataSource' must be one of Esri, Grab, Here.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	<PricingPlan> pricing plan is not supported	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' <DataSourceName> must only be used in following regions: ap-southeast-1.
400	ValidationException	PricingPlan must be 'RequestBasedUsage'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DeparturePositions[0][0]' must be between -180 and 180	'DeparturePositions[0][0]' must be between -180 and 180.
400	ValidationException	'DeparturePositions[0][1]' must be between -90 and 90	'DeparturePositions[0][1]' must be between -90 and 90.
400	ValidationException	'DestinationPositions[0][0]' must be between -180 and 180	'DestinationPositions[0][0]' must be between -180 and 180.
400	ValidationException	'DestinationPositions[0][1]' must be between -90 and 90.	'DestinationPositions[0][1]' must be between -90 and 90

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'DepartNow' may not be true if 'Departur eTime' is set	Only one of 'DepartNo w' or 'DepartureTime' may be set.
400	ValidationException	'<TravelModeOption >' may not be set when 'TravelMode' has value <TravelMo deOption>	'<TravelModeOption >' must not be set when 'TravelMode' has value <TravelMo deOption>.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Walking	'CarModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Walking	'TruckModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Car	'TruckModeOptions' must not be set when 'TravelMode' has value Car.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Truck	'CarModeOptions' must not be set when 'TravelMode' has value Truck.
400	ValidationException	At least one of [Height, Length, Width] must be set in 'TruckModeOptions. Dimensions'	At least one of the following attribute must be set in TruckModeOptions.D imensions: Height, Length, Width.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	At least one of [Total] must be set in 'TruckModeOptions.Weight'	At least one of the following attribute must be set in TruckModeOptions.Weight: Total.
400	ValidationException	'DeparturePositions' count must be 10 or less with DataSource set to Esri	'DeparturePositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DestinationPositions' count must be 10 or less with DataSource set to Esri	'DestinationPositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DeparturePositions[0]' is more than 40km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 40 km away from 'DestinationPositions[0]'.
400	ValidationException	'DeparturePositions[0]' is more than 400km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 400 km away from 'DestinationPositions[0]'.
400	ValidationException	DeparturePositions[0] is contained within an unsupported region. Korea is not supported for CalculateRouteMatrix with the provider Esri.	DeparturePositions[0] is located in Korea, which is not supported when using CalculateRouteMatrix with data provider Esri.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'<HereTruckDimension>' must be between <Min> and <Max> <Unit>	'HereTruckDimension' must be between <Min> and <Max> <Unit>.
400	ValidationException	'WaypointPositions[0][0]' must be between -180 and 180	'WaypointPositions[0][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[0][1]' must be between -90 and 90	'WaypointPositions[0][1]' must be between -90 and 90.
400	ValidationException	'WaypointPositions[1][0]' must be between -180 and 180	'WaypointPositions[1][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[1][1]' must be between -90 and 90	'WaypointPositions[1][1]' must be between -90 and 90.
400	ValidationException	No road segment could be matched for one or more coordinates within a radius (1km)	One or more provided positions are more than 1 km from the nearest road segment.
400	ValidationException	Some positions in the request are unreachable	Some positions in the request are unreachable.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Total distance between all waypoints must be not be greater than 40km for DataSource Esri when using TravelMode Walking	Total distance between all route positions must not be greater than 40 km for 'DataSource' Esri and 'TravelMode' Walking.
400	ValidationException	Total distance between all waypoints must be not be greater than 400km for DataSource Esri	Total distance between all route positions must not be greater than 400 km for 'DataSource' Esri.
400	ValidationException	Following positions in the request are unreachable: <UnreachablePositions>	The following positions are unreachable: <UnreachablePositions>.
400	ValidationException	'DepartureTime' contains a badly-formatted timestamp	'DepartureTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	'TravelMode' <TravelMode> is not supported by <DataProvider>	'TravelMode' <TravelMode> not supported by data provider <DataProvider>.
400	ValidationException	'DeparturePositions' must be set	'DeparturePositions' must not be empty.
400	ValidationException	'DestinationPositions' must be set	'DestinationPositions' must not be empty.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Some inputs in the request are invalid	Some inputs in the request are invalid.
400	ValidationException	No route found between position <FirstPosition> and position <SecondPosition>	No route found between position <FirstPosition> and position <SecondPosition>.
400	ValidationException	No route found	No route found. For more information, see https://developer.amazon.com/documentation/routing-api/dev_guide/topics/notice.html .
400	ValidationException	No route found	No route found.
402	ServiceQuotaExceededException	Route calculators per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Route calculator resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Route calculator already exists: <RouteCalculatorName>.

Metadaten

Metadaten

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Error Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	APIKey not found	Api key not found: <APIKeyName>.
404	ResourceNotFoundException	APIKeyID not found	ApiKeyId not found: <APIKeyID>.
400	ValidationException	Either ExpireTime or NoExpiry must be provided	At least one of the following fields must be set: 'ExpireTime', 'NoExpiry'.
400	ValidationException	NoExpiry cannot be set to false if no ExpireTime is provided	'ExpireTime' must be set when 'NoExpiry' has value false.
400	ValidationException	ExpireTime cannot be set if NoExpiry is true	'ExpireTime' must not be set when 'NoExpiry' has value true.
400	ValidationException	Expire time '<ExpireTimeValue>' is not a valid time format	'ExpireTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	Expire time '<ExpireTimeValue>' cannot be in the past when creating a key	'ExpireTime' must not be in the past.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	The API Key %s has been recently used and the requested update may impact current usage. Specify ForceUpdate=true to update the API Key configuration.	This update may cause some users to lose API access. Because this API Key has been used in the last 7 days, you must set 'ForceUpdate' to true to confirm this change.
400	ValidationException	Expire time '<ExpireTimeValue>' must not be more than 1 minute in the past	'ExpireTime' must not be more than 1 minute in the past.
400	ValidationException	Description, ExpireTime, NoExpiry and Restrictions can't all be empty	At least one of the following fields must be set: 'Description', 'ExpireTime', 'NoExpiry', 'Restrictions'.
400	ValidationException	API Key expired	'ApiKeyId' must not be expired.
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

Geofences

Geozäune

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	<p>internal server error</p> <p>Internal server error</p> <p>Unsupported geofence geometry encountered</p> <p>geometry marshal error</p> <p>geometry load error</p> <p>unable to get geofence collection</p> <p>unable to delete geofences</p> <p>unable to retrieve geofence</p> <p>Error processing List request</p>	<p>Internal server error.</p> <p>Try again later.</p>
404	ResourceNotFoundException	<p>collection not found: <GeofenceCollectionName></p> <p><GeofenceCollectionName> geofence collection not found</p> <p>Resource not found error</p>	<p>Geofence Collection not found: <GeofenceCollectionName>.</p>

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		no geofence with given name found	
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	KMS key must be a symmetric CMK. Invalid usage type: <UsageType>	KMS key must be a symmetric Customer Master Key (CMK). Invalid usage type <UsageType>. For how to create a symmetric CMK, refer to https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk .
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	PricingPlanDataSource cannot be updated without updating PricingPlan	'PricingPlan' must be provided to update 'PricingPlanDataSource'.
400	ValidationException	nothing to update	At least one of the following fields must be set: 'Description'

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state <InvalidState>. For more information about how key state affects the use of a KMS key, see https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.
400	ValidationException	duplicate geofence ID in batch	'GeofenceId' <DuplicatedGeofenceId> is duplicated in batch.
400	ValidationException	missing GeofenceId	'GeofenceId' must not be empty.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000km.
400	ValidationException	no geofence with given name found	Geofence not found: <CollectionName>.
400	ValidationException	Geometry must contain either a Circle or Polygon, not both	Only one of 'Circle' or 'Polygon' may be set within 'Geometry'.
400	ValidationException	Geometry must contain a Polygon or a Circle	One of 'Polygon' or 'Circle' must be set within 'Geometry'.
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0m.
400	ValidationException	empty polygon	'Geometry.Polygon' must not be empty.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	empty polygon ring	'Geometry.Polygon' must not be empty.
400	ValidationException	circle can not cross antimeridian	'Geometry.Circle' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not cross antimeridian	'Geometry.Polygon' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not have interior rings (holes), remove holes	'Geometry.Polygon' must not have interior rings (holes). For more information about interior rings see https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3 .
400	ValidationException	polygon ring is not closed	'Geometry.Polygon' contains an open ring. Close the ring by ensuring the first and last positions are equal.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	polygon ring has more than 1000 vertices	'Geometry.Polygon' must not have more than 1000 vertices.
400	ValidationException	polygon ring has fewer than 4 positions	Number of vertices in 'Geometry.Polygon' must be greater or equal to 4.
400	ValidationException	invalid center	'Geometry.Circle.Center' must be a valid position (longitude/latitude pair).
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0 m.
400	ValidationException	longitude range should be between -180 and 180 degrees	Longitude must be between -180 and 180 degrees, but was set to <Provided Longitude>.
400	ValidationException	latitude range should be between -90 and 90 degrees	Latitude must be between -90 and 90 degrees, but was set to <Provided Longitude>.
400	ValidationException	polygon exterior ring is expected to be counter clockwise	'Geometry.Polygon' must be oriented counter-clockwise.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	polygon interior ring should be clockwise oriented	'Geometry.Polygon' must be oriented clockwise.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000 km.
400	ValidationException	timestamp.Parse() error	'SampleTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	invalid input	'SourceArn' must refer to a tracker resource.
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .
400	ValidationException	arn: not enough sections	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	invalid resource part	'SourceArn' must refer to a tracker resource.
402	ServiceQuotaExceededException	Geofence collections per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Geofence collection resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

Codebeispiele und Tutorials für die Arbeit mit Amazon Location Service

In diesem Thema finden Sie eine Liste mit Codebeispielen, Tutorials und Blogbeiträgen, die Ihnen helfen, mehr über Amazon Location Service zu erfahren. Jedes Codebeispiel enthält eine Beschreibung seiner Funktionsweise.

Weitere Beispiele finden Sie auf der [AWS GitHub Geospatial-Seite](#), der [AWS GitHub Beispielseite für Amazon Location](#) und auf der [AWS Blog-Website](#).

Note

Es ist gut, den Unterschied zwischen der AWS GitHub Geospatial-Seite und der AWS GitHub Beispielseite zu verstehen.

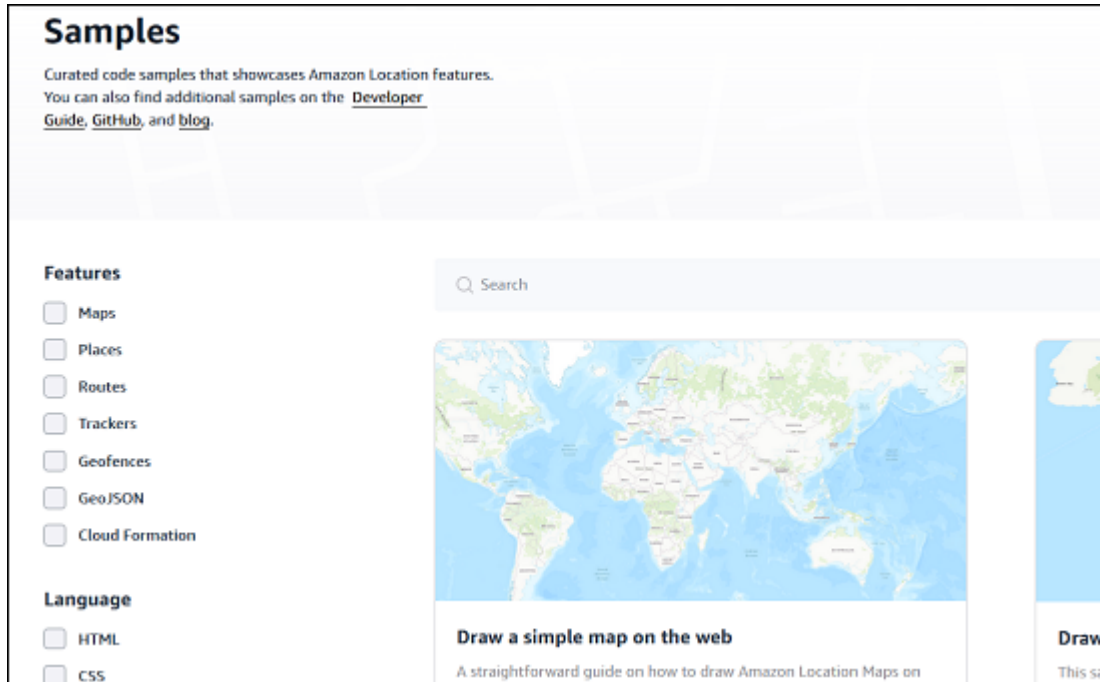
- Geospatial GitHub — Die [AWS GitHub Geospatial-Seite](#) enthält Beispiele, die vom Amazon Location Service Service-Team erstellt und verwaltet wurden.
- Beispiele GitHub — Die [AWS GitHub Musterseite für Amazon Location](#) enthält Beispiele, die für Amazon Location erstellt wurden, aber möglicherweise nicht aktiv verwaltet werden.

Das [Schnellstart-Tutorial](#) ist ein guter Ausgangspunkt, bevor Sie andere Beispiele verwenden, da es zeigt, wie Sie die Voraussetzungen erfüllen, die für die meisten Beispiele nützlich sind.

Themen

- [Demo-Website für Amazon-Standorte](#)
- [Tutorial: Schnellstart](#)
- [Tutorial: Datenbankanreicherung](#)
- [Beispiel: Explore App](#)
- [Beispiel: Gestalte eine Karte](#)
- [Beispiel: Markierungen zeichnen](#)
- [Beispiel: Zeichnet geclusterte Punkte](#)
- [Beispiel: Zeichne ein Polygon](#)
- [Beispiel: Ändern Sie die Kartensprache](#)
- [Blog: Benachrichtigungen zur voraussichtlichen Lieferzeit](#)
- [Beispiel: Aktualisierungen der Stream-Position](#)
- [Beispiel: Mobilanwendung für Geofencing und Tracking](#)

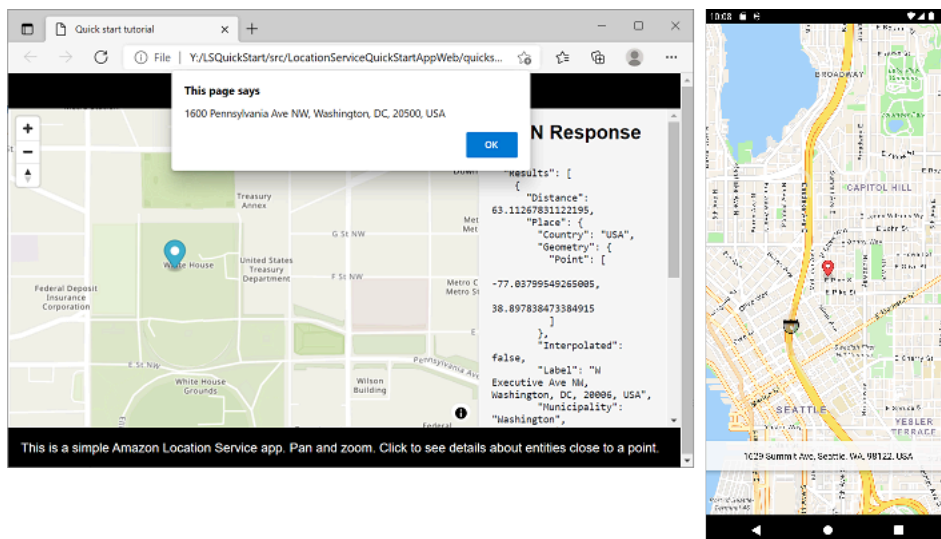
Demo-Website für Amazon-Standorte



Auf der Amazon Location [Demo-Website](#) können Sie Demos mit dem Quellcode von Amazon Location Service in Aktion sehen. Diese Website enthält eine [gehostete Webdemo](#) und auch eine Demo-App für [Android](#).

[Auf der Beispielseite der Website](#) finden Sie auch eine Vielzahl von Beispielen, die nach Funktionen, Sprache und Plattform gefiltert werden können.

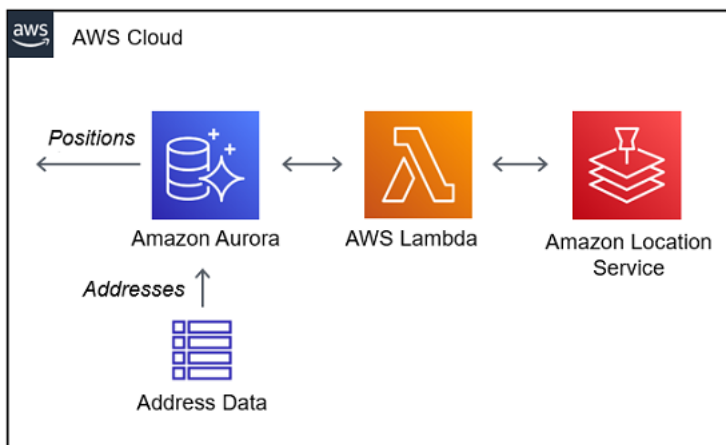
Tutorial: Schnellstart



Es gibt Schnellstart-Tutorials für Web-, iOS- und Android-Geräte. Für jede Plattform zeigt Ihnen das Tutorial, wie Sie einer Anwendung eine interaktive Karte hinzufügen und wie Sie die Amazon Location Service Service-APIs von Ihrer Anwendung aus aufrufen. Das Tutorial ist für JavaScript eine statische Webseite, Kotlin für eine Android-Telefonanwendung oder Swift für eine iOS-Anwendung verfügbar.

- JavaScript für einen Link zur Dokumentation einer statischen Webseite: [Erstellen einer Webanwendung](#)
- Link zur Dokumentation zu Kotlin für eine Android-Anwendung: [Schneller Start mit Amazon Location Service](#)
- Swift für einen Link zur iOS-App-Dokumentation: [Eine iOS-App erstellen](#)

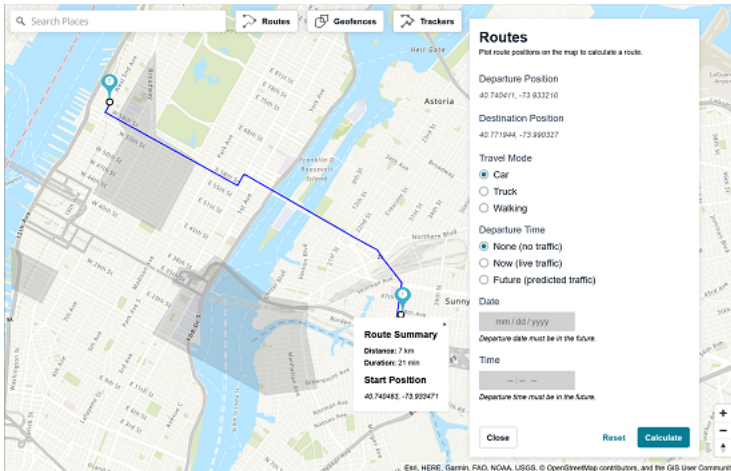
Tutorial: Datenbankanreicherung



In diesem Tutorial erfahren Sie, wie Sie Amazon Location Service verwenden, der von aufgerufen wird, um Adressen AWS Lambda zu normalisieren und Datensätze in einer Amazon Aurora Aurora-Datenbank um Längen- und Breitengrad zu erweitern. Verwendet Amazon Aurora und AWS Lambda.

Link zur Dokumentation: [Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service](#)

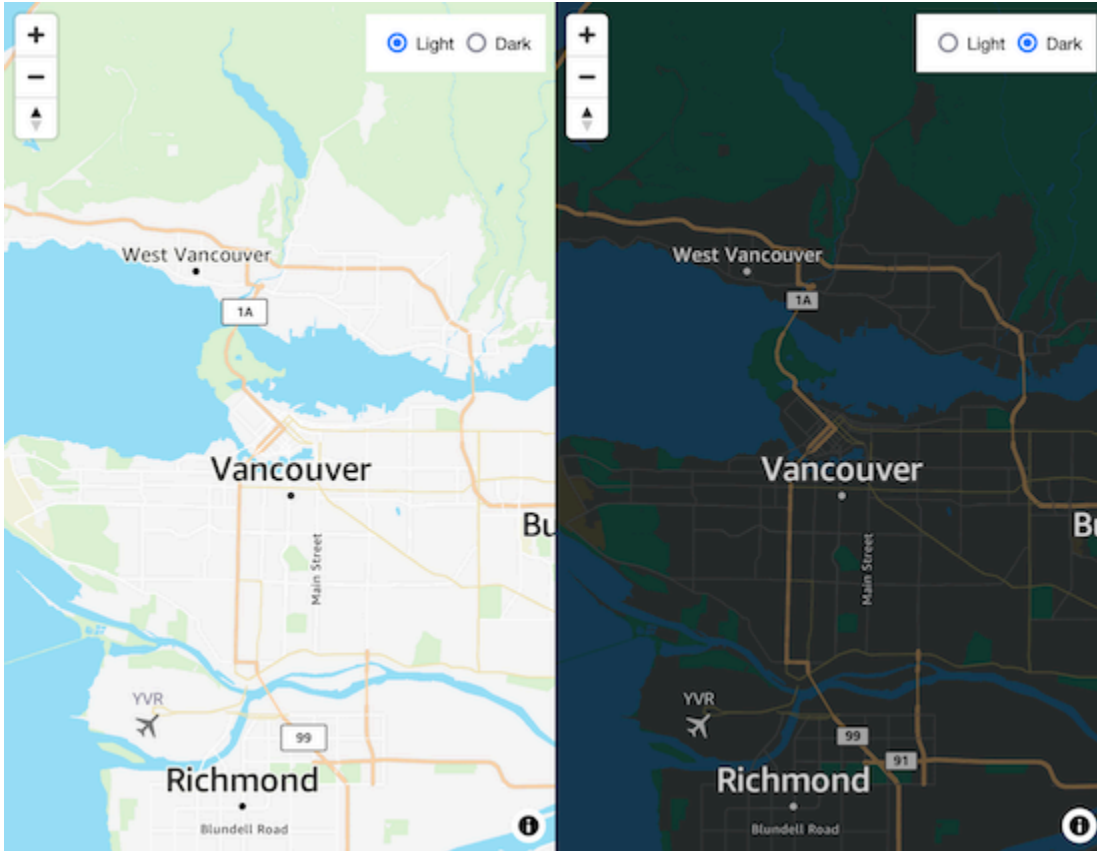
Beispiel: Explore App



Eine der besten Möglichkeiten, mehr über die Funktionen von Amazon Location Service zu erfahren, ist die Nutzung der [Explore-Funktion](#) in der Amazon Location-Konsole. Dieses vollständige Beispiel für eine Webanwendung ahmt die Funktionen von Karten, Orten, Routen, Geofences und Trackern aus der Konsole nach, um Ihnen zu zeigen, wie Sie diese Funktionen in Ihrer eigenen App neu erstellen können. Verwendet Amplify, React und JavaScript.

GitHub Link zu den Beispielen: [Erkunden Sie die Beispielanwendung](#)

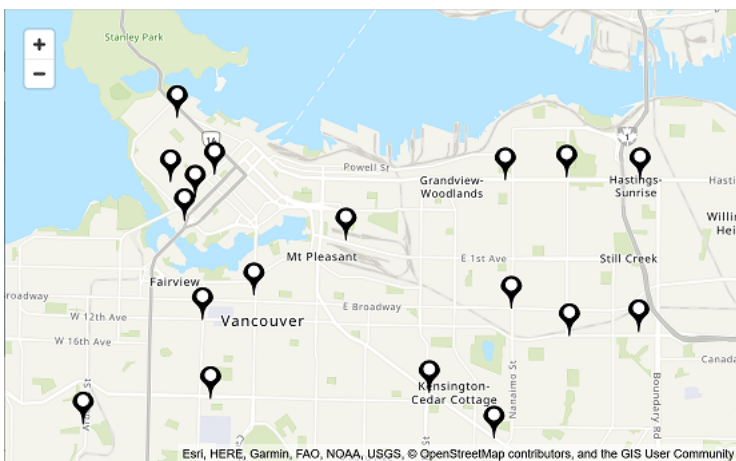
Beispiel: Gestalte eine Karte



Dieses Codebeispiel zeigt, wie Sie mit MapLibre in zwischen einer Satellitenkarte und einer Vektor-Straßenkarte wechseln können JavaScript. Verwendet MapLibre, den Amazon Location-Authentifizierungshelfer und JavaScript.

Geospatialer GitHub Link: [Interaktive Karte mit Stilwechsel](#)

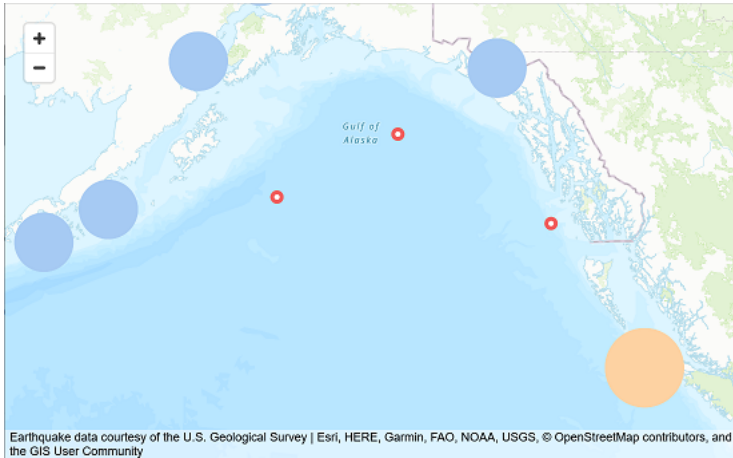
Beispiel: Markierungen zeichnen



Dieses Codebeispiel zeigt Amazon Locker-Standorte in Vancouver, BC, Kanada. Es zeigt, wie Markierungen an Punktpositionen gezeichnet werden. Verwendet Node.js MapLibre, React, den Amazon Location-Authentifizierungshelfer und JavaScript.

Geospatialer GitHub Link: [Interaktive Karte mit Markierungen an Punkten](#)

Beispiel: Zeichnet geclusterte Punkte



Dieses Codebeispiel zeigt anhand von USGS-Erdbebendaten, wie Punkte gezeichnet werden, die sich gruppieren, wenn sie auf der Karte nahe beieinander liegen. Verwendet Node.js MapLibre, React, Amplify und JavaScript.

GitHub Link zu den Beispielen: [Interaktive Karte mit Punktclustern](#)

Beispiel: Zeichne ein Polygon



Dieses Codebeispiel zeigt, wie ein Polygon auf der Karte gezeichnet wird. Verwendet Node.js MapLibre, React, den Amazon Location-Authentifizierungshelfer, und JavaScript.

Geospatialer GitHub Link: [Interaktive Karte mit Polygonen](#)

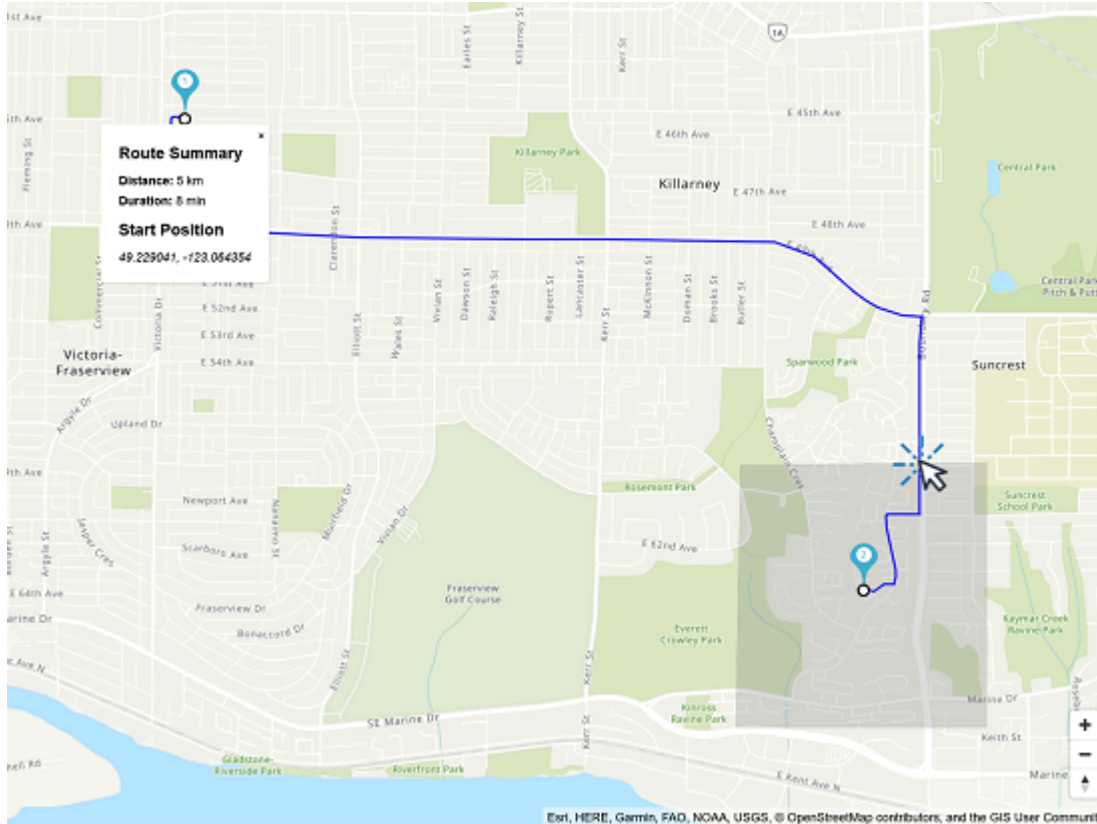
Beispiel: Ändern Sie die Kartensprache



Dieses Codebeispiel zeigt, wie Sie die Anzeigesprache von Karten in Amazon Location ändern können. Verwendet Amplify, React und MapLibre.

GitHub Link zu den Beispielen: Beispiel für [die Kartensprache ändern](#)

Blog: Benachrichtigungen zur voraussichtlichen Lieferzeit



Dieser Blogbeitrag zeigt verschiedene Möglichkeiten, Kunden über geschätzte Lieferzeiten zu informieren. Es wird erklärt, wie Routen verwendet werden, um die geschätzte Fahrzeit anzuzeigen, und anschließend mithilfe von Trackern und Geofences benachrichtigt werden, wenn sich ein Fahrer dem Kunden nähert. Verwendet Amplify, React EventBridge, Amazon und Amazon Simple Notification Service (Amazon SNS).

Blog-Link: [Voraussichtliche Ankunftszeit und Näherungsbenedachrichtigungen](#)

Beispiel: Aktualisierungen der Stream-Position



Kinesis Stream To Tracker App: Dieses Beispiel zeigt, wie Kinesis Data Stream verwendet wird, um Tracker-Updates mit Amazon Location Service zu posten. Das Beispiel ist eine bereitstellbare, in Python geschriebene Lambda-Anwendung, die in einen Kinesis Data Stream integriert werden kann, um die Kinesis-Ereignisse zu verarbeiten und Gerätepositionen stapelweise zu aktualisieren.

Link zum Repository: [Amazon Location Amazon Kinesis Data Streams Stream zur Tracker-App](#)

Weitere Informationen zu Tracking und Geofences finden Sie in der Dokumentation zu [Geofences](#) and Trackers. [Entwickler können die App gemäß der Dokumentation zum Serverless Application Repository von AWS oder direkt von der Lambda-Konsole aus bereitstellen.](#)

Beispiel-App für Gerätepositions-Streaming: Dieses Codebeispiel zeigt, wie Gerätepositionsdaten in einen Kinesis Data Stream gestreamt werden und wie Geofence-Benachrichtigungen funktionieren. Diese App hängt davon ab, dass die oben aufgeführte Kinesis Stream to Tracker-Beispiel-App ausgeführt wird, damit die gestreamten Tracker-Positionen in Amazon Location Service aktualisiert werden können.

Repository-Link: [Streaming-Beispiel-App Amazon Location Device Position](#)

Beispiel: Mobilanwendung für Geofencing und Tracking

Diese Beispielanwendung zeigt, wie ein Tracker und Geofence mithilfe einer Kombination aus Lambda AWS IoT - und Amazon Location-Funktionen interagieren. Es gibt Tutorials für iOS und Android.

Link zum Tutorial: [Beispiel für eine mobile Geofence](#) - und Tracker-Anwendung

So verwenden Sie Amazon Location Service

Sie können die Funktionen von Amazon Location Service verwenden, um geografische und standortbezogene Aufgaben auszuführen. Sie können diese Aufgaben dann kombinieren, um komplexere Anwendungsfälle wie GeoMarketing, Bereitstellung und Komponentenverfolgung zu bewältigen.

Wenn Sie bereit sind, Standortfunktionen in Ihre Anwendung zu integrieren, verwenden Sie die folgenden Methoden, um die Amazon Location Service-Funktionalität zu verwenden, abhängig von Ihren Zielen und Einschränkungen:

- **Nutzungstools** – Wenn Sie mit Amazon Location-Ressourcen experimentieren möchten, sind die folgenden Tools die schnellste Möglichkeit, auf die APIs zuzugreifen und diese auszuprobieren:
 - Die [Amazon Location-Konsole](#) bietet eine Vielzahl von Quick-Access-Tools. Sie können Ihre -Ressourcen erstellen und verwalten und die APIs auf [der Seite Erkunden](#) ausprobieren. Die Konsole ist auch nützlich, um Ressourcen (in der Regel eine einmalige Aufgabe) zu erstellen, um die Verwendung einer der anderen später beschriebenen Methoden vorzubereiten.
 - Mit der [-AWSBefehlszeilenschnittstelle](#) (CLI) können Sie Ressourcen erstellen und über ein Terminal auf die Amazon Location APIs zugreifen. Die AWS CLI übernimmt die Authentifizierung, wenn Sie sie mit Ihren -Anmeldeinformationen konfigurieren.
 - Sie können [Codebeispiele und Tutorials](#) sehen, die zeigen, wie Aufgaben mit den Amazon Location Service APIs ausgeführt werden. Dazu gehört [ein Beispiel](#), das einen Großteil der Funktionalität der Seite Erkunden in der -Konsole nachahmt.
- **Plattform-SDKs** – Wenn Sie keine Daten auf einer Karte visualisieren, können Sie jedes der [AWS Standard-Tools](#) verwenden, um auf zu bauenAWS.
 - Die folgenden SDKs sind verfügbar: C++, Go, Java JavaScript, .NET, Node.js, PHP, Python und Ruby.
- **Frontend-SDKs und Bibliotheken** – Wenn Sie Amazon Location verwenden möchten, um eine Anwendung auf einer mobilen Plattform zu erstellen oder Daten auf einer Karte auf einer beliebigen Plattform zu visualisieren, haben Sie die folgenden Optionen:
 - Die AWS Amplify Bibliotheken integrieren Amazon Location in [iOS-](#), [Android-](#) und -[JavaScript](#)Webanwendungen.
 - Mit den MapLibre Bibliotheken können Sie clientseitige Zuordnungen in [iOS-](#), [Android-](#) und [JavaScript](#) Webanwendungen mithilfe von Amazon Location rendern.

- Mit Tangram ES-Bibliotheken können Sie 2D- und 3D-Zuweisungen aus Vektordaten mithilfe von OpenGL ES in [iOS](#)- und [Android](#)-Webanwendungen rendern. Es gibt auch Tangram für [JavaScript](#) Webanwendungen.
- Senden von direkten HTTPS-Anfragen – Wenn Sie mit einer Programmiersprache arbeiten, für die kein SDK verfügbar ist, oder wenn Sie mehr Kontrolle darüber haben möchten, wie Sie eine Anfrage an sendenAWS, können Sie auf Amazon Location zugreifen, indem Sie direkte HTTPS-Anfragen senden, die durch den Signature Version 4-Signaturprozess authentifiziert wurden. Weitere Informationen zum [Signaturprozess mit Signature Version 4](#) finden Sie unter Allgemeine AWS-Referenz.

In diesem Kapitel werden viele Aufgaben beschrieben, die Anwendungen unter Verwendung von Standortdaten gemeinsam sind. Im Abschnitt [Häufige Anwendungsfälle](#) wird beschrieben, wie Sie diese mit anderen -AWSServices kombinieren, um komplexere Anwendungsfälle zu erzielen.

Themen

- [Voraussetzungen für die Nutzung von Amazon Location Service](#)
- [Verwenden von Amazon Location Maps in Ihrer Anwendung](#)
- [Suchen von Orts- und Geolocation-Daten mit Amazon Location](#)
- [Berechnen von Routen mit Amazon Location Service](#)
- [Geofencing eines Interessengebiets mithilfe von Amazon Location](#)
- [Markieren Ihrer Amazon Location Service-Ressourcen](#)
- [Gewähren des Zugriffs auf Amazon Location Service](#)
- [Überwachen von Amazon Location Service](#)
- [Erstellen von Amazon Location Service-Ressourcen mit AWS CloudFormation](#)

Voraussetzungen für die Nutzung von Amazon Location Service

In diesem Abschnitt wird beschrieben, was Sie tun müssen, um Amazon Location Service nutzen zu können. Sie müssen über einen Amazon-Standort verfügen AWS-Konto und Zugriff darauf für Benutzer eingerichtet haben, die ihn verwenden möchten.

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal](#).

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter [Einen Berechtigungssatz erstellen](#).AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen](#).

Zugriff auf Amazon Location Service gewähren

Ihre Benutzer ohne Administratorrechte haben standardmäßig keine Berechtigungen. Bevor sie auf Amazon Location zugreifen können, müssen Sie die Erlaubnis erteilen, indem Sie eine IAM-Richtlinie mit bestimmten Berechtigungen anhängen. Achten Sie darauf, dass Sie bei der Gewährung des Zugriffs auf Ressourcen das Prinzip der geringsten Rechte einhalten.

Note

Informationen darüber, wie Sie nicht authentifizierten Benutzern Zugriff auf die Funktionen von Amazon Location Service gewähren können (z. B. in einer webbasierten Anwendung), finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#)

Die folgende Beispielrichtlinie gibt einem Benutzer die Erlaubnis, auf alle Amazon Location-Operationen zuzugreifen. Weitere Beispiele finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

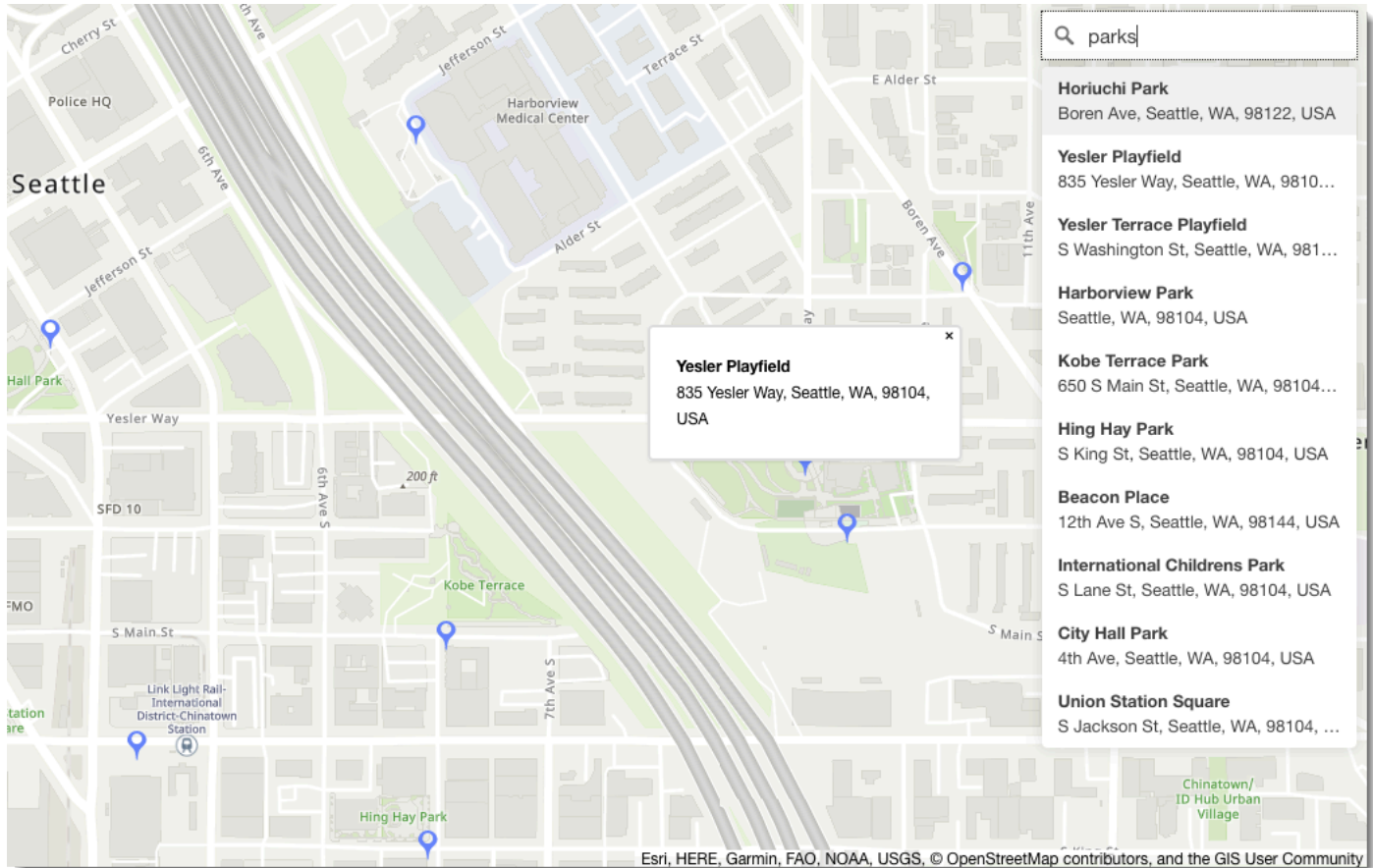
- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter [Erstellen einer Rolle für einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Wenn Sie Anwendungen erstellen, die Amazon Location Service verwenden, benötigen einige Benutzer möglicherweise nicht authentifizierten Zugriff. Informationen zu diesen Anwendungsfällen finden Sie unter [Aktivieren eines nicht authentifizierten Zugriffs mit Amazon](#) Cognito.

Verwenden von Amazon Location Maps in Ihrer Anwendung

Amazon Location Maps sind kostengünstig und interaktiv. Sie können eine vorhandene Karte in Ihrer Anwendung ersetzen, um Geld zu sparen, oder eine neue hinzufügen, um standortbasierte Daten visuell anzuzeigen, z. B. Ihren Standort.



Mit Amazon Location Service können Sie einen Datenanbieter für Kartenoperationen auswählen, indem Sie eine Kartenressource erstellen und konfigurieren. Die Kartenressource konfiguriert den Datenanbieter und den Stil, der zum Rendern der Karte verwendet wird.

Nachdem Sie Ihre Ressource erstellt haben, können Sie Anfragen direkt über das AWS SDK oder mithilfe einer Bibliothek senden, die speziell für das Rendern von Karten in Ihrer Umgebung erstellt wurde.

Note

Eine Übersicht über Kartenkonzepte finden Sie unter [Zuordnungen](#).

Themen

- [Voraussetzungen](#)
- [Anzeigen einer Karte in Ihrer Anwendung](#)
- [Zeichnen von Datenfunktionen auf einer Karte](#)
- [Festlegen von Erweiterungen für eine Zuordnung mit MapLibre](#)
- [Verwalten Ihrer Kartenressourcen](#)

Voraussetzungen

Bevor Sie eine Zuordnung in Ihrer Anwendung anzeigen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Kartenressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Kartenressource

Um eine Karte in Ihrer Anwendung zu verwenden, benötigen Sie eine Kartenressource, die den Kartenstil und den Datenanbieter angibt, der in Ihren Karten verwendet werden soll.

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Sie können eine Kartenressource mithilfe der Amazon Location Service-Konsole, der AWS CLI oder der Amazon Location APIs erstellen.

Console

So erstellen Sie eine Kartenressource mit der Amazon Location Service-Konsole

1. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
2. Fügen Sie einen Namen und eine Beschreibung für die neue Kartenressource hinzu.
3. Wählen Sie einen Kartenstil aus.

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

4. Wählen Sie aus der zu [Politische Ansichten](#) verwendenden aus.
5. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
6. Damit Ihre Benutzer den Stil wechseln können (z. B. damit sie zwischen Satellitenbildern und Vektorstil wechseln können), müssen Sie für jeden Stil eine Kartenressource erstellen.

Sie können Ressourcen mit Kartenstilen löschen, die Sie nicht auf der [Karten-Startseite](#) in der -Konsole verwenden möchten.

API

So erstellen Sie eine Kartenressource mithilfe der Amazon Location APIs

Verwenden Sie die [CreateMap](#) Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Kartenressource namens *ExampleMap* unter Verwendung des *VectorEsriStreets* Kartenstils.

```
POST /maps/v0/maps HTTP/1.1
```

```
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

AWS CLI

So erstellen Sie eine Kartenressource mit -AWS CLIBefehlen

Verwenden Sie den [create-map](#)-Befehl.

Im folgenden Beispiel wird eine Kartenressource namens *ExampleMap* mit *VectorEsriStreets* als Kartenstil erstellt.

```
aws location \
  create-map \
  --configuration Style="VectorEsriStreets" \
  --map-name "ExampleMap"
```

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Kartenressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, müssen Sie auswählen, wie Sie Ihre Anfragen authentifizieren würden.

Note

Die meisten Frontend-Anwendungen für Karten erfordern nicht authentifizierten Zugriff auf die Karten oder andere Funktionen von Amazon Location Service. Abhängig von Ihrer Anwendung möchten Sie möglicherweise AWS Signature v4 verwenden, um Anfragen zu authentifizieren, oder Sie können Amazon Cognito- oder Amazon-Location-API-Schlüssel für die nicht authentifizierte Verwendung verwenden. Weitere Informationen zu all diesen Optionen finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Anzeigen einer Karte in Ihrer Anwendung

Dieser Abschnitt enthält Tutorials zur Verwendung von Karten-Rendering-Tools, um eine Karte in Ihrer mobilen oder Webanwendung anzuzeigen, wenn Sie Amazon Location APIs verwenden. Wie im [So verwenden Sie Amazon Location Service](#) Thema erwähnt, haben Sie eine Auswahl an Bibliotheken, die Sie beim Rendern von Karten mit Amazon Location verwenden können, einschließlich Amplify MapLibre, und Tangram.

Führen Sie einen der folgenden Schritte aus, um eine Zuordnung in Ihrer Anwendung anzuzeigen:

- Die direkteste Möglichkeit, eine Karte in Ihren Web- und mobilen Frontend-Anwendungen anzuzeigen, ist die Verwendung von MapLibre. Sie können den [MapLibre Tutorials](#) oder sogar dem [Schnellstart-Tutorial](#) folgen, um zu erfahren, wie Sie verwenden MapLibre.
- Wenn Sie bereits AWS Amplify Entwickler sind, sollten Sie das Amplify Geo SDK verwenden. Weitere Informationen finden Sie im [Amplify-Tutorial](#) .
- Wenn Sie bereits ein Benutzer von Tangram sind und es weiterhin zum Rendern Ihrer Karte verwenden möchten, während Sie zu Amazon Location Service wechseln, folgen Sie dem [Tutorial zu Tangram](#) .

Themen

- [Verwenden der MapLibre Bibliothek mit Amazon Location Service](#)
- [Verwenden der Amplify-Bibliothek mit Amazon Location Service](#)

- [Verwenden von Tangram mit Amazon Location Service](#)

Verwenden der MapLibre Bibliothek mit Amazon Location Service

Die folgenden Tutorials führen Sie durch die Verwendung der - MapLibre Bibliothek mit Amazon Location.

Themen

- [Verwenden von MapLibre microSD JS mit Amazon Location Service](#)
- [Verwenden des MapLibre nativen SDK für Android mit Amazon Location Service](#)
- [Verwenden des MapLibre nativen SDK für iOS mit Amazon Location Service](#)

Verwenden von MapLibre microSD JS mit Amazon Location Service

Verwenden Sie [MapLibre microSD JS](#), um clientseitige Maps in Webanwendungen einzubetten.

MapLibre microSD JS ist eine Open-Source- JavaScript Bibliothek, die mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre microSD JS in einen einfachen HTML oder eine einfache JavaScript Anwendung integrieren, um anpassbare und reaktionsschnelle clientseitige Karten einzubetten.

In diesem Tutorial wird beschrieben, wie MapLibre microSD JS in Amazon Location in einem einfachen HTML und einer einfachen JavaScript Anwendung integriert wird. Die gleichen Bibliotheken und Techniken, die in diesem Tutorial vorgestellt werden, gelten auch für Frameworks wie [React](#) und [Angular](#) .

Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Gerüsterstellung

In diesem Tutorial wird eine Webanwendung erstellt, die verwendet, JavaScript um eine Karte auf einer HTML-Seite zu erstellen.

Erstellen Sie zunächst eine HTML-Seite (`index.html`), die den Container der Karte enthält:

- Geben Sie ein `-div`Element mit einem `id` von `einmap`, um die Dimensionen der Karte auf die Kartenansicht anzuwenden. Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie Ihrer Anwendung die folgenden Abhängigkeiten hinzu:

- MapLibre microSD JS (v3.x) und das zugehörige CSS.
- Der Amazon-Standort [JavaScript Authentifizierungshelfer](#).

```
<!-- CSS dependencies -->
<link
  href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
  rel="stylesheet"
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
<script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></script>
<script>
  // application-specific code
</script>
```

Dadurch wird eine leere Seite mit dem Container der Zuordnung erstellt.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit JavaScript:

1. Geben Sie die Namen und Kennungen Ihrer Ressourcen ein.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service Map name
const mapName = "ExampleMap";
```

2. Instanzieren Sie einen Anmeldeinformationsanbieter mithilfe des nicht authentifizierten Identitätspools, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten](#) erstellt haben. Wir werden dies in eine Funktion namens `initializeMap`, die auch anderen Karteninitialisierungscode enthält, der im nächsten Schritt hinzugefügt wird

```
// extract the Region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":")[0];

async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await
  amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // ... more here, later
}
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie können die anfängliche Kartenposition anpassen, zusätzliche Steuerelemente hinzufügen und Daten überlagern.

```
async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // Initialize the map
  const map = new maplibregl.Map({
    container: "map",
```

```
center: [-123.1187, 49.2819], // initial map centerpoint
zoom: 10, // initial map zoom
style: 'https://maps.geo.{$region}.amazonaws.com/maps/v0/maps/{$mapName}/style-
descriptor',
...authHelper.getMapAuthenticationOptions(), // authentication, using cognito
});

map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `sources.grabmaptiles.attribution` Schlüsseln `sources.esri.attribution` `sources.here.attribution`, und enthalten. MapLibre GL JS stellt automatisch die Attribution bereit. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Ausführen der Anwendung

Sie können diese Beispielanwendung ausführen, indem Sie sie auf einem lokalen Webserver verwenden oder sie in einem Browser öffnen.

Um einen lokalen Webserver zu verwenden, können Sie `npx` verwenden, da es als Teil von Node.js installiert ist. Sie können `npx serve` innerhalb desselben Verzeichnisses wie `index.html` verwenden. Dies dient der Anwendung auf `localhost:5000`.

Note

Wenn die Richtlinie, die Sie für Ihre nicht authentifizierte Amazon Cognito-Rolle erstellt haben, eine `referrer` Bedingung enthält, werden Sie möglicherweise daran gehindert, mit `localhost: URLs` zu testen. In diesem Fall können Sie mit einem Webserver testen, der eine URL bereitstellt, die sich in Ihrer Richtlinie befindet.

Nach Abschluss des Tutorials sieht die endgültige Anwendung wie im folgenden Beispiel aus.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body {
        margin: 0;
      }
      #map {
        height: 100vh;
      }
    </style>
  </head>

  <body>
    <!-- map container -->
    <div id="map" />
    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></
script>
    <script>
      // configuration
      const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd"; //
Cognito Identity Pool ID
      const mapName = "ExampleMap"; // Amazon Location Service Map Name

      // extract the region from the Identity Pool ID
      const region = identityPoolId.split(":")[0];

      async function initializeMap() {
        // Create an authentication helper instance using credentials from Cognito
        const authHelper = await
amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

        // Initialize the map
        const map = new maplibregl.Map({
          container: "map",
          center: [-123.115898, 49.295868],
          zoom: 10,
```



```
        style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/
style-descriptor`,
        ...authHelper.getMapAuthenticationOptions(),
    });
    map.addControl(new maplibregl.NavigationControl(), "top-left");
}

    initializeMap();
</script>
</body>
</html>
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte mit dem von Ihnen gewählten Kartenstil angezeigt. Dieses Beispiel ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#).

Verwenden des MapLibre nativen SDK für Android mit Amazon Location Service

Verwenden Sie [MapLibre das native](#) SDK, um interaktive Karten in Ihre Android-Anwendungen einzubetten.

Das MapLibre native SDK für Android ist eine Bibliothek, die auf [Mapbox Native](#) basiert und mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre das native SDK für Android integrieren, um interaktive Kartenansichten in skalierbare, anpassbare Vektorkarten in Ihre Android-Anwendungen einzubetten.

In diesem Tutorial wird beschrieben, wie Sie das MapLibre native SDK für Android in Amazon Location integrieren. Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Android Studio-Projekt aus der Vorlage Leere Aktivität.
2. Stellen Sie sicher, dass Kotlin für die Projektsprache ausgewählt ist.
3. Wählen Sie ein Mindest-SDK von API 14: Android 4.0 (IceSpeed Speed) oder höher aus.
4. Öffnen Sie Projektstruktur und navigieren Sie dann zu Datei > Projektstruktur..., um den Abschnitt Abhängigkeiten auszuwählen.
5. Wenn <Alle Module> ausgewählt ist, wählen Sie dann die Schaltfläche +, um eine neue Bibliotheksabhängigkeit hinzuzufügen.

6. Fügen Sie AWS Android SDK Version 2.20.0 oder höher hinzu. Beispiel:
`com.amazonaws:aws-android-sdk-core:2.20.0`
7. Fügen Sie das MapLibre native SDK für Android Version 9.4.0 oder höher hinzu. Beispiel:
`org.maplibre.gl:android-sdk:9.4.0`
8. Fügen Sie auf Projektebene Ihrer build.gradle-Datei das folgende Maven-Repository hinzu, um auf die MapLibre Pakete für Android zuzugreifen:

```
allprojects {
    repositories {
        // Retain your existing repositories
        google()
        jcenter()

        // Declare the repositories for MapLibre
        mavenCentral()
    }
}
```

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Ihrer AWS Region:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
    <string name="mapName">ExampleMap</string>
    <string name="awsRegion">us-east-1</string>
</resources>
```

Erstellen der Anwendung: Aktivitätslayout

Bearbeiten von `app/src/main/res/layout/activity_main.xml`:

- Fügen Sie ein `hinzuMapView` hinzu, das die Zuordnung wiedergibt. Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.
- Fügen Sie ein `hinzuTextView` hinzu, das die Zuordnung anzeigt.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapbox.mapboxsdk.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:mapbox_cameraTargetLat="49.2819"
        app:mapbox_cameraTargetLng="-123.1187"
        app:mapbox_cameraZoom="12"
        app:mapbox_uiAttribution="false"
        app:mapbox_uiLogo="false" />

    <TextView
        android:id="@+id/attributionView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#80808080"
        android:padding="5sp"
        android:textColor="@android:color/black"
        android:textSize="10sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="SmallSp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attributionsources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Nutzungsbedingungen für den Service](#).

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine Klasse mit dem Namen `SigV4Interceptor`, um AWS-Anforderungen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird bei dem HTTP-Client registriert, der zum Abrufen von Kartenressourcen verwendet wird, wenn die Hauptaktivität erstellt wird.

```
package aws.location.demo.okhttp

import com.amazonaws.DefaultRequest
import com.amazonaws.auth.AWS4Signer
import com.amazonaws.auth.AWSCredentialsProvider
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()

        if (originalRequest.url().host().contains("amazonaws.com")) {
            val signer = if (originalRequest.url().encodedPath().contains("@")) {
                // the presence of "@" indicates that it doesn't need to be double URL-
                encoded
                AWS4Signer(false)
            } else {
                AWS4Signer()
            }

            val awsRequest = toAWSRequest(originalRequest, serviceName)
            signer.setServiceName(serviceName)
            signer.sign(awsRequest, credentialsProvider.credentials)

            return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
        }
    }
}
```

```
        return chain.proceed(originalRequest)
    }

    companion object {
        fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
            // clone the request (AWS-style) so that it can be populated with
credentials
            val dr = DefaultRequest<Any>(serviceName)

            // copy request info
            dr.httpMethod = HttpMethodName.valueOf(request.method())
            with(request.url()) {
                dr.resourcePath = uri().path
                dr.endpoint = URI.create("${scheme()}://${host()}")

                // copy parameters
                for (p in queryParameterNames()) {
                    if (p != "") {
                        dr.addParameter(p, queryParameter(p))
                    }
                }
            }

            // copy headers
            for (h in request.headers().names()) {
                dr.addHeader(h, request.header(h))
            }

            // copy the request body
            val bodyBytes = request.body()?.let { body ->
                val buffer = Buffer()
                body.writeTo(buffer)
                IOUtils.toByteArray(buffer.inputStream())
            }

            dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

            return dr
        }

        fun toSignedOkHttpRequest(
            awsRequest: DefaultRequest<Any>,
            originalRequest: Request
```

```
    ): Request {
        // copy signed request back into an OkHttp Request
        val builder = Request.Builder()

        // copy headers from the signed request
        for ((k, v) in awsRequest.headers) {
            builder.addHeader(k, v)
        }

        // start building an HttpUrl
        val urlBuilder = HttpUrl.Builder()
            .host(awsRequest.endpoint.host)
            .scheme(awsRequest.endpoint.scheme)
            .encodedPath(awsRequest.resourcePath)

        // copy parameters from the signed request
        for ((k, v) in awsRequest.parameters) {
            urlBuilder.addQueryParameter(k, v)
        }

        return builder.url(urlBuilder.build())
            .method(originalRequest.method(), originalRequest.body())
            .build()
    }
}
```

Erstellen der Anwendung: Hauptaktivität

Die Hauptaktivität ist für die Initialisierung der Ansichten verantwortlich, die Benutzern angezeigt werden. Dazu gehören:

- Instanzieren eines Amazon Cognito-CredentialsProvider.
- Registrieren des Signature Version 4-Interceptors.
- Konfigurieren der Karte, indem sie auf einen Deskriptor im Kartenstil verweist und die entsprechende Zuordnung anzeigt.

MainActivity ist auch für die Weiterleitung von Lebenszyklusereignissen an die Kartenansicht verantwortlich, sodass das aktive Ansichtsfenster zwischen Aufrufen beibehalten werden kann.

```
package aws.location.demo.maplibre
```

```
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapbox.mapboxsdk.Mapbox
import com.mapbox.mapboxsdk.maps.MapView
import com.mapbox.mapboxsdk.maps.Style
import com.mapbox.mapboxsdk.module.http.HttpRequestUtil
import okhttp3.OkHttpClient

private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity() {
    private var mapView: MapView? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // configuration
        val identityPoolId = getString(R.string.identityPoolId)
        val region = getString(R.string.awsRegion)
        val mapName = getString(R.string.mapName)

        // Credential initialization
        val credentialProvider = CognitoCachingCredentialsProvider(
            applicationContext,
            identityPoolId,
            Regions.fromName(identityPoolId.split(":").first())
        )

        // initialize MapLibre
        Mapbox.getInstance(this, null)
        HttpRequestUtil.setOkHttpClient(
            OkHttpClient.Builder()
                .addInterceptor(SigV4Interceptor(credentialProvider, SERVICE_NAME))
                .build()
        )

        // initialize the view
        setContentView(R.layout.activity_main)
    }
}
```

```
// initialize the map view
mapView = findViewById(R.id.mapView)
mapView?.onCreate(savedInstanceState)
mapView?.getMapAsync { map ->
    map.setStyle(
        Style.Builder()
            .fromUri("https://maps.geo.${region}.amazonaws.com/maps/v0/maps/
${mapName}/style-descriptor")
    ) { style ->
        findViewById<TextView>(R.id.attributionView).text =
style.sources.first()?.attribution
    }
}

override fun onStart() {
    super.onStart()
    mapView?.onStart()
}

override fun onResume() {
    super.onResume()
    mapView?.onResume()
}

override fun onPause() {
    super.onPause()
    mapView?.onPause()
}

override fun onStop() {
    super.onStop()
    mapView?.onStop()
}

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    mapView?.onSaveInstanceState(outState)
}

override fun onLowMemory() {
    super.onLowMemory()
    mapView?.onLowMemory()
}
```



```
override fun onDestroy() {  
    super.onDestroy()  
    mapView?.onDestroy()  
}  
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden des MapLibre nativen SDK für iOS mit Amazon Location Service

Verwenden Sie [MapLibre das native SDK für iOS](#), um clientseitige Karten in iOS-Anwendungen einzubetten.

Das MapLibre native SDK für iOS ist eine Bibliothek, die auf [Mapbox microSD Native](#) basiert und mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre das native SDK für iOS integrieren, um interaktive Kartenansichten mit skalierbaren, anpassbaren Vektorkarten in Ihre iOS-Anwendungen einzubetten.

In diesem Tutorial wird beschrieben, wie Sie das MapLibre native SDK für iOS in Amazon Location integrieren. Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Xcode-Projekt aus der -App-Vorlage.
2. Wählen Sie SwiftUI als Schnittstelle aus.
3. Wählen Sie die SwiftUI-Anwendung für ihren Lebenszyklus aus.
4. Wählen Sie Swift als Sprache aus.

Hinzufügen MapLibre von Abhängigkeiten mit Swift Packages

So fügen Sie Ihrem Xcode-Projekt eine Paketabhängigkeit hinzu:

1. Navigieren Sie zu Datei > Pakete austauschen > Paketabhängigkeit hinzufügen.
2. Geben Sie die Repository-URL ein: **<https://github.com/maplibre/maplibre-gl-native-distribution>**

Note

Weitere Informationen zu Swift-Paketen finden Sie unter [Hinzufügen von Paketabhängigkeiten zu Ihrer App](#) unter Apple.com

3. Installieren Sie in Ihrem Terminal CocoaPods:

```
sudo gem install cocoapods
```

4. Navigieren Sie zum Projektverzeichnis Ihrer Anwendung und initialisieren Sie die Podfile mit dem CocoaPods Paketmanager:

```
pod init
```

5. Öffnen Sie die Podfile-Datei, die AWSCore Sie als Abhängigkeit hinzufügen möchten:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
end
```

6. Abhängigkeiten herunterladen und installieren:

```
pod install --repo-update
```

7. Öffnen Sie den Xcode-Workspace, den CocoaPods erstellt hat:

```
xed .
```

Erstellen der Anwendung: Konfiguration

Fügen Sie die folgenden Schlüssel und Werte zu Info.plist hinzu, um die Anwendung zu konfigurieren:

Schlüssel	Wert
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaaa5-0d97fb7ca3bd
MapName	ExampleMap

Erstellen der Anwendung: ContentView Layout

Um die Karte zu rendern, bearbeiten Sie `ContentView.swift`:

- Fügen Sie eine `hinzuMapView`, die die Zuordnung wiedergibt.
- Fügen Sie einen `hinzuTextField`, der die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

```
import SwiftUI

struct ContentView: View {
    @State private var attribution = ""

    var body: some View {
        MapView(attribution: $attribution)
            .centerCoordinate(.init(latitude: 49.2819, longitude: -123.1187))
            .zoomLevel(12)
            .edgesIgnoringSafeArea(.all)
            .overlay(
                TextField("", text: $attribution)
                    .disabled(true)
                    .font(.system(size: 12, weight: .light, design: .default))
                    .foregroundColor(.black)
                    .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
                alignment: .bottomTrailing)
    }
}
```

```
struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attributionsources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine neue Swift-Datei mit dem Namen `AWSSignatureV4Delegate.swift`, die die folgende Klassendefinition `AWSSignatureV4Delegate.swift` enthält, um AWS-Anfragen abzufangen und sie mit [Signature Version 4](#) zu signieren. Eine Instance dieser Klasse wird als Offline-Speicherdelegierter zugewiesen, der auch für das Umschreiben von URLs in der Kartenansicht verantwortlich ist.

```
import AWSCore
import Mapbox

class AWSSignatureV4Delegate : NSObject, MGLOfflineStorageDelegate {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
            identityPoolId: identityPoolId)
        super.init()
    }

    class func doubleEncode(path: String) -> String? {
```

```
        return path.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)?
            .addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
    }

    func offlineStorage(_ storage: MGLOfflineStorage, urlForResourceOf kind:
MGLResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return url
        }

        // URL-encode spaces, etc.
        let keyPath = String(url.path.dropFirst())
        guard let percentEncodedKeyPath =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
            print("Invalid characters in path '\(keyPath)'; unsafe to sign")
            return url
        }

        let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
        let requestHeaders: [String: String] = ["host": endpoint!.hostName]

        // sign the URL
        let task = AWSSignatureV4Signer
            .generateQueryStringForSignatureV4(
                withCredentialProvider: credentialsProvider,
                httpMethod: .GET,
                expireDuration: 60,
                endpoint: endpoint!,
                // workaround for https://github.com/aws-amplify/aws-sdk-ios/
issues/3215
                keyPath: AWSSignatureV4Delegate.doubleEncode(path:
percentEncodedKeyPath),
                requestHeaders: requestHeaders,
                requestParameters: .none,
                signBody: true)
        task.waitUntilFinished()

        if let error = task.error as NSError? {
            print("Error occurred: \(error)")
        }

        if let result = task.result {
```

```

        var urlComponents = URLComponents(url: (result as URL),
        resolvingAgainstBaseURL: false)!
        // re-use the original path; workaround for https://github.com/aws-amplify/
aws-sdk-ios/issues/3215
        urlComponents.path = url.path

        // have Mapbox GL fetch the signed URL
        return (urlComponents.url)!
    }

    // fall back to an unsigned URL
    return url
}
}

```

Erstellen der Anwendung: Kartenansicht

Die Kartenansicht ist für die Initialisierung einer Instance von `AWSSignatureV4Delegate` und die Konfiguration der zugrunde liegenden verantwortlich `MGLMapView`, die Ressourcen abrufen und die Karte rendert. Außerdem übernimmt es die Weitergabe von Attributionszeichenfolgen aus der Quelle des Stildesktors zurück an die `ContentView`.

Erstellen Sie eine neue Swift-Datei mit dem Namen `MapView.swift`, die die folgende `struct` Definition enthält:

```

import SwiftUI
import AWSCore
import Mapbox

struct MapView: UIViewRepresentable {
    @Binding var attribution: String

    private var mapView: MGLMapView
    private var signingDelegate: MGLOfflineStorageDelegate

    init(attribution: Binding<String>) {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String

        let region = (regionName as NSString).aws_regionTypeValue()

```

```
// MGLOfflineStorage doesn't take ownership, so this needs to be a member here
signingDelegate = AWSSignatureV4Delegate(region: region, identityPoolId:
identityPoolId)

// register a delegate that will handle SigV4 signing
MGLOfflineStorage.shared.delegate = signingDelegate

mapView = MGLMapView(
    frame: .zero,
    styleURL: URL(string: "https://maps.geo.\(regionName).amazonaws.com/maps/
v0/maps/\(mapName)/style-descriptor"))

    _attribution = attribution
}

func makeCoordinator() -> Coordinator {
    Coordinator($attribution)
}

class Coordinator: NSObject, MGLMapViewDelegate {
    var attribution: Binding<String>

    init(_ attribution: Binding<String>) {
        self.attribution = attribution
    }

    func mapView(_ mapView: MGLMapView, didFinishLoading style: MGLStyle) {
        let source = style.sources.first as? MGLVectorTileSource
        let attribution = source?.attributionInfos.first
        self.attribution.wrappedValue = attribution?.title.string ?? ""
    }
}

// MARK: - UIViewRepresentable protocol

func makeUIView(context: UIViewRepresentableContext<MapView>) -> MGLMapView {
    mapView.delegate = context.coordinator

    mapView.logoView.isHidden = true
    mapView.attributionButton.isHidden = true
    return mapView
}
```

```
func updateUIView(_ uiView: MGLMapView, context:
UIViewRepresentableContext<MapView>) {
}

// MARK: - MGLMapView proxy

func centerCoordinate(_ centerCoordinate: CLLocationCoordinate2D) -> MapView {
    mapView.centerCoordinate = centerCoordinate
    return self
}

func zoomLevel(_ zoomLevel: Double) -> MapView {
    mapView.zoomLevel = zoomLevel
    return self
}
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden der Amplify-Bibliothek mit Amazon Location Service

Das folgende Tutorial führt Sie durch die Verwendung von AWS Amplify mit Amazon Location. Amplify verwendet MapLibre microSD JS, um Karten in Ihrer JavaScript-basierten Anwendung zu rendern.

Amplify ist eine Reihe von Open-Source-Clientbibliotheken, die Schnittstellen für verschiedene Kategorien von -Services bereitstellen, einschließlich Amplify Geo, das von Amazon Location Service unterstützt wird. [Erfahren Sie mehr über die AWS Amplify Geo- JavaScript Bibliothek](#) .

Note

In diesem Tutorial wird davon ausgegangen, dass Sie die Schritte unter [Verwenden von Karten – So fügen Sie Ihrer Anwendung eine Zuordnung hinzu](#) bereits befolgt haben.

Erstellen der Anwendung: Gerüsterstellung

In diesem Tutorial wird eine Webanwendung erstellt, die verwendet, JavaScript um eine Karte auf einer HTML-Seite zu erstellen.

Erstellen Sie zunächst eine HTML-Seite (`index.html`), die den Container der Karte enthält:

- Geben Sie ein `-div`Element mit einem `id` von `einmap`, um die Dimensionen der Karte auf die Kartenansicht anzuwenden. Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body { margin: 0; }
      #map { height: 100vh; } /* 100% of viewport height */
    </style>
  </head>

  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie Ihrer Anwendung die folgenden Abhängigkeiten hinzu:

- AWS Amplify -Map- und Geobibliotheken.
- AWS Amplify -Core-Bibliothek.
- AWS Amplify -Authentifizierungsbibliothek.
- AWS Amplify Stylesheet.

```
<!-- CSS dependencies -->
  <link href="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-DrPVD9GufrixGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD" crossorigin="anonymous" referrerpolicy="no-referrer"></link>

<!-- JavaScript dependencies -->
  <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js" integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDLprcUXHnDDUcrNU" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js" integrity="sha384-70h+5w0l7XGyYvSqBki2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

```

<script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
integrity="sha384-jfkXCEfYyVmDXYKlgWNwv54xRaZgk14m7sjeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
integrity="sha384-TFMTyWuCbiptXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>
<script>
  // application-specific code
</script>

```

Dadurch wird eine leere Seite mit dem Container der Zuordnung erstellt.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit JavaScript:

1. Geben Sie die Kennungen des nicht authentifizierten Identitätspools ein, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten erstellt haben](#).

```

// Cognito Identity Pool ID
const identityPoolId = "region:identityPoolID"; // for example: us-
east-1:123example-1234-5678
// extract the Region from the Identity Pool ID
const region = identityPoolId.split(":")[0];

```

2. Konfigurieren Sie AWS Amplify für die Verwendung der von Ihnen erstellten Ressourcen, einschließlich des Identitätspools und der Map-Ressource (hier mit dem Standardnamen `dargestelltexplora.map`).

```

// Configure Amplify
const { Amplify } = aws_amplify_core;
const { createMap } = AmplifyMapLibre;

Amplify.configure({
  Auth: {
    identityPoolId,
    region,
  },
  geo: {

```

```
AmazonLocationService: {
  maps: {
    items: {
      "explore.map": {
        style: "Default style"
      },
    },
    default: "explore.map",
  },
  region,
},
});
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie können die anfängliche Kartenposition anpassen, zusätzliche Steuerelemente hinzufügen und Daten überlagern.

```
async function initializeMap() {
  const map = await createMap(
    {
      container: "map",
      center: [-123.1187, 49.2819],
      zoom: 10,
      hash: true,
    }
  );

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `sources.grabmaptiles.attribution` Schlüsseln

`sources.esri.attributionsources.here.attribution`, und enthalten. Amplify stellt automatisch eine Zuordnung bereit. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Ausführen der Anwendung

Sie können diese Beispielanwendung ausführen, indem Sie sie auf einem lokalen Webserver verwenden oder sie in einem Browser öffnen.

Um einen lokalen Webserver zu verwenden, können Sie `npx` verwenden, das als Teil von Node.js installiert ist, oder einen anderen Webserver Ihrer Wahl. Um `npx` zu verwenden, geben `npx serve` Sie aus demselben Verzeichnis wie ein `index.html`. Dies dient der Anwendung auf `localhost:5000`.

Note

Wenn die Richtlinie, die Sie für Ihre nicht authentifizierte Amazon Cognito-Rolle erstellt haben, eine `referrer` Bedingung enthält, werden Sie möglicherweise daran gehindert, mit `localhost`: URLs zu testen. In diesem Fall können Sie mit einem Webserver testen, der eine URL bereitstellt, die sich in Ihrer Richtlinie befindet.

Nach Abschluss des Tutorials sieht die endgültige Anwendung wie im folgenden Beispiel aus.

```
<html>
  <head>
    <!-- CSS dependencies -->
    <link href="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-DrPVD9GufrixGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD" crossorigin="anonymous" referrerpolicy="no-referrer"></link>

    <!-- JavaScript dependencies -->
    <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js" integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDLprcUXHnDDUcrNU" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js" integrity="sha384-70h+5w0l7XGyYvSqBKi2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

```

<script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
integrity="sha384-jfkXCEfYyVmDXyKlgWNwv54xRaZgk14m7sjeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
integrity="sha384-TFMTyWuCbiptXTzv0gzJbV8TPUUpG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

<style>
  body { margin: 0; }
  #map { height: 100vh; }
</style>
</head>

<body>
  <div id="map" />
  <script type="module">
    // Cognito Identity Pool ID
    const identityPoolId = "region:identityPoolId"; // for example: us-
east-1:123example-1234-5678
    // extract the Region from the Identity Pool ID
    const region = identityPoolId.split(":")[0];

    // Configure Amplify
    const { Amplify } = aws_amplify_core;
    const { createMap } = AmplifyMapLibre;

    Amplify.configure({
      Auth: {
        identityPoolId,
        region,
      },
      geo: {
        AmazonLocationService: {
          maps: {
            items: {
              "explore.map": {
                style: "Default style"
              },
            },
          },
          default: "explore.map",

```

```
        },
        region,
    },
}
});

async function initializeMap() {
    const map = await createMap(
        {
            container: "map",
            center: [-123.1187, 49.2819],
            zoom: 10,
            hash: true,
        }
    );

    map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
</script>
</body>
</html>
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte mit dem von Ihnen gewählten Kartenstil angezeigt. Dieses Beispiel wird auch auf der Registerkarte Karte einbetten einer beliebigen Seite mit Kartenressourcen in der [Amazon Location Service-Konsole](#) beschrieben.

Nachdem Sie dieses Tutorial abgeschlossen haben, gehen Sie zum Thema [Eine Karte anzeigen](#) in der -AWS AmplifyDokumentation, um mehr zu erfahren, einschließlich der Anzeige von Markierungen auf der Karte.

Verwenden von Tangram mit Amazon Location Service

Dieser Abschnitt enthält die folgenden Tutorials zur Integration von Tangram in Amazon Location.

Important

Die Tangram-Stile in den folgenden Tutorials sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Im Folgenden finden Sie ein Beispiel für einen AWS -CLI-Befehl zum Erstellen einer neuen Kartenressource mit dem Namen *TangramExampleMap* unter Verwendung des *-VectorHereContrast*Stils:

```
aws --region us-east-1 \  
  location \  
  create-map \  
  --map-name "TangramExampleMap" \  
  --configuration "Style=VectorHereContrast"
```

Note

Die Fakturierung wird durch Ihre -Nutzung bestimmt. Möglicherweise fallen Gebühren für die Nutzung anderer -AWSServices an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Themen

- [Verwenden von Tangram mit Amazon Location Service](#)
- [Verwenden von Tangram ES für Android mit Amazon Location Service](#)
- [Verwenden von Tangram ES für iOS mit Amazon Location Service](#)

Verwenden von Tangram mit Amazon Location Service

[Tangram](#) ist eine flexible Mapping-Engine, die für das Echtzeit-Rendering von 2D- und 3D-Karten aus Vektorkacheln entwickelt wurde. Es kann mit von Mapzen gestalteten Stilen und den HERE-Kacheln verwendet werden, die von der Amazon Location Service Maps API bereitgestellt werden. In diesem Handbuch wird beschrieben, wie Sie Tangram in Amazon Location in eine einfache HTML/JavaScript Anwendung integrieren, obwohl dieselben Bibliotheken und Techniken auch bei der Verwendung von Frameworks wie React und Angular gelten.

Tangram wurde auf [Leaflet](#) erstellt, einer Open-Source- JavaScript Bibliothek für mobile interaktive Karten. Das bedeutet, dass viele Leaflet-kompatible Plug-Ins und Kontrollen auch mit Tangram funktionieren.

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasenumbruch](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte
- [Cinnabar](#) – Ein klassisches Erscheinungsbild für allgemeine Mapping-Anwendungen
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und dem halben Toner-Stil von Stamen Design entspricht
- [Boln](#) – Eine Untersuchung von Skalierungstransformationen in der visuellen Sprache von TRON
- [Walkaround](#) – Ein auf Telefonie ausgerichteter Stil, der sich perfekt für Fußweg oder Fortfahren eignet

In diesem Handbuch wird beschrieben, wie Sie Tangram mit Amazon Location in einem einfachen HTML/einer einfachen JavaScript Anwendung integrieren, indem Sie den Tangram-Stil verwenden, der als „[Bulk Wrap](#)“ bezeichnet wird. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

Important

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Erstellen der Anwendung: Gerüsterstellung

Die Anwendung ist eine HTML-Seite mit JavaScript , um die Karte auf Ihrer Webanwendung zu erstellen. Erstellen Sie eine HTML-Seite (`index.html`) und erstellen Sie den Container der Zuordnung:

- Geben Sie ein `-divElement` mit einem `id` der Karte ein, um die Dimensionen der Karte auf die Kartenansicht anzuwenden.
- Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body {
```



```
    margin: 0;
  }

  #map {
    height: 100vh; /* 100% of viewport height */
  }
</style>
</head>
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie die folgenden Abhängigkeiten hinzu:

- Leaflet und das zugehörige CSS.
- Tangram.
- AWS SDK für JavaScript.

```
<!-- CSS dependencies -->
<link
  rel="stylesheet"
  href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/keqq/
sMZMZ19scR4PsZChSR7A=="
  crossorigin=""
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
<script src="https://unpkg.com/tangram"></script>
<script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
<script>
  // application-specific code
</script>
```

Dadurch wird eine leere Seite mit den erforderlichen Voraussetzungen erstellt. Der nächste Schritt führt Sie durch das Schreiben des JavaScript Codes für Ihre Anwendung.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Anmeldeinformationen:

1. Geben Sie die Namen und Kennungen Ihrer Ressourcen ein.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service map name; must be HERE-backed
const mapName = "TangramExampleMap";
```

2. Instanzieren Sie einen Anmeldeinformationsanbieter mithilfe des nicht authentifizierten Identitätspools, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten](#) erstellt haben. Da Anmeldeinformationen außerhalb des normalen AWS SDK-Workflows verwendet werden, laufen Sitzungen nach einer Stunde ab.

```
// extract the region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":", 1)[0];

// instantiate a Cognito-backed credential provider
const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: identityPoolId,
});
```

3. Mit Tangram können Sie zwar die URL(s) überschreiben, die zum Abrufen von Kacheln verwendet werden, aber es beinhaltet nicht die Möglichkeit, Anfragen abzufangen, damit sie signiert werden können.

Um dies zu umgehen, überschreiben Sie `sources.mapzen.url` um auf Amazon Location zu verweisen, indem Sie einen synthetischen Hostnamen verwenden `amazon.location`, der von einem [Service-Worker](#) bearbeitet wird. Im Folgenden finden Sie ein Beispiel für eine Szenenkonfiguration mit [dem Blasen-Wrap](#):

```
const scene = {
  import: [
    // Bubble Wrap style
    "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-usa.zip",
```

```
"https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-international.zip",
],
// override values beneath the `sources` key in the style above
sources: {
  mapzen: {
    // point at Amazon Location using a synthetic URL, which will be handled by
the service
    // worker
    url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
  },
// effectively disable raster tiles containing encoded normals
normals: {
  max_zoom: 0,
},
"normals-elevation": {
  max_zoom: 0,
},
},
};
```

Erstellen der Anwendung: Anforderungstransformation

Um den Service-Worker zu registrieren und zu initialisieren, erstellen Sie eine `registerServiceWorker` Funktion, die aufgerufen werden soll, bevor die Zuordnung initialisiert wird. Dadurch wird der JavaScript Code registriert, der in einer separaten Datei mit dem Namen `sw.js` als Service-Worker bereitgestellt wird, der `steuertindex.html`.

Anmeldeinformationen werden aus Amazon Cognito geladen und zusammen mit der Region an den Service-Worker übergeben, um Informationen zum Signieren von Kachelanforderungen mit [Signature Version 4](#) bereitzustellen.

```
/**
 * Register a service worker that will rewrite and sign requests using Signature
Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
```

```
    await credentials.refreshPromise();

    await reg.active.ready;

    if (navigator.serviceWorker.controller == null) {
      // trigger a navigate event to active the controller for this page
      window.location.reload();
    }

    // pass credentials to the service worker
    reg.active.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
      region: AWS.config.region,
    });
  } catch (error) {
    console.error("Service worker registration failed:", error);
  }
} else {
  console.warn("Service worker support is required for this example");
}
}
```

Die Service-Worker-Implementierung in `sw.js` lauscht auf `message` Ereignisse, um Konfigurationsänderungen an Anmeldeinformationen und Regionen aufzunehmen. Es fungiert auch als Proxy-Server, indem es auf `fetch` Ereignisse wartet. `fetch` Ereignisse, die auf den `amazon.location` synthetischen Hostnamen abzielen, werden neu geschrieben, um auf die entsprechende Amazon Location API abzielen und mit dem von Amplify Core signierten `Signer`.

```
// sw.js
self.importScripts(
  "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
```

```
// install immediately
event.waitUntil(self.skipWaiting());
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  } = event;

  if (newCredentials !== null) {
    credentials = newCredentials;
  }

  if (newRegion !== null) {
    region = newRegion;
  }
});

async function signedFetch(request) {
  const url = new URL(request.url);
  const path = url.pathname.slice(1).split("/");

  // update URL to point to Amazon Location
  url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
  url.host = `maps.geo.${region}.amazonaws.com`;
  // strip params (Tangram generates an empty api_key param)
  url.search = "";

  const signed = Signer.signUrl(url.toString(), {
    access_key: credentials.accessKeyId,
    secret_key: credentials.secretAccessKey,
    session_token: credentials.sessionToken,
  });

  return fetch(signed);
}

self.addEventListener("fetch", (event) => {
  const { request } = event;
```

```
// match the synthetic hostname we're telling Tangram to use
if (request.url.includes("amazon.location")) {
  return event.respondWith(signedFetch(request));
}

// fetch normally
return event.respondWith(fetch(request));
});
```

Verwenden Sie die folgende Funktion in `index.html`, um Anmeldeinformationen automatisch zu erneuern und sie vor Ablauf an den Service-Worker zu senden:

```
async function refreshCredentials() {
  await credentials.refreshPromise();

  if ("serviceWorker" in navigator) {
    const controller = navigator.serviceWorker.controller;

    controller.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
    });
  } else {
    console.warn("Service worker support is required for this example.");
  }

  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie haben die Möglichkeit, die anfängliche Kartenposition anzupassen, zusätzliche Steuerelemente hinzuzufügen und Daten zu überlagern.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsselnamen `sources.esri.attribution`, `sources.here.attribution`, und enthalten. Da Tangram diese Ressourcen nicht anfordert und nur mit Karten von HERE kompatibel ist, verwenden Sie „here 2020“. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

```
/**
 * Initialize a map.
 */
async function initializeMap() {
  // register the service worker to handle requests to https://amazon.location
  await registerServiceWorker();

  // Initialize the map
  const map = L.map("map").setView([49.2819, -123.1187], 10);
  Tangram.leafletLayer({
    scene,
  }).addTo(map);
  map.attributionControl.setPrefix("");
  map.attributionControl.addAttribution("© 2020 HERE");
}

initializeMap();
```

Ausführen der Anwendung

Um dieses Beispiel auszuführen, können Sie:

- Verwenden Sie einen Host, der HTTPS unterstützt,
- Verwenden Sie einen lokalen Webserver, um die Sicherheitseinschränkungen von Service-Workern einzuhalten.

Um einen lokalen Webserver zu verwenden, können Sie `npx` verwenden, da es als Teil von `Node.js` installiert ist. Sie können `npx serve` innerhalb desselben Verzeichnisses wie `index.html` und `verwendensw.js`. Dies dient der Anwendung auf localhost:5000.

Im Folgenden finden Sie die `index.html` Datei :

```
<!-- index.html -->
<html>
  <head>
    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
      integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/
keqq/sMzMZ19scR4PsZChSR7A=="
      crossorigin=""
    />
    <style>
      body {
        margin: 0;

        #map {
          height: 100vh;
        }
      </style>
    </head>

    <body>
      <div id="map" />
      <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
      <script src="https://unpkg.com/tangram"></script>
      <script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
      <script>
        // configuration
        // Cognito Identity Pool ID
        const identityPoolId = "<Identity Pool ID>";
        // Amazon Location Service Map name; must be HERE-backed
        const mapName = "<Map name>";

        AWS.config.region = identityPoolId.split(":")[0];

        // instantiate a credential provider
        credentials = new AWS.CognitoIdentityCredentials({
```



```
    IdentityPoolId: identityPoolId,
  });

const scene = {
  import: [
    // Bubble Wrap style
    "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-
shields-usa.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-
shields-international.zip",
  ],
  // override values beneath the `sources` key in the style above
  sources: {
    mapzen: {
      // point at Amazon Location using a synthetic URL, which will be handled by
the service
      // worker
      url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
    },
    // effectively disable raster tiles containing encoded normals
    normals: {
      max_zoom: 0,
    },
    "normals-elevation": {
      max_zoom: 0,
    },
  },
},
];

/**
 * Register a service worker that will rewrite and sign requests using Signature
Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;
    }
  }
}
```

```
    if (navigator.serviceWorker.controller == null) {
      // trigger a navigate event to active the controller for this page
      window.location.reload();
    }

    // pass credentials to the service worker
    reg.active.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
      region: AWS.config.region,
    });
  } catch (error) {
    console.error("Service worker registration failed:", error);
  }
} else {
  console.warn("Service Worker support is required for this example");
}
}

/**
 * Initialize a map.
 */
async function initializeMap() {
  // register the service worker to handle requests to https://amazon.location
  await registerServiceWorker();

  // Initialize the map
  const map = L.map("map").setView([49.2819, -123.1187], 10);
  Tangram.leafletLayer({
    scene,
  }).addTo(map);
  map.attributionControl.setPrefix("");
  map.attributionControl.addAttribution("© 2020 HERE");
}

initializeMap();
</script>
</body>
</html>
```

Im Folgenden finden Sie die `-sw.js` Datei:

```
// sw.js
self.importScripts(
  "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
  // install immediately
  event.waitUntil(self.skipWaiting());
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  } = event;

  if (newCredentials !== null) {
    credentials = newCredentials;
  }

  if (newRegion !== null) {
    region = newRegion;
  }
});

async function signedFetch(request) {
  const url = new URL(request.url);
  const path = url.pathname.slice(1).split("/");

  // update URL to point to Amazon Location
  url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
  url.host = `maps.geo.${region}.amazonaws.com`;
  // strip params (Tangram generates an empty api_key param)
```

```
url.search = "";

const signed = Signer.signUrl(url.toString(), {
  access_key: credentials.accessKeyId,
  secret_key: credentials.secretAccessKey,
  session_token: credentials.sessionToken,
});

return fetch(signed);
}

self.addEventListener("fetch", (event) => {
  const { request } = event;

  // match the synthetic hostname we're telling Tangram to use
  if (request.url.includes("amazon.location")) {
    return event.respondWith(signedFetch(request));
  }

  // fetch normally
  return event.respondWith(fetch(request));
});
```

Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden von Tangram ES für Android mit Amazon Location Service

[Tangram ES](#) ist eine C++-Bibliothek zum Rendern von 2D- und 3D-Zuweisungen aus Vektordaten mit OpenGL ES. Es ist das native Gegenstück zu [Tangram](#).

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasen-Wrap](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte.
- [Cinnabar](#) – Ein klassischer Look und Go-To für allgemeine Mapping-Anwendungen.
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und vom halben Toner-Stil von Stamen Design passt.
- [Boln](#) – Eine Erkundung von Skalierungstransformationen in der visuellen Sprache von TRON.
- [Walkthrough](#) – Ein auf den Fuß gelegter Stil, der sich perfekt für Walk oder Fortfahren eignet.

In diesem Handbuch wird beschrieben, wie Sie Tangram ES für Android mit Amazon Location integrieren, indem Sie den Tangram-Stil namens Cinnabar verwenden. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

Important

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Android Studio-Projekt aus der Vorlage Leere Aktivität.
2. Stellen Sie sicher, dass Kotlin für die Projektsprache ausgewählt ist.
3. Wählen Sie ein Mindest-SDK von API 16 aus: Android 4.1 (Jelly Bean) oder höher.
4. Öffnen Sie Projektstruktur, um Datei , Projektstruktur... auszuwählen, und wählen Sie den Abschnitt Abhängigkeiten aus.
5. Wenn <Alle Module> ausgewählt ist, wählen Sie die Schaltfläche +, um eine neue Bibliotheksabhängigkeit hinzuzufügen.
6. Fügen Sie AWS Android SDK Version 2.19.1 oder höher hinzu. Beispiel:
`com.amazonaws:aws-android-sdk-core:2.19.1`
7. Fügen Sie die Tangram-Version 0.13.0 oder höher hinzu. Zum Beispiel:
`com.mapzen.tangram:tangram:0.13.0`.

Note

Suche nach Tangram : `com.mapzen.tangram:tangram:0.13.0` generiert eine Meldung, dass sie „nicht gefunden“ ist, aber wenn Sie OK auswählen, kann sie hinzugefügt werden.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Ihrer AWS Region:

1. Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf `app/src/main/res/values/configuration.xml`.
2. Geben Sie die Namen und Kennungen Ihrer Ressourcen sowie die AWS Region ein, in der sie erstellt wurden:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
  <string name="mapName">TangramExampleMap</string>
  <string name="awsRegion">us-east-1</string>
  <string name="sceneUrl">https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip</string>
  <string name="attribution">© 2020 HERE</string>
</resources>
```

Erstellen der Anwendung: Aktivitätslayout

Bearbeiten von `app/src/main/res/layout/activity_main.xml`:

- Fügen Sie ein `hinzuMapView`, das die Zuordnung wiedergibt. Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.
- Fügen Sie ein `hinzuTextView`, das die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüssel `sources.esri.attribution` `sources.here.attribution`, und enthalten.

Da Tangram diese Ressourcen nicht anfordert und nur mit Karten von HERE kompatibel ist, verwenden Sie „here 2020“. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapzen.tangram.MapView
        android:id="@+id/map"
        android:layout_height="match_parent"
        android:layout_width="match_parent" />

    <TextView
        android:id="@+id/attributionView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#80808080"
        android:padding="5sp"
        android:textColor="@android:color/black"
        android:textSize="10sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="SmallSp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine Klasse mit dem Namen `SigV4Interceptor`, um AWS Anfragen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird bei dem HTTP-Client registriert, der zum Abrufen von Kartenressourcen verwendet wird, wenn die Hauptaktivität erstellt wird.

```
package aws.location.demo.okhttp

import com.amazonaws.DefaultRequest
import com.amazonaws.auth.AWS4Signer
```

```
import com.amazonaws.auth.AWSCredentialsProvider
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()

        if (originalRequest.url().host().contains("amazonaws.com")) {
            val signer = if (originalRequest.url().encodedPath().contains("@")) {
                // the presence of "@" indicates that it doesn't need to be double URL-
                encoded
                AWS4Signer(false)
            } else {
                AWS4Signer()
            }

            val awsRequest = toAWSRequest(originalRequest, serviceName)
            signer.setServiceName(serviceName)
            signer.sign(awsRequest, credentialsProvider.credentials)

            return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
        }

        return chain.proceed(originalRequest)
    }

    companion object {
        fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
            // clone the request (AWS-style) so that it can be populated with
            credentials
            val dr = DefaultRequest<Any>(serviceName)

            // copy request info
```



```
dr.httpMethod = HttpMethodName.valueOf(request.method())
with(request.url()) {
    dr.resourcePath = uri().path
    dr.endpoint = URI.create("${scheme()}://${host()}")

    // copy parameters
    for (p in queryParameterNames()) {
        if (p != "") {
            dr.addParameter(p, queryParameter(p))
        }
    }
}

// copy headers
for (h in request.headers().names()) {
    dr.addHeader(h, request.header(h))
}

// copy the request body
val bodyBytes = request.body()?.let { body ->
    val buffer = Buffer()
    body.writeTo(buffer)
    IOUtils.toByteArray(buffer.inputStream())
}

dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

return dr
}

fun toSignedOkHttpRequest(
    awsRequest: DefaultRequest<Any>,
    originalRequest: Request
): Request {
    // copy signed request back into an OkHttp Request
    val builder = Request.Builder()

    // copy headers from the signed request
    for ((k, v) in awsRequest.headers) {
        builder.addHeader(k, v)
    }

    // start building an HttpUrl
    val urlBuilder = HttpUrl.Builder()
```

```
        .host(awsRequest.endpoint.host)
        .scheme(awsRequest.endpoint.scheme)
        .encodedPath(awsRequest.resourcePath)

        // copy parameters from the signed request
        for ((k, v) in awsRequest.parameters) {
            urlBuilder.addQueryParameter(k, v)
        }

        return builder.url(urlBuilder.build())
            .method(originalRequest.method(), originalRequest.body())
            .build()
    }
}
}
```

Erstellen der Anwendung: Hauptaktivität

Die Hauptaktivität ist für die Initialisierung der Ansichten verantwortlich, die Benutzern angezeigt werden. Dazu gehören:

- Instanzieren eines Amazon Cognito-CredentialsProvider.
- Registrieren des Signature Version 4-Interceptors.
- Konfigurieren der Karte, indem sie auf einen Kartenstil verweist, Kachel-URLs überschreibt und die entsprechende Zuordnung anzeigt.

MainActivity ist auch für die Weiterleitung von Lebenszyklusereignissen an die Kartenansicht verantwortlich.

```
package aws.location.demo.tangram

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapzen.tangram.*
import com.mapzen.tangram.networking.DefaultHttpHandler
import com.mapzen.tangram.networking.HttpHandler
```

```
private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity(), MapView.MapReadyCallback {
    private var mapView: MapView? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)

        mapView = findViewById(R.id.map)

        mapView?.getMapAsync(this, getHttpHandler())
        findViewById<TextView>(R.id.attributionView).text =
            getString(R.string.attribution)
    }

    override fun onMapReady(mapController: MapController?) {
        val sceneUpdates = arrayListOf(
            SceneUpdate(
                "sources.mapzen.url",
                "https://maps.geo.${getString(R.string.awsRegion)}.amazonaws.com/maps/
v0/maps/${
                    getString(
                        R.string.mapName
                    )
                }/tiles/{z}/{x}/{y}"
            )
        )

        mapController?.let { map ->
            map.updateCameraPosition(
                CameraUpdateFactory.newLngLatZoom(
                    LngLat(-123.1187, 49.2819),
                    12F
                )
            )
            map.loadSceneFileAsync(
                getString(R.string.sceneUrl),
                sceneUpdates
            )
        }
    }
}
```

```
private fun getHandler(): Handler {
    val builder = DefaultHandler.getClientBuilder()

    val credentialsProvider = CognitoCachingCredentialsProvider(
        applicationContext,
        getString(R.string.identityPoolId),
        Regions.US_EAST_1
    )

    return DefaultHandler(
        builder.addInterceptor(
            SigV4Interceptor(
                credentialsProvider,
                SERVICE_NAME
            )
        )
    )
}

override fun onResume() {
    super.onResume()
    mapView?.onResume()
}

override fun onPause() {
    super.onPause()
    mapView?.onPause()
}

override fun onLowMemory() {
    super.onLowMemory()
    mapView?.onLowMemory()
}

override fun onDestroy() {
    super.onDestroy()
    mapView?.onDestroy()
}
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden von Tangram ES für iOS mit Amazon Location Service

[Tangram ES](#) ist eine C++-Bibliothek zum Rendern von 2D- und 3D-Zuweisungen aus Vektordaten mit OpenGL ES. Es ist das native Gegenstück zu [Tangram](#).

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasen-Wrap](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte
- [Cinnabar](#) – Ein klassisches Erscheinungsbild für allgemeine Mapping-Anwendungen
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und dem halben Toner-Stil von Stamen Design entspricht
- [Boln](#) – Eine Untersuchung von Skalierungstransformationen in der visuellen Sprache von TRON
- [Walkaround](#) – Ein auf den Fußabdruck ausgerichteter Stil, der sich perfekt zum Erkunden oder Verlassen von eignet

In diesem Handbuch wird beschrieben, wie Sie Tangram ES für iOS mit Amazon Location integrieren, indem Sie den Tangram-Stil namens Cinnabar verwenden. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

Important

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Erstellen der Anwendung: Initialisierung

So initialisieren Sie die Anwendung:

1. Erstellen Sie ein neues Xcode-Projekt aus der -App-Vorlage.
2. Wählen Sie SwiftUI als Schnittstelle aus.
3. Wählen Sie SwiftUI-Anwendung für ihre Lebensdauer aus.
4. Wählen Sie Swift als Sprache aus.

Erstellen der Anwendung: Abhängigkeiten hinzufügen

Um Abhängigkeiten hinzuzufügen, können Sie einen Abhängigkeitsmanager verwenden, z. B. [CocoaPods](#):

1. Installieren Sie in Ihrem Terminal CocoaPods:

```
sudo gem install cocoapods
```

2. Navigieren Sie zum Projektverzeichnis Ihrer Anwendung und initialisieren Sie die Podfile mit dem CocoaPods Paketmanager:

```
pod init
```

3. Öffnen Sie das Podfile, um AWSCore und Tangram-es als Abhängigkeiten hinzuzufügen:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
  pod 'Tangram-es'  
end
```

4. Abhängigkeiten herunterladen und installieren:

```
pod install --repo-update
```

5. Öffnen Sie den Xcode-Workspace, den CocoaPods erstellt hat:

```
xed .
```

Erstellen der Anwendung: Konfiguration

Fügen Sie die folgenden Schlüssel und Werte zu Info.plist hinzu, um die Anwendung zu konfigurieren und Telemetrie zu deaktivieren:

Schlüssel	Wert
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaaa5-0d97fb7ca3bd
MapName	ExampleMap
SceneURL	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

Erstellen der Anwendung: ContentView Layout

Um die Karte zu rendern, bearbeiten Sie `ContentView.swift`:

- Fügen Sie eine `hinzuMapView`, die die Zuordnung wiedergibt.
- Fügen Sie einen `hinzuTextField`, der die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attribution` `sources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Nutzungsbedingungen für -Services](#).

```
import SwiftUI
import TangramMap

struct ContentView: View {
    var body: some View {
        MapView()
    }
}
```

```

        .cameraPosition(TGCameraPosition(
            center: CLLocationCoordinate2DMake(49.2819, -123.1187),
            zoom: 10,
            bearing: 0,
            pitch: 0))
        .edgesIgnoringSafeArea(.all)
        .overlay(
            Text("© 2020 HERE")
                .disabled(true)
                .font(.system(size: 12, weight: .light, design: .default))
                .foregroundColor(.black)
                .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
            alignment: .bottomTrailing)
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

```

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine neue Swift-Datei mit dem Namen `AWSSignatureV4URLHandler.swift`, die die folgende Klassendefinition `AWSSignatureV4URLHandler` enthält, um AWS Anforderungen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird als URL-Handler innerhalb des Tangram `registriertMapView`.

```

import AWSCore
import TangramMap

class AWSSignatureV4URLHandler: TGDefaultURLHandler {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
    }
}

```



```
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
        super.init()
    }

    override func downloadRequestAsync(_ url: URL, completionHandler: @escaping
TGDownloadCompletionHandler) -> UInt {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return super.downloadRequestAsync(url, completionHandler:
completionHandler)
        }

        // URL-encode spaces, etc.
        let keyPath = String(url.path.dropFirst())
        guard let keyPathSafe =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
            print("Invalid characters in path '\(keyPath)'; unsafe to sign")
            return super.downloadRequestAsync(url, completionHandler:
completionHandler)
        }

        // sign the URL
        let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
        let requestHeaders: [String: String] = ["host": endpoint!.hostName]
        let task = AWSSignatureV4Signer
            .generateQueryStringForSignatureV4(
                withCredentialProvider: credentialsProvider,
                httpMethod: .GET,
                expireDuration: 60,
                endpoint: endpoint!,
                keyPath: keyPathSafe,
                requestHeaders: requestHeaders,
                requestParameters: .none,
                signBody: true)
        task.waitUntilFinished()

        if let error = task.error as NSError? {
            print("Error occurred: \(error)")
        }

        if let result = task.result {
            // have Tangram fetch the signed URL

```

```

        return super.downloadRequestAsync(result as URL, completionHandler:
completionHandler)
    }

    // fall back to an unsigned URL
    return super.downloadRequestAsync(url, completionHandler: completionHandler)
}
}

```

Erstellen der Anwendung: Kartenansicht

Die Kartenansicht ist für die Initialisierung einer Instance von `AWSSignatureV4Delegate` und die Konfiguration der zugrunde liegenden `verantwortlichMGLMapView`, die Ressourcen abrufen und die Karte rendert. Außerdem übernimmt es die Weitergabe von Attributionszeichenfolgen aus der Quelle des Stildeskriptors zurück an die `ContentView`.

Erstellen Sie eine neue Swift-Datei mit dem Namen `MapView.swift`, die die folgende `struct` Definition enthält:

```

import AWSCore
import TangramMap
import SwiftUI

struct MapView: UIViewRepresentable {
    private let mapView: TGMapView

    init() {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String
        let sceneURL = URL(string: Bundle.main.object(forKey: "SceneURL")
as! String)!

        let region = (regionName as NSString).aws_regionTypeValue()

        // rewrite tile URLs to point at AWS resources
        let sceneUpdates = [
            TGSceneUpdate(path: "sources.mapzen.url",
                value: "https://maps.geo.\(regionName).amazonaws.com/maps/v0/
maps/\(mapName)/tiles/{z}/{x}/{y}")
        ]
    }
}

```

```
// instantiate a TGURLHandler that will sign AWS requests
let urlHandler = AWSSignatureV4URLHandler(region: region, identityPoolId:
identityPoolId)

// instantiate the map view and attach the URL handler
mapView = TGMapView(frame: .zero, urlHandler: urlHandler)

// load the map style and apply scene updates (properties modified at runtime)
mapView.loadScene(from: sceneURL, with: sceneUpdates)
}

func cameraPosition(_ cameraPosition: TGCameraPosition) -> MapView {
    mapView.cameraPosition = cameraPosition

    return self
}

// MARK: - UIViewRepresentable protocol

func makeUIView(context: Context) -> TGMapView {
    return mapView
}

func updateUIView(_ uiView: TGMapView, context: Context) {
}
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Zeichnen von Datenfunktionen auf einer Karte

Nachdem Sie über eine Anwendung verfügen, die eine Karte mithilfe von Amplify oder Tangram rendert MapLibre, besteht ein natürlicher nächster Schritt darin, Features über die Karte zu zeichnen. Möglicherweise möchten Sie Ihre Kundenstandorte als Markierungen auf der Karte rendern.

Im Allgemeinen können Sie die [Ortssuchfunktionen](#) verwenden, um Standorte aus Ihren Daten zu finden, und dann die Funktionalität von Amplify oder Tangram verwenden MapLibre, um die Standorte zu rendern.

Beispiele für das Rendern verschiedener Objekttypen auf der Karte finden Sie in den folgenden MapLibre Beispielen:

- [Beispiel: Markierungen zeichnen](#)
- [Beispiel: Zeichnet geclusterte Punkte](#)
- [Beispiel: Zeichne ein Polygon](#)

Weitere Beispiele und Tutorials finden Sie unter [Codebeispiele und Tutorials für die Arbeit mit Amazon Location Service](#).

Festlegen von Erweiterungen für eine Zuordnung mit MapLibre

Es gibt Situationen, in denen Sie nicht möchten, dass Ihre Benutzer die gesamte Welt schwenken oder zoomen können. Wenn Sie MapLibre das Kartensteuerelement verwenden, können Sie die Ausmaße des Kartensteuerelements mit der `maxBounds` Option einschränken und den Zoom mit den `maxZoom` Optionen `minZoom` und einschränken.

Das folgende Codebeispiel zeigt, wie das Kartensteuerelement initialisiert wird, um das Panning auf eine bestimmte Grenze zu beschränken (in diesem Fall die Ausmaße der Grab-Datenquelle).

Note

Diese Beispiele befinden sich in JavaScript und funktionieren im Kontext des [Erstellen einer Webanwendung](#) Tutorials.

```
// Set bounds to Grab data provider region
var bounds = [
  [90.0, -21.943045533438166], // Southwest coordinates
  [146.25, 31.952162238024968] // Northeast coordinates
];

var mglMap = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxBounds: bounds // Sets bounds as max
    transformRequest,
  }
);
```

Ebenso können Sie eine minimale und maximale Zoomstufe für die Karte festlegen. Die Werte für beide Werte können zwischen 0 und 24 liegen, obwohl die Standardwerte für den minimalen Zoom 0 und für den maximalen Zoom 22 sind (Datenanbieter stellen möglicherweise nicht auf allen Zoomstufen Daten bereit. Die meisten Kartenbibliotheken behandeln dies automatisch). Im folgenden Beispiel werden die `maxZoom` Optionen `minZoom` und für das `Map` `MapLibre` -Steuerelement initialisiert.

```
// Set the minimum and maximum zoom levels
var mglMap = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxZoom: 12,
    minZoom: 5,
    transformRequest,
  }
);
```

Tip

Die `MapLibre` Kartensteuerung ermöglicht auch das Festlegen dieser Optionen zur Laufzeit und nicht während der Initialisierung mit den `set . . .` Funktionen `get . . .` und `. Verwenden Sie beispielsweise getMaxBounds und , setMaxBounds um die Kartengrenzen zur Laufzeit zu ändern.`

Verwalten Ihrer Kartenressourcen

Sie können Ihre Kartenressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten von Kartenressourcen

Sie können eine Liste Ihrer Kartenressourcen über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen.

Console

So zeigen Sie eine Liste der vorhandenen Kartenressourcen mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Maps aus.
3. Zeigen Sie eine Liste Ihrer Kartenressourcen unter Meine Karten an.

API

Verwenden Sie die [ListMaps](#) Operation aus den Amazon Location Maps APIs.

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Kartenressourcen im AWS Konto.

```
POST /maps/v0/list-maps
```

Im Folgenden finden Sie eine Beispielantwort für [ListMaps](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-30T01:38:36Z,
      "DataSource": "Esri",
      "Description": "string",
      "MapName": "ExampleMap",
      "UpdateTime": 2020-10-30T01:38:36Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-map](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Kartenressourcen im AWS Konto abzurufen.

```
aws location list-maps
```

Abrufen von Kartenressourcendetails

Sie können Details zu jeder Kartenressource in Ihrem AWS-Konto über die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs abrufen.

Console

So zeigen Sie die Details einer Kartenressource mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich **Maps** aus.
3. Wählen Sie unter **Meine Karten** den Namenslink der Zielkartenressource aus.

API

Verwenden Sie die [-DescribeMap](#) Operation aus den Amazon Location Maps APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Kartenressourcendetails für *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeMap](#):

```
{
  "Configuration": {
    "Style": "VectorEsriNavigation"
  },
  "CreateTime": 2020-10-30T01:38:36Z,
  "DataSource": "Esri",
  "Description": "string",
  "MapArn": "arn:aws:geo:us-west-2:123456789012:maps/ExampleMap",
  "MapName": "ExampleMap",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-10-30T01:40:36Z
}
```

CLI

Verwenden Sie den [describe-map](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Kartenressourcendetails für *ExampleMap* abzurufen.

```
aws location describe-map \  
  --map-name "ExampleMap"
```

Löschen einer Kartenressource

Sie können eine Kartenressource aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

Console

So löschen Sie eine vorhandene Kartenressource mithilfe der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich **Maps** aus.
3. Wählen Sie unter **Liste Meine Karten** die Zielkarte aus der Liste aus.
4. Wählen Sie **Zuordnung löschen** aus.

API

Verwenden Sie die [DeleteMap](#) Operation aus den Amazon Location Maps APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Kartenressource *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [DeleteMap](#):


```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-map](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Kartenressource *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

Suchen von Orts- und Geolocation-Daten mit Amazon Location

Amazon Location bietet die Möglichkeit, die Geolocation-Daten des von Ihnen ausgewählten Anbieters zu durchsuchen oder zu platzieren. Es gibt verschiedene Arten von Suchen.

- **Geokodierung** – Geokodierung ist der Prozess der Suche nach Adressen, Regionen, Geschäftsnamen oder anderen Interessenpunkten, basierend auf Texteingabe. Es gibt Details und die Position (in Breiten- und Längengrad) der gefundenen Ergebnisse zurück.
- **Umgekehrte Geokodierung** – Mit der umgekehrten Geokodierung können Sie Stellen in der Nähe eines bestimmten Standorts finden.
- **Autocomplete** – Autocomplete ist der Prozess, bei dem automatische Vorschläge gemacht werden, wie die Benutzer in einer Abfrage eingeben. Wenn sie beispielsweise **Par** einen Vorschlag eingeben, könnte er sein **Paris, France**.

Mit Amazon Location können Sie einen Datenanbieter für Ortssuchvorgänge auswählen, indem Sie eine Ortsindexressource erstellen und konfigurieren.

Sobald Sie Ihre Ressource erstellt haben, können Sie Anfragen mit dem AWS SDK für Ihre bevorzugte Sprache, Amplify oder die REST-API-Endpunkte senden. Sie können Daten aus der Antwort verwenden, um Standorte auf einer Karte zu markieren, Positionsdaten anzureichern und Positionen in für Menschen lesbaren Text umzuwandeln.

Note

Eine Übersicht über die Konzepte für die Suche nach Orten finden Sie unter [Suche nach Orten](#).

Themen

- [Voraussetzungen](#)
- [Geokodierung](#)
- [Geokodierung umkehren](#)
- [AutoVervollständigen](#)
- [Verwenden von Orts-IDs](#)
- [Kategorien und Filterergebnisse platzieren](#)
- [Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service](#)
- [Verwalten Ihrer Ortsindexressourcen](#)

Voraussetzungen

Bevor Sie mit der Geokodierung, der umgekehrten Geokodierung oder der Suche nach Orten beginnen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Ortsindexressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Ortsindexressource

Erstellen Sie zunächst eine Ortsindex-Ressource in Ihrem AWS Konto.

Wenn Sie eine Ortsindexressource erstellen, können Sie aus den Datenanbietern wählen, die zur Unterstützung von Abfragen für Geokodierung, Reverse-Geokodierung und Suchen verfügbar sind:

1. Esri – Weitere Informationen zur Abdeckung von Esri in Ihrer relevanten Region finden Sie unter [Abdeckung der Esri-Geocodierung](#) in der Esri-Dokumentation.
2. HERE Technologies – Weitere Informationen zur Abdeckung von HERE in Ihrer relevanten Region finden Sie unter [Abdeckung der HERE-Geokodierung](#) in der HERE-Dokumentation.
3. Grab – Grab stellt Daten nur für Südostasien bereit. Weitere Informationen zur Abdeckung von Grab finden Sie unter [Länder/Regionen und abgedecktes Gebiet](#) in diesem Leitfaden.

Sie können dies über die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs tun.

Console

So erstellen Sie eine Ortsindexressource mit der Amazon Location Service-Konsole

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie Ortsindex erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Ortsindex-Ressource ein. Zum Beispiel *ExamplePlaceIndex*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie unter Datenanbieter einen verfügbaren [Datenanbieter](#) aus, der mit Ihrer Ortsindexressource verwendet werden soll.

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS -Servicebedingungen](#).

6. Geben Sie unter Datenspeicheroptionen an, ob Sie Suchergebnisse aus Ihrer Ortsindexressource speichern möchten.
7. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag für Ihre neue Ortsindexressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
8. Wählen Sie Ortsindex erstellen aus.

API

So erstellen Sie eine Ortsindex-Ressource mithilfe der Amazon Location APIs

Verwenden Sie die [-CreatePlaceIndex](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Ortsindexressource mit dem Namen *ExamplePlaceIndex* unter Verwendung des Datenanbieters *Esri* .

```
POST /places/v0/indexes
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

So erstellen Sie eine Ortsindex-Ressource mit AWS CLI -Befehlen

Verwenden Sie den [create-place-index](#)-Befehl.

Im folgenden Beispiel wird eine Ortsindexressource namens *ExamplePlaceIndex* mit *Esri* als Datenanbieter erstellt.

```
aws location \
  create-place-index \
  --data-source "Esri" \
  --description "Example place index" \
  --index-name "ExamplePlaceIndex" \
  --tags Tag1=Value1
```

Note

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer AWS -Services an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Ortsindexressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, wählen Sie aus, wie Sie Ihre Anfragen authentifizieren würden:

- Informationen dazu, wie Sie auf die Services zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie eine Website mit anonymen Benutzern haben, können Sie API-Schlüssel oder Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines API-Schlüssels für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon Location [JavaScript Authentifizierungshelfer](#).

```
import { LocationClient, SearchPlaceIndexForTextCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  IndexName: "ExamplePlaceIndex",
  Text: "Anyplace",
  BiasPosition: [-123.4567, 45.6789]
};

const command = new SearchPlaceIndexForTextCommand(input);

const response = await client.send(command);
```

Geokodierung

Geokodierung ist ein Prozess, der Text, z. B. eine Adresse, eine Region, einen Geschäftsnamen oder einen Interessenpunkt, in eine Reihe geografischer Koordinaten umwandelt. Sie können Ortsindexressourcen verwenden, um Geocodierungsanforderungen zu senden und Daten, die aus der Geocodierung abgerufen wurden, zu integrieren, um Daten auf einer Karte für Ihre Web- oder mobile Anwendung anzuzeigen.

Dieser Abschnitt führt Sie durch das Senden einer einfachen Geocodierungsanforderung und das Senden von Geocodierungsanfragen mit optionalen Spezifikationen.

Geokodierung

Sie können eine einfache Anfrage an Geocode senden, indem Sie die [-SearchPlaceIndexForText](#) Operation verwenden, um eine Adresse in eine Reihe von Koordinaten zu konvertieren. Eine einfache -Anforderung enthält den folgenden erforderlichen Parameter:

- **Text** – Eine Adresse, ein Name, eine Stadt oder eine Region, die in eine Reihe von Koordinaten konvertiert werden soll. Zum Beispiel die Zeichenfolge Any Town.

Verwenden Sie den folgenden optionalen Parameter, um eine maximale Anzahl von Ergebnissen pro Seite anzugeben:

- **MaxResults** – Beschränkt die maximale Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, die Ortsindex-Ressource *ExamplePlaceIndex* nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt zu* durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
```

```
"Text": "Any Town",
"MaxResults": 10
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-text](#) Befehl, um die Ortsindex-Ressource *ExamplePlaceIndex* nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt zu* durchsuchen.

```
aws location \
  search-place-index-for-text \
    --index-name ExamplePlaceIndex \
    --text "Any Town" \
    --max-results 10
```

Geocode nahe einer Position

Bei der Geokodierung können Sie eine bestimmte Position mit dem folgenden optionalen Parameter geocodieren:

- *BiasPosition* – Die Position, an der Sie in der Nähe suchen möchten. Dadurch wird Ihre Suche eingeschränkt, indem nach Ergebnissen gesucht wird, die der angegebenen Position am nächsten liegen. Definiert als [longitude, latitude]

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, um die Ortsindexressource nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Any Town* in der Nähe der Position [-123.4567,45,6789] zu durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "BiasPosition": [-123.4567,45.6789]
}
```

Geocode innerhalb eines Begrenzungsrahmens

Mit dem folgenden optionalen Parameter können Sie innerhalb eines Begrenzungsrahmens Geocodieren, um Ihre Ergebnisse auf Koordinaten innerhalb einer bestimmten Grenze einzugrenzen:

- `FilterBoundingBox` – Ein Begrenzungsrahmen, den Sie angeben, um Ihre Ergebnisse nach Koordinaten innerhalb der Grenzen des Rahmens zu filtern. Definiert als [`LongitudeSW`, `LatitudeSW`, `LongitudeNE`, `LatitudeNE`]

Note

Eine Anforderung darf nicht sowohl die `FilterBoundingBox` `BiasPosition` Parameter als auch enthalten. Die Angabe beider Parameter in der Anforderung gibt einen `ValidationException` Fehler zurück.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, innerhalb eines Begrenzungsrahmens nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt* zu suchen. Der Begrenzungsrahmen folgt:

- Der Längengrad der Südwest-Ecke beträgt *-124.1450*.
- Der Breitengrad der Westecke beträgt *41,7045*.
- Der Längengrad der nordöstlichen Ecke beträgt *-124.1387*.
- Der Breitengrad der nordöstlichen Ecke beträgt *41,7096*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterBoundingBox": [
    -124.1450,41.7045,
    -124.1387,41.7096
  ]
}
```


Geocode innerhalb eines Landes

Sie können mit dem folgenden optionalen Parameter innerhalb eines oder mehrerer Länder, die Sie angeben, geokodieren:

- **FilterCountries** – Das Land oder die Region, in dem bzw. der Sie Geocode erstellen möchten. Sie können bis zu 100 Länder in einer Anfrage mit einem dreistelligen [ISO 3166](#)-Ländercode definieren. Verwenden Sie beispielsweise AUS für Australien.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anfrage zur Suche nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt* in Deutschland und Frankreich.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "FilterCountries": ["DEU", "FRA"]
}
```

Filtern nach Kategorien

Sie können die Kategorien filtern, die in Ihrer Geocode-Anforderung zurückgegeben werden, indem Sie den folgenden optionalen Parameter verwenden:

- **FilterCategories** – Die Ergebniskategorien, die in Ihrer Abfrage zurückgegeben werden sollen. Sie können bis zu 5 Kategorien in einer Anforderung angeben. Die Liste der Amazon Location Service-Kategorien finden Sie im Abschnitt [Kategorien](#). Sie können beispielsweise angeben, `Hotel` um nur Rückgabe-Ensembles in Ihrer Abfrage anzugeben.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anfrage, in den USA nach einem Lebensmittelgeschäft namens *Home* zu suchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Hometown Coffee",
```

```
"FilterCategories": ["Coffee Shop"],  
"FilterCountries": ["USA"]  
}
```

Weitere Informationen zum Filtern nach Kategorien finden Sie unter [Kategorien und Filterergebnisse platzieren](#)

Geocode in einer bevorzugten Sprache

Sie können eine Sprachpräferenz für Ergebnisse Ihrer Suche festlegen, indem Sie den optionalen Language Parameter verwenden. Beispielsweise **100 Main St, Anytown, USA** kann eine Suche nach **100 Main St, Any Town, USA** standardmäßig zurückgeben. Wenn Sie jedoch für `as` als `auswählenLanguage`, können die Ergebnisse **100 Rue Principale, Any Town, États-Unis** stattdessen zurückgegeben werden.

- `Language` – Ein Sprachcode, der zum Rendern der Ergebnisse Ihrer Abfrage verwendet werden soll. Der Wert muss ein gültiger [BCP-47](#)-Sprachcode sein. Zum Beispiel `en` für Englisch.

Note

Wenn nicht angegeben `Language` ist oder die angegebene Sprache für ein Ergebnis nicht unterstützt wird, wird die Standardsprache des Partners für dieses Ergebnis verwendet.

Das folgende Beispiel ist eine `SearchPlaceIndexForText` Anforderung zur Suche nach einem Ort namens **Any Town** mit der bevorzugten Sprache, die als angegeben ist `de`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text  
Content-type: application/json  
{  
  "Text": "Any Town",  
  "Language": "de"  
}
```

Beispielantwort

Example

Im Folgenden finden Sie eine Beispielantwort, wenn Sie die [-SearchPlaceIndexForText](#) Operation über die Amazon Location Places APIs aufrufen. Die

Ergebnisse umfassen relevante [Stellen](#) und die [Anforderungszusammenfassung](#) . Basierend auf der Auswahl von Esri oder HERE als Partner werden zwei Antworten angezeigt.

Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Amazon",
  "MaxResults": 1,
  "FilterCountries": ["USA"],
  "BiasPosition": [-112.10, 46.32]
}
```

Example response (Esri)

```
{
  "Results": [
    {
      "Place": {
        "Country": "USA",
        "Geometry": {
          "Point": [
            -112.10667999999998,
            46.319090000000074
          ]
        },
        "Interpolated": false,
        "Label": "Amazon, MT, USA",
        "Municipality": "Amazon",
        "Region": "Montana",
        "SubRegion": "Jefferson County"
      },
      "Distance": 523.4619749879726,
      "Relevance": 1
    }
  ],
  "Summary": {
    "BiasPosition": [
      -112.1,
      46.32
    ]
  }
}
```

```
    "DataSource": "Esri",
    "FilterCountries": [
      "USA"
    ],
    "MaxResults": 1,
    "ResultBBox": [
      -112.10667999999998,
      46.319090000000074,
      -112.10667999999998,
      46.319090000000074
    ],
    "Text": "Amazon"
  }
}
```

Example response (HERE)

```
{
  "Summary": {
    "Text": "Amazon",
    "BiasPosition": [
      -112.1,
      46.32
    ],
    "FilterCountries": [
      "USA"
    ],
    "MaxResults": 1,
    "ResultBBox": [
      -112.10668,
      46.31909,
      -112.10668,
      46.31909
    ],
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "Amazon, Jefferson City, MT, United States",
        "Geometry": {
          "Point": [
            -112.10668,
```

```

        46.31909
      ],
      "Neighborhood": "Amazon",
      "Municipality": "Jefferson City",
      "SubRegion": "Jefferson",
      "Region": "Montana",
      "Country": "USA",
      "Interpolated": false,
      "TimeZone": {
        "Name": "America/Denver",
        "Offset": -25200
      }
    },
    "PlaceId": "AQAAAIADsn2T3KdrWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
    "Distance":
523.4619749905755
  }
]
}

```

Geokodierung umkehren

Reverse Geocoding ist ein Prozess, der eine Reihe von Koordinaten in aussagekräftigen Text umwandelt, z. B. eine Adresse, eine Region, einen Geschäftsnamen oder einen Interessenpunkt. Sie können Ortsindexressourcen verwenden, um Anforderungen zur Reverse-Geocodierung zu senden und Daten, die aus Reverse-Geocodierung abgerufen wurden, zu integrieren, um Daten auf einer Karte für Ihre Web- oder mobile Anwendung anzuzeigen.

Dieser Abschnitt führt Sie durch das Senden einer einfachen Reverse-Geocodierungsanforderung.

Geokodierung umkehren

Sie können eine einfache Anfrage einreichen, um eine Reihe von Koordinaten umzukehren und sie mithilfe der [SearchPlaceIndexForPosition](#) Operation in eine aussagekräftige Adresse, einen Interessenpunkt oder einen allgemeinen Ort ohne Adresse zu konvertieren. Eine einfache -Anforderung enthält den folgenden erforderlichen Parameter:

- **Position** – Eine Reihe von Koordinaten, die Sie in eine Adresse, einen Interessenpunkt oder eine allgemeine Position konvertieren möchten. Definiert im Format `[longitude, latitude]`.

Um eine maximale Anzahl von Ergebnissen pro Seite anzugeben, fügen Sie den folgenden optionalen Parameter hinzu:

- `MaxResults` – Beschränkt die maximale Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Wenn Sie eine bevorzugte Sprache für die Ergebnisse Ihrer Abfrage angeben möchten, verwenden Sie den folgenden optionalen Parameter:

- `Language` – Ein Sprachcode, der zum Rendern von Ergebnissen verwendet werden soll. Der Wert muss ein gültiger [BCP-47](#)-Sprachcode sein. Zum Beispiel `en` für Englisch.

Note

Wenn nicht angegeben `Language` ist oder die angegebene Sprache für ein Ergebnis nicht unterstützt wird, wird die Standardsprache des Partners für dieses Ergebnis verwendet.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForPosition](#) Anforderung, die Ortsindex-Ressource *ExamplePlaceIndex* nach einer aussagekräftigen Adresse, einem Interessenpunkt oder einem allgemeinen Standort in der Nähe der Position `[122.3394,47,6159]` zu durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 5,
  "Language": "de"
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-position](#) Befehl, um die Ortsindex-Ressource *ExamplePlaceIndex* nach einer aussagekräftigen Adresse, einem Interessenpunkt oder einer allgemeinen Position in der Nähe der Position *[122.3394,47,6159]* zu durchsuchen.

```
aws location \  
  search-place-index-for-position \  
    --index-name ExamplePlaceIndex \  
    --position -122.3394 47.6159 \  
    --max-results 5 \  
    --language de
```

Beispielantwort

Example

Im Folgenden finden Sie eine Beispielantwort beim Aufrufen der [-SearchPlaceIndexForPosition](#) Operation über die Amazon Location Places APIs . Die Ergebnisse geben relevante [Stellen](#) und die [Anforderungszusammenfassung](#) zurück. Basierend auf der Auswahl von Esri oder Here als Partner werden zwei Antworten angezeigt.

Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position  
Content-type: application/json  
  
{  
  "Position": [-122.3394,47.6159],  
  "MaxResults": 1  
}
```

Example response (Esri)

```
{  
  "Results": [  
    {  
      "Place": {  
        "AddressNumber": "2111",
```

```

        "Country": "USA",
        "Geometry": {
            "Point": [
                -122.33937999999995,
                47.615910000000004
            ]
        },
        "Interpolated": false,
        "Label": "The Spheres, 2111 7th Ave, Seattle, WA, 98121, USA",
        "Municipality": "Seattle",
        "Neighborhood": "Belltown",
        "PostalCode": "98121",
        "Region": "Washington",
        "SubRegion": "King County"
    },
    "Distance": 1.8685861313438727
}
],
"Summary": {
    "DataSource": "Esri",
    "MaxResults": 1,
    "Position": [
        -122.3394,
        47.6159
    ]
}
}
}

```

Example response (HERE)

```

{
  "Summary": {
    "Position": [
      -122.3394,
      47.6159
    ],
    "MaxResults": 1,
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "2111 7th Ave, Seattle, WA 98121-5114, United States",

```



```

    "Geometry": {
      "Point": [
        -122.33938,
        47.61591
      ]
    },
    "AddressNumber": "2111",
    "Street": "7th Ave",
    "Neighborhood": "Belltown",
    "Municipality": "Seattle",
    "SubRegion": "King",
    "Region": "Washington",
    "Country": "USA",
    "PostalCode": "98121-5114",
    "Interpolated": false,
    "TimeZone": {
      "Name": "America/Los_Angeles",
      "Offset": -28800
    }
  },
  "PlaceId": "AQAAlIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
  "Distance": 1.868586125090601
}
]
}

```

AutoVervollständigen

Die automatische Vervollständigung bietet Endbenutzern reaktives Feedback, während sie ihre Suchabfrage eingeben. Es enthält Vorschläge für Adressen und Interessenpunkte, die auf teilweise oder falsch geschriebenem Freiformtext basieren. Sie können Ortsindexressourcen verwenden, um Vorschläge zur automatischen Vervollständigung anzufordern und die resultierenden Vorschläge in Ihrer Anwendung anzuzeigen.

Amazon Location unterstützt keine Speicherung von Vorschlägen zur automatischen Vervollständigung. Ein Fehler wird zurückgegeben, wenn der für einen Autocomplete-Aufruf verwendete Ortsindex für die Verwendung mit gespeicherten Geocodes konfiguriert ist. Um gespeicherte Geocodes zu verwenden und Vorschläge abzufragen, erstellen und konfigurieren Sie mehrere Ortsindizes.

In diesem Abschnitt wird beschrieben, wie Sie eine automatische Vervollständigungsanforderung senden. Es beginnt mit der einfachsten Form der Anforderung und zeigt dann optionale Parameter an, mit denen Sie die Relevanz von Suchergebnissen zur automatischen Vervollständigung erhöhen können.

Verwenden der automatischen Vervollständigung

Sie können eine einfache Anfrage für Vorschläge zur automatischen Vervollständigung über die [SearchPlaceIndexForSuggestions](#) Operation einreichen. Die einfachste Form der Anforderung hat einen einzelnen erforderlichen Parameter, die Abfrage Text:

- **Text** – Der Freiform-Teilttext, der zum Generieren von Ortsvorschlägen verwendet werden soll. Zum Beispiel die Zeichenfolge `eiffel tow`.

Um die Anzahl der zurückgegebenen Ergebnisse zu begrenzen, fügen Sie den optionalen `MaxResults` Parameter hinzu:

- **MaxResults** – Beschränkt die Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Sie können die Amazon Location APIs oder die verwenden AWS CLI.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForSuggestions](#) Anforderung, die Ortsindex-Ressource nach bis zu **5** Vorschlägen zu durchsuchen *ExamplePlaceIndex*, die auf dem partiellen Ortsnamen .

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-suggestions](#) Befehl, um die Ortsindex-Ressource nach bis zu **5** Vorschlägen zu durchsuchen *ExamplePlaceIndex*, die auf dem partiellen Ortsnamen **.**

```
aws location \
    search-place-index-for-suggestions \
    --index-name ExamplePlaceIndex \
    --text kamp \
    --max-results 5
```

Der Aufruf von SearchPlaceIndexForSuggestions führt zu einer Liste von Orten mit jeweils einem Namen und einer ID. Sie können diese Ergebnisse verwenden, um Vorschläge darüber zu präsentieren, wonach der Benutzer während der Eingabe suchen könnte, z. B. eine Dropdownliste mit Auswahlmöglichkeiten unter einem Textfeld. Im Folgenden finden Sie beispielsweise die Ergebnisse für Vorschläge, die auf einem Benutzer basieren, der *eingibt*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhYmsYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
```

```

    "PlaceId":
      "AQAAAIAAzZfZt3qMruKG0byhP6MM0pqy2L8SUL1VWT7a3ertLBRS6Q5n7I4s9D7E0nRHADAJ7mL7kvX1Q8HD-
      mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
      dfB0080oqujKZldvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAIAAvbXXx-
      sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yYOLhZx4k
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
        "AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkBnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
    }
  ]
}

```

Im nächsten Abschnitt wird erläutert, wie Sie die PlaceID aus diesen Ergebnissen verwenden.

Verwenden der Ergebnisse der automatischen Vervollständigung

Der Aufruf von `SearchPlaceIndexForSuggestions` führt zu einer Liste von Orten mit jeweils einem Namen und einer ID. Sie können diese Ergebnisse verwenden, um Vorschläge darüber zu präsentieren, wonach der Benutzer während der Eingabe suchen könnte, z. B. eine Dropdownliste mit Auswahlmöglichkeiten unter einem Textfeld. Wenn der Benutzer eines der Ergebnisse auswählt, können Sie die [GetPlace](#) Operation mit der ID seiner Auswahl aufrufen, um die Details dieses Orts zurückzugeben, einschließlich Standort, Adresse oder anderer Details.

Note

Eine PlaceId ist nur gültig, wenn alle der folgenden Bedingungen in der ursprünglichen Suchanfrage und im Aufruf von `identisch sindGetPlace`.

- Kunde AWS-Konto
- AWS-Region
- Der in der Ortsindex-Ressource angegebene Datenanbieter

In der Regel verwenden Sie `GetPlace` mit den Amazon Location APIs . Das folgende Beispiel ist eine [GetPlace](#) Anforderung, einen der Vorschläge aus dem vorherigen Abschnitt zu finden. Dieses Beispiel basiert auf dem partiellen Ortsnamen .

```
POST /places/v0/indexes/ExamplePlaceIndex/
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdLJZAdgcT2oWi1w9pS4wXX0k301vsKLGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Automatische Vervollständigung nahe einer Position

Wenn Sie mithilfe von nach Vorschlägen zur automatischen Vervollständigung von Orten suchen [SearchPlaceIndexForSuggestions](#), können Sie lokal relevantere Vorschläge erhalten, indem Sie den folgenden optionalen Parameter hinzufügen:

- `BiasPosition` – Die Position, an der Sie in der Nähe suchen möchten. Definiert als `[longitude, latitude]`.

Im folgenden Beispiel wird eine [SearchPlaceIndexForSuggestions](#) Anforderung verwendet, um die Ortsindex-Ressource `ExamplePlaceIndex` nach Ortsvorschlägen zu durchsuchen, die mit dem partiellen *Abfrageblock* nahe der Position `[32.5827,0.3169]` übereinstimmen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json


{
  "Text": "kamp",
  "BiasPosition": [32.5827,0.3169]
}
```

Die für dasselbe zurückgegebenen Vorschläge Text können unterschiedlich sein, wenn ein anderes ausgewählt `BiasPosition` wird, z. B. `[-96.7977, 32.776]`.

Automatische Vervollständigung innerhalb eines Begrenzungsrahmens

Sie können Ihre Suche nach automatischer Vervollständigung einschränken, um nur Vorschläge für Stellen zu erhalten, die sich innerhalb einer bestimmten Grenze befinden, indem Sie den folgenden optionalen Parameter hinzufügen:

- **FilterBBox** – Ein Begrenzungsrahmen, den Sie angeben, um Ihre Ergebnisse nach Koordinaten innerhalb der Grenzen des Rahmens zu filtern. Definiert als [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

 Note

Eine Anforderung darf nicht sowohl die `FilterBBox` `BiasPosition` Parameter als auch enthalten. Die Angabe beider Parameter in der Anforderung gibt einen `ValidationException` Fehler zurück.

Im folgenden Beispiel wird eine [-SearchPlaceIndexForSuggestions](#) Anforderung verwendet, um die Ortsindex-Ressource *ExamplePlaceIndex* nach Ortsvorschlägen zu durchsuchen, die mit der partiellen Abfrage „“ übereinstimmen und die im Begrenzungsrahmen enthalten sind, wobei:

- Der Längengrad der Westecke des Begrenzungsrahmens beträgt *32,5020*.
- Der Breitengrad der Bolwesten Ecke des Begrenzungsrahmens beträgt *0,2678*.
- Der Längengrad der nordöstlichen Ecke des Begrenzungsrahmens beträgt *32,6129*.
- Der Breitengrad der nordöstlichen Ecke des Begrenzungsrahmens beträgt *0,3502*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
```

```
Content-type: application/json
```

```
{
  "Text": "kamp",
  "FilterBBox": [
    32.5020, 0.2678,
    32.6129, 0.3502
  ]
}
```

Die für dasselbe zurückgegebenen Vorschläge Text unterscheiden sich, wenn ein anderes ausgewählt `FilterBBox` wird, z. B. [-97.9651, 32.0640, -95.1196, 34.0436].

Automatische Vervollständigung innerhalb eines Landes

Sie können Ihre Suche nach automatischer Vervollständigung einschränken, um nur Vorschläge für Orte zu erhalten, die sich in einem bestimmten Land oder einer bestimmten Gruppe von Ländern befinden, indem Sie den folgenden optionalen Parameter hinzufügen:

- `FilterCountries` – Die Länder, in denen Sie nach Ortsvorschlägen suchen möchten. Sie können bis zu 100 Länder in einer Anfrage mit einem dreistelligen [ISO-3166](#)-Ländercode angeben. Verwenden Sie beispielsweise AUS für Australien.

Im folgenden Beispiel wird eine [SearchPlaceIndexForSuggestions](#)Anforderung verwendet, um die Ortsindex-Ressource *ExamplePlaceIndex* nach Ortsvorschlägen zu durchsuchen, die mit der partiellen Abfrage „*Camp*“ übereinstimmen und in Ug Bol, Kenya oderania enthalten sind:

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}
```

Die für dasselbe zurückgegebenen Vorschläge Text unterscheiden sich, wenn eine andere `FilterCountries` Liste ausgewählt wird, z. B. ["USA "].

Beispielantwort

Im Folgenden finden Sie eine Beispielantwort vorgeschlagener automatischer Vervollständigungen für die [SearchPlaceIndexForSuggestions](#)Operation unter Verwendung des Text-*Zumpfes* .

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
```

```

    "PlaceId": "AQAAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
  },
  {
    "Text": "Kampoul, Kabul, AFG",
    "PlaceId":
"AQAAAIAAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeB1nbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
  },
  {
    "Text": "Kampala, UGA",
    "PlaceId":
"AQAAAIAAzZfZt3qMrUKG0byhP6MM0pqy2L8SUL1VWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8jLYKMwvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZ1dvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
  },
  {
    "Text": "Kampar, Riau, IDN",
    "PlaceId": "AQAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Col4fLu7IK0yY0LhZx4k
  },
  {
    "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
    "PlaceId":
"AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
  }
]
}

```

Verwenden von Orts-IDs

Die Suche nach Orten gibt eine Liste von Ergebnissen zurück. Die meisten Ergebnisse enthalten einen PlaceId für dieses Ergebnis. Sie können einen PlaceId in einer [GetPlace](#) Operation verwenden, um die Informationen über diesen Ort (einschließlich Name, Adresse, Standort oder anderer Details) zurückzugeben.

Note

Die Verwendung von [SearchPlaceIndexForSuggestions](#) gibt PlaceId Ergebnisse für alle Ortsindizes zurück, die mit einer beliebigen Datenquelle erstellt wurden. Die Verwendung von

[SearchPlaceIndexForText](#) oder [SearchPlaceIndexForPosition](#) gibt PlaceId nur dann einen zurück, wenn die verwendete Datenquelle HERE ist.

Jede definiert PlaceId eindeutig den Ort, auf den sie sich bezieht, aber ein einzelner Ort kann im PlaceId Laufe der Zeit mehr als einen haben und basiert auf dem Kontext. Die folgenden Regeln beschreiben die Eindeutigkeit und Lebensdauer eines PlaceId.

- Die in Aufrufen PlaceId zurückgegebenen sind spezifisch für Ihr AWS-Konto, die AWS Region und den Datenanbieter in Ihrer PlaceIndex Ressource. GetPlace findet Ergebnisse nur, wenn diese drei Attribute mit dem ursprünglichen Aufruf übereinstimmen, der das erstellt hat PlaceId.
- Die PlaceId für einen Ort ändert sich, wenn sich die Daten zu diesem Ort ändern. Wenn beispielsweise das Unternehmen, auf das es sich bezieht, den Standort verschiebt oder Namen ändert.
- Der von einem wiederholten Suchaufruf PlaceId zurückgegebene kann sich ändern, wenn der Backend-Service eine Aktualisierung vornimmt. Das ältere PlaceId wird weiterhin gefunden, aber neue Suchaufrufe geben möglicherweise eine andere ID zurück.

ist PlaceId eine Zeichenfolge. Es gibt keine spezifische Begrenzung für die Länge eines PlaceId. Im Folgenden finden Sie ein Beispiel für einen gültigen PlaceId.

```
AQAAAIADsn2T3KdrWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAi8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Wenn Sie GetPlace mit einem PlaceId für einen Ort aufrufen, dessen Daten sich geändert haben (z. B. einen Geschäftsstandort, der nicht mehr funktioniert), führt dies zu einem 404-, -ResourceNotFoundFehler. Wenn Sie GetPlace mit einer aufrufen, PlaceId die nicht gültig ist, oder einer, die nicht im Kontext ist, z. B. von einer anderen AWS-Konto, wird ein 400-ValidationExceptionFehler zurückgegeben.

Sie können PlaceID zwar in nachfolgenden Anforderungen verwenden, PlaceID ist jedoch nicht als permanente Kennung gedacht und die ID kann sich zwischen aufeinanderfolgenden API-Aufrufen ändern. Bitte beachten Sie das folgende PlaceID-Verhalten für jeden Datenanbieter:

- Esri : Orts-IDs ändern sich mindestens jedes Quartal. Der typische Zeitraum für diese Änderungen wäre März, Juni, September und Dezember. Orts-IDs können sich auch zwischen der typischen vierteljährlichen Änderung ändern, aber das wird viel seltener sein.
- HERE : Wir empfehlen, dass Sie Daten nicht länger als eine Woche zwischenspeichern, um Ihre Daten auf dem neuesten Stand zu halten. Sie können davon ausgehen, dass weniger als 1 % ID-Schichten gegenüber der Veröffentlichung freigegeben werden, was etwa 1–2 Mal pro Woche entspricht.
- Grab : Orts-IDs können in den folgenden Situationen ablaufen oder ungültig werden.
 - Datenoperationen: Das POI kann von Grab Map Ops auf der Grundlage der Grundwahrheit aus der Grab-POI-Datenbank entfernt werden, z. B. in der realen Welt geschlossen, als doppeltes POI erkannt oder falsche Informationen vorliegen. Grab synchronisiert Daten wöchentlich mit der Waypoint-Umgebung.
 - Interpoliertes POI: Interpoliertes POI ist ein temporäres POI, das bei der Verarbeitung einer Anforderung in Echtzeit generiert wird, und es wird im `place.result_type` Feld in der Antwort als abgeleitet markiert. Die Informationen zu interpolierten POIs werden mindestens 30 Tage lang aufbewahrt, was bedeutet, dass Sie innerhalb von 30 Tagen POI-Details nach Orts-ID von der API für Ortsdetails abrufen können. Nach 30 Tagen können die interpolierten POIs (sowohl Orts-ID als auch Details) ablaufen und über die API für Ortsdetails nicht mehr zugänglich sein.

Kategorien und Filterergebnisse platzieren

Standorte werden kategorisiert. Wenn Sie nach einem Unternehmen suchen, könnte das Unternehmen Restaurant beispielsweise ein sein. Selbst die Ergebnisse einer Suche nach einer Adresse können danach kategorisiert werden, ob sie mit einer Adresse, Straßen oder Schnittmenge übereinstimmt.

Im Allgemeinen kategorisiert Amazon Location Service Platzierungen in Ortstypen. Die Interessenpunkte werden weiter in die Interessenpunkte unterteilt.

Note

Nicht alle Ergebnisse haben Kategorien.

Sie können die Kategorien verwenden, um Ihre Geocodierungssuchen zu filtern.

Filtern von Ergebnissen

Wenn Sie verwenden `SearchPlaceIndexForText`, können Sie die Ergebnisse filtern, die von den Kategorien zurückgegeben werden, die Sie verwenden möchten. Beispielsweise:

- Wenn Sie nach einem Ort mit dem Namen „HomeSpeed“ suchen und nur Ergebnisse zurückgeben möchten, die als Fußbälle kategorisiert sind, können Sie dies tun, indem Sie aufrufen `SearchPlaceIndexForText` und die Kategorie `Interessenpunkt Coffee Shop` in den `-FilterCategoriesParameter` aufnehmen.
- Wenn Sie nach „123 Main St, Any, WA, 98123, USA“ suchen, können Sie das Ergebnis nur nach Adressen filtern, sodass Sie keine Übereinstimmungen erhalten, z. B. nach der Postleitzahl. Filtern Sie nur nach Adressen, indem Sie den Ortstyp `AddressType` in den `FilterCategories` Parameter einschließen.

Note

Nicht alle Datenanbieter unterstützen Filterung oder sie auf die gleiche Weise. Weitere Informationen finden Sie unter [Filtern von Einschränkungen nach Datenanbieter](#).

Im nächsten Abschnitt werden die Kategorien aufgeführt, nach denen Sie filtern können.

Kategorien

Die folgenden Listen zeigen die Kategorien, die Amazon Location Service zur Kategorisierung und Filterung verwendet. Diese Kategorien werden in allen Sprachen verwendet, unabhängig davon, ob der Sprachparameter auf eine andere Sprache festgelegt ist.

Note

Amazon Location Service ordnet Datenanbieterkategorien dieser Gruppe von Kategorien zu. Wenn ein Datenanbieter eine Stelle in eine Kategorie einfügt, die nicht Teil der Kategorieliste von Amazon Location Service ist, wird die Anbieterkategorie als zusätzliche Kategorie in die Ergebnisse aufgenommen.

Ortstypen – Diese Typen werden verwendet, um die Art der Übereinstimmung anzugeben, die zur Suche des Ergebnisses verwendet wurde.

- `AddressType` – Wird zurückgegeben, wenn das Ergebnis einer Adresse zugeordnet wurde.
- `StreetType` – Wird zurückgegeben, wenn das Ergebnis einer Straßen zugeordnet wurde.
- `IntersectionType` – Wird zurückgegeben, wenn das Ergebnis mit der Schnittmenge von zwei Straßen übereinstimmt.
- `PointOfInterestType` – Wird zurückgegeben, wenn das Ergebnis mit einem Interessenpunkt übereinstimmte, z. B. einem Unternehmen oder einem bivischen Standort.
- `CountryType` – Wird zurückgegeben, wenn das Ergebnis mit einem Land oder einer Hauptregion übereinstimmt.
- `RegionType` – Wird zurückgegeben, wenn das Ergebnis mit einer Region innerhalb eines Landes abgeglichen wurde, z. B. einem Bundesstaat oder einer Provinz.
- `SubRegionType` – Wird zurückgegeben, wenn das Ergebnis mit einer Unterregion innerhalb eines Landes abgeglichen wurde, z. B. einem Landkreis oder einem Stadtgebiet.
- `MunicipalityType` – Wird zurückgegeben, wenn das Ergebnis mit einer Stadt oder einem Arzt übereinstimmt.
- `NeighborhoodType` – Wird zurückgegeben, wenn das Ergebnis mit einer Nachbarschaft oder einem Gebiet innerhalb einer Stadt übereinstimmt.
- `PostalCodeType` – Wird zurückgegeben, wenn das Ergebnis mit einer Postleitzahl abgeglichen wurde.

Kategorien von Interessenpunkten – Diese Kategorien werden verwendet, um die Art des Unternehmens oder Standorts für die Ergebnisse von Interessenpunkten anzugeben.

- `Airport`
- `Amusement Park`
- `Aquarium`
- `Art Gallery`
- `ATM`
- `Bakery`
- `Bank`
- `Bar`
- `Beauty Salon`
- `Bus Station`

- Car Dealer
- Car Rental
- Car Repair
- Car Wash
- Cemetery
- Cinema
- City Hall
- Clothing Store
- Coffee Shop
- Consumer Electronics Store
- Convenience Store
- Court House
- Dentist
- Embassy
- Fire Station
- Fitness Center
- Gas Station
- Government Office
- Grocery
- Higher Education
- Hospital
- Hotel
- Laundry
- Library
- Liquor Store
- Lodging
- Market
- Medical Clinic
- Motel
- Museum

- Nightlife
- Nursing Home
- Park
- Parking
- Pet Store
- Pharmacy
- Plumbing
- Police Station
- Post Office
- Religious Place
- Restaurant
- School
- Shopping Mall
- Sports Center
- Storage
- Taxi Stand
- Tourist Attraction
- Train Station
- Veterinary Care
- Zoo

Filtern von Einschränkungen nach Datenanbieter

Nicht alle Anbieter verfügen über dieselbe Filterfunktionalität. In der folgenden Tabelle werden die Unterschiede beschrieben.

Anbieter	APIs mit Filterunterstützung	Für die Filterung unterstützte Kategorien	Rückgabewerte
Esri	SearchPlaceIndexFo	Filtern Sie nach Ortstypen und	Kategorien werden von SearchPla

Anbieter	APIs mit Filterunterstützung	Für die Filterung unterstützte Kategorien	Rückgabewerte
	<code>rText</code> , <code>SearchPlaceIndexForSuggestions</code>	Interessenpunktategorien.	<code>ceIndexForText</code> , <code>SearchPlaceIndexForPosition</code> und zurückgegeben <code>GetPlace</code>
Hier	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Nur nach Ortstypen filtern.	Kategorien werden von <code>SearchPlaceIndexForText</code> und <code>SearchPlaceIndexForSuggestions</code> , <code>SearchPlaceIndexForPosition</code> und zurückgegeben <code>GetPlace</code>
Grab	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt
Daten öffnen	n/a (Suche nach Orten wird nicht unterstützt)	–	–

Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service

Sie können Amazon Location Service verwenden, um mit Koordinaten und Adressen zu arbeiten, die in Datenbanktabellen gespeichert sind, um Ihre Geodaten zu bereinigen und zu ergänzen.

Beispielsweise:

- Sie können Geokodierung verwenden, um Adressen in Koordinaten zu konvertieren, um Datenlücken für Adressen zu normalisieren und zu füllen, die in einer Datenbanktabelle gespeichert sind.
- Sie können Adressen geocodieren, um ihre Positionen zu erhalten, und die Koordinaten mit räumlichen Datenbankfunktionen verwenden, z. B. mit einer Funktion, die Zeilen in einem bestimmten Bereich anzeigt.
- Sie können mit erweiterten Daten automatisierte Berichte erstellen, z. B. einen automatisierten Bericht, der alle Geräte in einem bestimmten Bereich veranschaulicht, oder einen automatisierten Bericht für Machine Learning, der Bereiche mit höheren Fehlerraten beim Senden von Standortaktualisierungen veranschaulicht.

Dieses Tutorial zeigt, wie Adressen formatiert und anreicht werden, die in einer -Amazon Aurora PostgreSQL-Datenbanktabelle mit Amazon Location Service gespeichert sind.

- Amazon Aurora PostgreSQL – Eine vollständig verwaltete, mit MySQL und PostgreSQL kompatible relationale Datenbank-Engine, die einen bis zu fünffachen Durchsatz von MySQL und einen bis zu dreimal höheren Durchsatz von PostgreSQL ausgibt, ohne den Großteil Ihrer vorhandenen Anwendung zu ändern. Weitere Informationen finden Sie unter [Was ist Amazon Aurora?](#) im Amazon-Aurora-Benutzerhandbuch.

Important

Die resultierende Anwendung in diesem Tutorial verwendet einen Ortsindex, der Geocodierungsergebnisse speichert. Weitere Informationen zu den Gebühren für die Speicherung von Geocodierungsergebnissen finden Sie unter [Amazon Location Service – Preise](#).

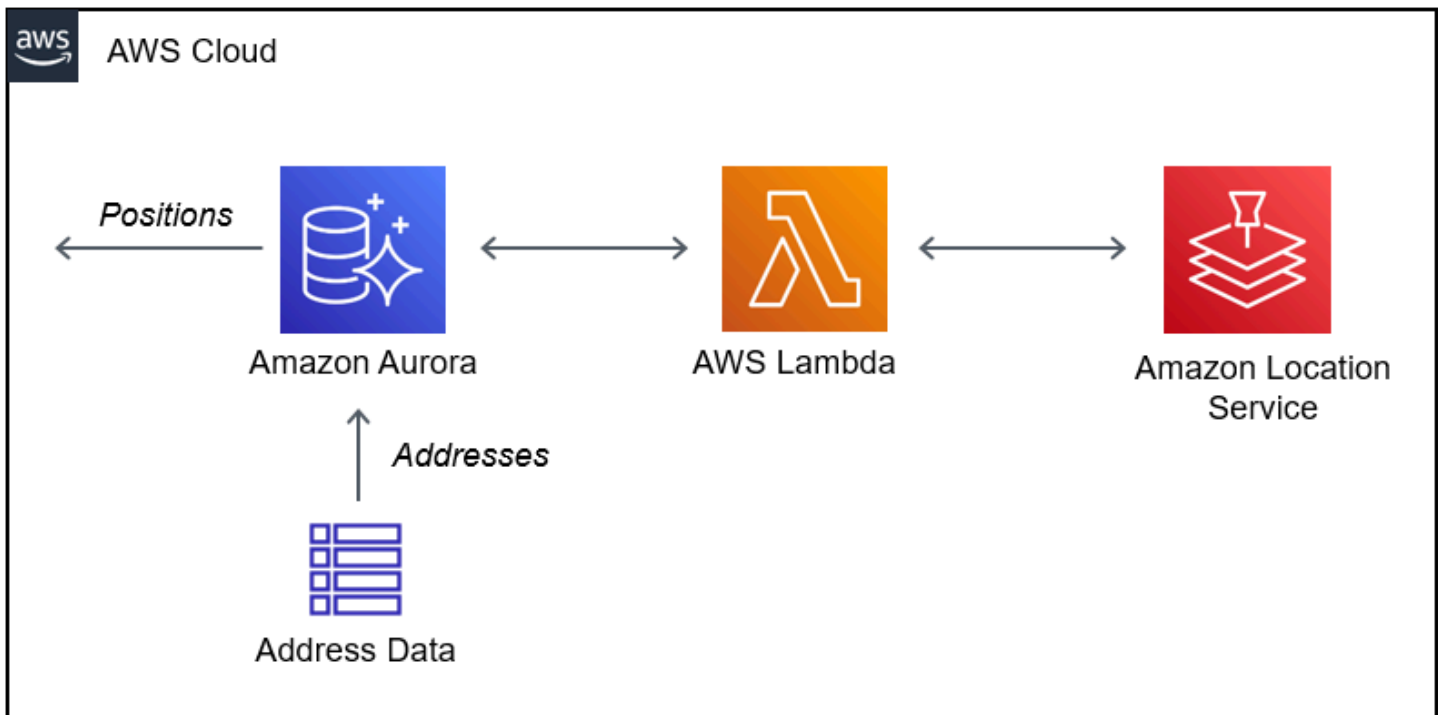
Beispielcode ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#), das [eine - AWS CloudFormation Vorlage](#) enthält.

Themen

- [Übersicht](#)
- [Voraussetzungen](#)
- [Schnellstart](#)

- [Erstellen einer Ortsindex-Ressource](#)
- [Erstellen einer -AWS LambdaFunktion für die Geocodierung](#)
- [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#)
- [Aufrufen der AWS Lambda-Funktion](#)
- [Anreicherung einer Datenbank mit Adressdaten](#)
- [Nächste Schritte](#)

Übersicht



Die Architektur umfasst die folgenden Integrationen:

- Diese Lösung verwendet eine Amazon-Location-Place-Indexressource, um Geocodierungsabfragen mit der Operation `SearchPlaceIndexForText` zu unterstützen.
- AWS Lambda verwendet ein Python-Lambda, das Adressen geocodiert, wenn eine IAM-Richtlinie die Berechtigung erteilt, AWS Lambda die Geocodierungsoperation von Amazon Location aufzurufen, `SearchPlaceIndexForText`.
- Erteilen Sie die Berechtigung Amazon Aurora PostgreSQL zum Aufrufen der Geocodierungs-Lambda-Funktion mithilfe einer benutzerdefinierten SQL-Funktion.

Voraussetzungen

Bevor Sie beginnen, benötigen Sie die folgenden Voraussetzungen:

- Ein -Amazon Aurora PostgreSQLCluster. Weitere Informationen zum [Erstellen eines Amazon-Aurora-DB-Clusters](#) finden Sie im Amazon-Aurora-Benutzerhandbuch.

Note

Wenn Ihr Amazon-Aurora-Cluster nicht öffentlich verfügbar ist, müssen Sie Amazon Aurora auch so konfigurieren, dass eine Verbindung zu AWS Lambda in einer Virtual Private Cloud (VPC) in Ihrem AWS Konto hergestellt wird. Weitere Informationen finden Sie unter [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#).

- Ein SQL-Entwicklertool zum Herstellen einer Verbindung mit dem Amazon Aurora PostgreSQL Cluster.

Schnellstart

Als Alternative zu den Schritten in diesem Tutorial können Sie einen Quick Stack starten, um eine -AWS LambdaFunktion bereitzustellen, die den Amazon Location-Vorgang unterstützt [SearchPlaceIndexForText](#). Dadurch wird Ihr AWS Konto automatisch so konfiguriert, dass Amazon Aurora aufrufen kannAWS Lambda.

Sobald Sie Ihr AWS Konto konfiguriert haben, müssen Sie:

- Fügen Sie das Lambda-Feature zu Amazon Aurora hinzu. Weitere Informationen finden Sie unter Hinzufügen der IAM-Rolle zu einem Amazon-Aurora-DB-Cluster in [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#).
- Laden Sie die benutzerdefinierte Funktion in Ihre Datenbank. Siehe [Aufrufen der AWS Lambda-Funktion](#).

Launch Stack 

Erstellen einer Ortsindex-Ressource

Erstellen Sie zunächst eine Ortsindexressource zur Unterstützung von Geocodierungsabfragen.

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Ortsindex-Ressource ein. Zum Beispiel *AuroraPlaceIndex*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Zum Beispiel den *Ortsindex für Amazon Aurora*.
4. Wählen Sie unter Datenanbieter einen verfügbaren [Datenanbieter](#) aus, der mit Ihrer Ortsindexressource verwendet werden soll. Wenn Sie keine Präferenz haben, empfehlen wir, mit *Esri* zu beginnen.
5. Geben Sie unter Datenspeicheroptionen die Option Ja an, Ergebnisse werden gespeichert. Dies weist darauf hin, dass Sie beabsichtigen, die Geocodierungsergebnisse in einer Datenbank zu speichern.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag für Ihre neue Ortsindexressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. Wählen Sie Ortsindex erstellen aus.

Erstellen einer -AWS LambdaFunktion für die Geocodierung

Um eine Verbindung zwischen Amazon Aurora PostgreSQL und Amazon Location Service herzustellen, benötigen Sie eine -AWS LambdaFunktion, um Anfragen von der Datenbank-Engine zu verarbeiten. Diese Funktion übersetzt das benutzerdefinierte Lambda-Funktionsereignis und ruft die Amazon Location-Operation aufSearchPlaceIndexForText.

Sie können die Funktion mithilfe der AWS LambdaKonsole, der AWS Command Line Interfaceoder der AWS Lambda APIs erstellen.

So erstellen Sie eine benutzerdefinierte Lambda-Funktion mithilfe der Konsole

1. Öffnen Sie die AWS Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie in der linken Navigation Funktionen aus.

3. Wählen Sie Funktion erstellen und stellen Sie sicher, dass Von Grund auf neu erstellen ausgewählt ist.
4. Füllen Sie die folgenden Felder aus:
 - Funktionsname – Geben Sie einen eindeutigen Namen für Ihre Funktion ein. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche und Unterstriche ohne Leerzeichen. Zum Beispiel *AuroraGeocoder*.
 - Laufzeit – Wählen Sie *Python 3.8* aus.
5. Wählen Sie Funktion erstellen.
6. Wählen Sie die Registerkarte Code, um den Editor zu öffnen.
7. Überschreiben Sie den Platzhaltercode in `lambda_function.py` wie folgt:

```
from os import environ

import boto3
from botocore.config import Config

# load the place index name from the environment, falling back to a default
PLACE_INDEX_NAME = environ.get("PLACE_INDEX_NAME", "AuroraPlaceIndex")

location = boto3.client("location", config=Config(user_agent="Amazon Aurora
  PostgreSQL"))

"""
This Lambda function receives a payload from Amazon Aurora and translates it to
an Amazon Location `SearchPlaceIndex` call and returns the results as-is, to be
post-processed by a PL/pgSQL function.
"""
def lambda_handler(event, context):
    kwargs = {}

    if event.get("biasPosition") is not None:
        kwargs["BiasPosition"] = event["biasPosition"]

    if event.get("filterBBox") is not None:
        kwargs["FilterBBox"] = event["filterBBox"]

    if event.get("filterCountries") is not None:
        kwargs["FilterCountries"] = event["filterCountries"]

    if event.get("maxResults") is not None:
```

```
kwargs["MaxResults"] = event["maxResults"]

return location.search_place_index_for_text(
    IndexName=PLACE_INDEX_NAME,
    Text=event["text"],
    **kwargs)["Results"]
```

8. Wenn Sie Ihren Ortsindex anders als benannt haben *AuroraPlaceIndex*, erstellen Sie eine Umgebungsvariable mit dem Namen `PLACE_INDEX_NAME`, um den Ressourcennamen zuzuweisen:
 - Wählen Sie auf der Registerkarte Konfiguration die Option Umgebungsvariablen aus.
 - Wählen Sie Bearbeiten und dann Umgebungsvariable hinzufügen aus.
 - Geben Sie für Schlüssel ein `PLACE_INDEX_NAME`.
 - Geben Sie für Wert den Namen Ihrer Ortsindex-Ressource ein.
9. Wählen Sie Bereitstellen, um die aktualisierte Funktion zu speichern.
10. Wählen Sie im Dropdown-Menü Test die Option Testereignis konfigurieren aus.
11. Wählen Sie Create new test event aus.
12. Geben Sie das folgende Testereignis ein:

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Wählen Sie Test, um die Lambda-Funktion zu testen.
14. Wählen Sie die Registerkarte Konfiguration aus.
15. Wählen Sie unter Allgemeine Konfiguration die Option Berechtigungen aus.
16. Wählen Sie unter Ausführungsrolle den Hyperlink-Rollennamen aus, um Amazon Location Service Berechtigungen für Ihre Lambda-Funktion zu erteilen.
17. Wählen Sie auf der Registerkarte Berechtigungen: Wählen Sie das Dropdown-Menü Berechtigungen hinzufügen und dann Inline-Richtlinie erstellen aus.
18. Wählen Sie den Tab JSON.
19. Fügen Sie die folgende IAM-Richtlinie hinzu:

- Die folgende Richtlinie erteilt die Berechtigung zum Senden `SearchPlaceIndexForText` an die Ortsindex-Ressource *AuroraPlaceIndex*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:SearchPlaceIndexForText",
      "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
    }
  ]
}
```

20. Wählen Sie Richtlinie prüfen.
21. Geben Sie den Namen einer Richtlinie ein. Zum Beispiel *AuroraPlaceIndexReadOnly*.
22. Wählen Sie Richtlinie erstellen aus.

Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda

Bevor eine -AWS LambdaFunktion aufrufen Amazon Aurora PostgreSQL kann, müssen Sie Zugriffsberechtigungen erteilen.

Wenn Ihr Amazon Aurora PostgreSQL Cluster nicht öffentlich zugänglich ist, müssen Sie zunächst einen VPC-Endpunkt für erstellenAWS Lambda, damit Amazon Aurora Ihre Lambda-Funktion aufrufen kann.

Erstellen eines VPC-Endpunkts für AWS Lambda

Note

Dieser Schritt ist nur erforderlich, wenn Ihr Amazon Aurora PostgreSQL Cluster nicht öffentlich zugänglich ist.

1. Öffnen Sie die [Amazon Virtual Private Cloud Console](#).
2. Wählen Sie in der linken Navigation Endpunkte aus.
3. Wählen Sie Endpunkt erstellen aus.

4. Geben Sie im Filter Service Name „lambda“ ein und wählen Sie dann `auscom.amazonaws.<region>.lambda`.
5. Wählen Sie die VPC aus, die Ihren Aurora-Cluster enthält.
6. Wählen Sie für jede Availability Zone ein Subnetz aus.
7. Geben Sie im Filter Sicherheitsgruppe „Standard“ oder den Namen der Sicherheitsgruppe ein, der Ihr Aurora-Cluster angehört, und wählen Sie dann die Sicherheitsgruppe aus.
8. Wählen Sie Endpunkt erstellen aus.

Erstellen Sie eine IAM-Richtlinie, um die Berechtigung zum Aufrufen Ihrer AWS Lambda Funktion zu erteilen

1. Öffnen Sie die [IAM-Konsole](#).
2. Erweitern Sie in der linken Navigation die Option Zugriffsverwaltung, um Richtlinien auszuwählen.
3. Wählen Sie Richtlinie erstellen aus.
4. Geben Sie auf der Registerkarte JSON die folgende Richtlinie ein:
 - Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die die Amazon Aurora PostgreSQL Berechtigung zum Aufrufen der AuroraGeocoder AWS Lambda Funktion erteilt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Wählen Sie Weiter: Tags, um optionale Tags hinzuzufügen.
6. Wählen Sie Weiter: Prüfen aus.
7. Überprüfen Sie Ihre Richtlinie und geben Sie die folgenden Details für die Richtlinie ein:

- Name – Verwenden Sie alphanumerische Zeichen und „+=,@-“. Maximal 128 Zeichen. Zum Beispiel *AuroraGeocoderInvoke*.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Verwenden Sie alphanumerische Zeichen und '+=,@-'. Maximal 1000 Zeichen.
8. Wählen Sie Richtlinie erstellen aus. Notieren Sie sich den ARN für diese Richtlinie, mit dem Sie die Richtlinie an eine IAM-Rolle anfügen.

Erstellen einer IAM-Rolle, um Amazon Relational Database Service (Amazon RDS) die Berechtigung zu erteilen

Durch Erstellen einer IAM-Rolle Amazon Aurora PostgreSQL kann die Rolle in Ihrem Namen übernehmen, um auf Ihre Lambda-Funktion zuzugreifen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.

Das folgende Beispiel ist ein -AWS CLIBefehl, der eine Rolle mit dem Namen erstellt *AuroraGeocoderInvokeRole*:

```
aws iam create-role --role-name rds-lambda-role --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Hängen Sie Ihre IAM-Richtlinie an die IAM-Rolle an

Wenn Sie über eine IAM-Rolle verfügen, fügen Sie die von Ihnen erstellte IAM-Richtlinie an.

Das folgende Beispiel ist ein -AWS CLIBefehl, der die Richtlinie *AuroraGeocoderInvoke* an die Rolle anfügt *AuroraGeocoderInvokeRole*.


```
aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-  
name AuroraGeocoderInvokeRole
```

Hinzufügen der IAM-Rolle zu einem Amazon Aurora-DB-Cluster

Das folgende Beispiel ist ein -AWS CLIBefehl zum Hinzufügen einer IAM-Rolle zu einem Amazon Aurora PostgreSQL DB-Cluster mit dem Namen *MyAuroraCluster*.

```
aws rds add-role-to-db-cluster \  
--db-cluster-identifier MyAuroraCluster \  
--feature-name Lambda \  
--role-arn AuroraGeocoderInvokeRole \  
--region your-region
```

Aufrufen der AWS Lambda-Funktion

Nachdem Sie die Berechtigung zum Aufrufen Ihrer Lambda-Funktion Amazon Aurora PostgreSQL für die Geocodierung erteilt haben, können Sie eine Amazon Aurora PostgreSQL benutzerdefinierte Funktion erstellen, um die AWS Lambda Geocodierungsfunktion aufzurufen. Weitere Informationen finden Sie unter [Aufrufen einer -AWS Lambda-Funktion aus einem Amazon Aurora PostgreSQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.

Installieren der erforderlichen PostgreSQL-Erweiterungen

Informationen zum Installieren der erforderlichen PostgreSQL-Erweiterungen `aws_lambda` und `aws_commons` Erweiterungen finden Sie unter [Übersicht über die Verwendung einer Lambda-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

Installieren der erforderlichen PostGIS-Erweiterungen

PostGIS ist eine Erweiterung von PostgreSQL zur Speicherung und Verwaltung von Geodaten. Weitere Informationen finden Sie unter [Arbeiten mit der PostGIS-Erweiterung](#) im Amazon Relational Database Service-Benutzerhandbuch.

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Erstellen einer benutzerdefinierten SQL-Funktion, die die Lambda-Funktion aufruft

Erstellen Sie in einem SQL-Editor eine neue benutzerdefinierte Funktion, `f_SearchPlaceIndexForText` um die Funktion aufzurufen *AuroraGeocoder*:

```
CREATE OR REPLACE FUNCTION f_SearchPlaceIndexForText(
  text text,
  bias_position geometry(Point, 4326) DEFAULT NULL,
  filter_bbox box2d DEFAULT NULL,
  filter_countries text[] DEFAULT NULL,
  max_results int DEFAULT 1
)
RETURNS TABLE (
  label text,
  address_number text,
  street text,
  municipality text,
  postal_code text,
  sub_region text,
  region text,
  country text,
  geom geometry(Point, 4326)
)
LANGUAGE plpgsql
IMMUTABLE
AS $function$
begin
  RETURN QUERY
  WITH results AS (
    SELECT json_array_elements(payload) rsp
    FROM aws_lambda.invoke(
      aws_commons.create_lambda_function_arn('AuroraGeocoder'),
      json_build_object(
        'text', text,
        'biasPosition',
        CASE WHEN bias_position IS NOT NULL THEN
          array_to_json(ARRAY[ST_X(bias_position), ST_Y(bias_position)])
        END,
        'filterBBox',
        CASE WHEN filter_bbox IS NOT NULL THEN
          array_to_json(ARRAY[ST_XMin(filter_bbox), ST_YMin(filter_bbox),
ST_XMax(filter_bbox), ST_YMax(filter_bbox)])
        END,
        'filterCountries', filter_countries,
        'maxResults', max_results
```

```
    )
  )
)
SELECT
  rsp->'Place'->>'Label' AS label,
  rsp->'Place'->>'AddressNumber' AS address_number,
  rsp->'Place'->>'Street' AS street,
  rsp->'Place'->>'Municipality' AS municipality,
  rsp->'Place'->>'PostalCode' AS postal_code,
  rsp->'Place'->>'SubRegion' AS sub_region,
  rsp->'Place'->>'Region' AS region,
  rsp->'Place'->>'Country' AS country,
  ST_GeomFromGeoJSON(
    json_build_object(
      'type', 'Point',
      'coordinates', rsp->'Place'->'Geometry'->'Point'
    )
  ) geom
FROM results;
end;
$function$;
```

Rufen Sie die SQL-Funktion auf, um von Aurora aus zu geocodieren

Wenn Sie die SQL-Anweisung ausführen, wird die Lambda-Funktion aufgerufen *AuroraGeocoder*, die Adressdatensätze aus der Datenbanktabelle in der Amazon Aurora PostgreSQL Datenbank übernimmt und sie mithilfe einer Ortsindexressource geocodiert.

Note

Amazon Aurora PostgreSQL ruft die Lambda-Funktion für jeden Aufruf der benutzerdefinierten SQL-Funktion auf.

Wenn Sie 50 Zeilen geocodieren, Amazon Aurora PostgreSQL ruft die Lambda-Funktion 50 Mal auf. Ein Aufruf für jede Zeile.

Die folgende `f_SearchPlaceIndexForText` SQL-Funktion stellt Anforderungen an die [SearchPlaceIndexForText](#) API von Amazon Location über die *AuroraGeocoder* Lambda-Funktion. Die Funktion gibt eine `geom` Spalte zurück, die eine PostGIS-Geometrie ist, die in `Text_ST_AsText(geom)` konvertiert wird.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

Standardmäßig enthält die Rückgabe eine Zeile. Um zusätzliche Zeilen bis zum `MaxResults` Limit anzufordern, führen Sie die folgende SQL-Anweisung aus, während Sie ein `BiasPosition` und eine Einschränkung auf die Ergebnisse in Kanada angeben.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Um Ergebnisse mithilfe eines Begrenzungsrahmens zu filtern, übergeben Sie einen [Box2D](#) als `filter_bbox`:

- [FilterBBox](#) – Filtert die Ergebnisse, indem es Stellen innerhalb eines Begrenzungsrahmens zurückgibt. Dieser Parameter ist optional.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)::box2d, '{"CAN"}', 5);
```

Weitere Informationen zu PostGIS-Typen und -Funktionen finden Sie in der [PostGIS-Referenz](#).

Anreicherung einer Datenbank mit Adressdaten

Sie können eine formatierte Adresse erstellen und gleichzeitig mithilfe der Amazon Location-Operation normalisieren und geocodieren, `SearchPlaceIndexForText` wenn eine Datenbanktabelle mit den folgenden Daten in die folgenden Spalten unterteilt ist:

- `id`
- `address`
- `city`
- `state`
- `zip`

```
WITH source_data AS (
```

```

SELECT
  id,
  address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
FROM addresses
),
geocoded_data AS (
  SELECT
    *,
    (f_SearchPlaceIndexForText(formatted_address)).*
  FROM source_data
)
SELECT
  id,
  formatted_address,
  label normalized_address,
  ST_Y(geom) latitude,
  ST_X(geom) longitude
FROM geocoded_data
-- limit the number of rows that will be geocoded; remove this to geocode the entire
table
LIMIT 1;

```

Das folgende Beispiel veranschaulicht eine resultierende Datentabellenzeile:

```

id |          formatted_address          |          normalized_address          |
latitude | longitude |
-----+-----+-----
+-----+-----+-----
42 | 123 Anytown Ave N, Seattle, WA | 123 Anytown Ave N, Seattle, WA, 12345, USA |
47.6223000127926 | -122.336745971039
(1 row)

```

Aktualisieren der Datenbanktabelle und Auffüllen von Spalten

Im folgenden Beispiel wird die Tabelle aktualisiert und Spalten mit Ergebnissen von `SearchPlaceIndexForText` Abfragen gefüllt:

```

WITH source_data AS (
  -- select rows that have not been geocoded and created a formatted address for each
  SELECT
    id,
    address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
  FROM addresses

```

```
WHERE label IS NULL
-- limit the number of rows that will be geocoded; remove this to geocode the entire
table
LIMIT 1
),
geocoded_data AS (
-- geocode each row and keep it linked to the source's ID
SELECT
    id,
    (f_SearchPlaceIndexForText(formatted_address)).*
FROM source_data
)
UPDATE addresses
-- populate columns
SET
    normalized_address = geocoded_data.label,
    latitude = ST_Y(geocoded_data.geom),
    longitude = ST_X(geocoded_data.geom)
FROM geocoded_data
-- ensure that rows match
WHERE addresses.id = geocoded_data.id;
```

Nächste Schritte

Beispielcode ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#), das [eine - AWS CloudFormationVorlage](#) enthält.

Verwalten Ihrer Ortsindexressourcen

Sie können Ihre Ortsindexressourcen mithilfe der Amazon-Standortkonsole, der AWS CLI oder der Amazon-Standort-APIs verwalten.

Auflisten Ihrer Ortsindexressourcen

Sie können Ihre Ortsindex-Ressourcenliste über die Amazon-Standortkonsole, die AWS CLI, die oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der Ortsindexressourcen mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.

2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Zeigen Sie eine Liste Ihrer Ortsindexressourcen unter Meine Ortsindizes an.

API

Verwenden Sie die [-ListPlaceIndexes](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Ortsindexressourcen im AWS Konto.

```
POST /places/v0/list-indexes
```

Im Folgenden finden Sie eine Beispielantwort für [ListPlaceIndexes](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-30T01:38:36Z,
      "DataSource": "Esri",
      "Description": "string",
      "IndexName": "ExamplePlaceIndex",
      "UpdateTime": 2020-10-30T01:40:36Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-place-indexes](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste der Ortsindexressourcen im AWS Konto abzurufen.

```
aws location list-place-indexes
```

Abrufen von Details zu Ortsindexressourcen

Sie können Details zu jeder Ortsindexressource in Ihrem AWS Konto mithilfe der Amazon Location-Konsole AWS CLI, der oder der Amazon Location APIs abrufen:

Console

So zeigen Sie die Details einer Ortsindexressource mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie unter Meine Ortsindizes den Namenslink der Zielort-Indexressource aus.

API

Verwenden Sie die [DescribePlaceIndex](#) Operation aus den Amazon Location Place APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Ortsindex-Ressourcendetails für *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

Im Folgenden finden Sie eine Beispielantwort für [DescribePlaceIndex](#):

```
{
  "CreateTime": 2020-10-30T01:38:36Z,
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexArn": "arn:aws:geo:us-west-2:123456789012:place-indexes/ExamplePlaceIndex",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "string" : "string"
  },
  "UpdateTime": 2020-10-30T01:40:36Z
}
```

CLI

Verwenden Sie den [describe-place-index](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details zur Ortsindex-Ressource für *abzurufenExamplePlaceIndex*.


```
aws location describe-place-index \  
  --index-name "ExamplePlaceIndex"
```

Löschen einer Ortsindex-Ressource

Sie können eine Ortsindexressource aus Ihrem AWS-Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen:

Console

So löschen Sie eine Ortsindexressource mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie unter Mein Ortsindex die Zielortindexressource aus.
4. Wählen Sie Ortsindex löschen aus.

API

Verwenden Sie die [DeletePlaceIndex](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Ortsindex-Ressource *ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [DeletePlaceIndex](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-place-index](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Ortsindex-Ressource *ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```

Berechnen von Routen mit Amazon Location Service

Mit Amazon Location können Sie einen Datenanbieter für die Berechnung einer Route auswählen, indem Sie eine Routenrechner-Ressource erstellen und konfigurieren.

Sie können die Routenrechner-Ressource verwenden, um [eine Route für bestimmte Parameter mithilfe des SDK oder der REST-API-Endpunkte zu berechnen](#). AWS Verwenden Sie diese Routenrechner-Ressource, um Routen zwischen einem Ursprung, einem Ziel und bis zu 23 Routenpunkten für verschiedene Transportarten, Abstände und Verkehrsbedingungen zu berechnen.

Sie können auch die Routenrechner-Ressource verwenden, um Eingaben für Ihre Routenplanungsalgorithmen oder [-produkte zu erstellen, indem Sie eine Routenmatrix berechnen](#). Berechnen Sie die Reisezeit und die Entfernung zwischen einer Reihe von Verlassenspositionen und einer Reihe von Zielpositionen. Routenplanungssoftware kann diese Zeit- und Entfernungsdaten verwenden, um eine Route oder eine Reihe von Routen zu optimieren, z. B. wenn Sie mehrere Zustellrouten planen und die beste Route und Zeit für jeden Stopp finden möchten. Sie können eine Matrix von Routen für verschiedene Arten von Transporten, Vermeidungen und Verkehrsbedingungen berechnen.

Note

Eine Übersicht über Routing-Konzepte finden Sie unter [Routen](#).

Themen

- [Voraussetzungen](#)
- [Berechnen einer Route](#)
- [Routenplanung mit einer Routenmatrix](#)
- [Positionen, die sich nicht auf einer Straßen befinden](#)
- [Zeit für die Warteschlange](#)

- [Reisemodus](#)
- [Verwalten Ihrer Routenrechner-Ressourcen](#)

Voraussetzungen

Bevor Sie mit der Berechnung von Routen beginnen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Routenrechner-Ressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Routenrechner-Ressource

Bevor Sie eine Route berechnen können, erstellen Sie eine Routenrechner-Ressource in Ihrem AWS Konto.

Wenn Sie eine Routenrechner-Ressource erstellen, können Sie aus den verfügbaren Datenanbietern wählen:

1. Esri – Weitere Informationen zur Abdeckung von Esri in Ihrer relevanten Region finden Sie unter [Esri-Details zu Straßennetzen und Verkehrsabdeckung](#).
2. HERE Technologies – Weitere Informationen zur Abdeckung von HERE in Ihrer relevanten Region finden Sie unter [HERE-Abdeckung für das Auto-Routing](#) und [HERE-Abdeckung für das Lkw-Routing](#).
3. Grab – Weitere Informationen zur Abdeckung von Grab finden Sie unter [Länder/Regionen und abgedecktes Gebiet](#).

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Sie können dies über die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs tun.

Console

So erstellen Sie eine Routenrechner-Ressource mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie Routenrechner erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Routenrechner-Ressource ein. Zum Beispiel *ExampleCalculator*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie für Datenanbieter einen [Datenanbieter](#) aus, der als Routenrechner verwendet werden soll.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag zu Ihrer neuen Routenrechner-Ressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. Wählen Sie Routenrechner erstellen aus.

API

So erstellen Sie eine Routenrechner-Ressource mithilfe der Amazon Location APIs

Verwenden Sie die [CreateRouteCalculator](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Routenrechner-Ressource mit dem Namen *ExampleCalculator* unter Verwendung des Datenanbieters *Esri* .

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
```

```
"DataSource": "Esri",
"Description": "string",
"Tags": {
  "Tag1" : "Value1"
}
}
```

AWS CLI

So erstellen Sie eine Routenrechner-Ressource mit -AWS CLIBefehlen

Verwenden Sie den `create-route-calculator`-Befehl.

Im folgenden Beispiel wird eine Routenrechner-Ressource namens *ExampleCalculator* mit *Esri* als Datenanbieter erstellt.

```
aws location \
  create-route-calculator \
  --calculator-name "ExampleCalculator" \
  --data-source "Esri" \
  --tags Tag1=Value1
```

Note

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer -AWSServices an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Routenrechner-Ressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, wählen Sie aus, wie Sie Ihre Anforderungen authentifizieren würden:

- Informationen dazu, wie Sie auf die Services zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie eine Website mit anonymen Benutzern haben, können Sie API-Schlüssel oder Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines API-Schlüssels für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon Location [JavaScript Authentifizierungshelfer](#).

```
import { LocationClient, CalculateRouteCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  CalculatorName: "ExampleCalculator",
  DeparturePosition: [-123.4567, 45.6789],
  DestinationPosition: [-123.123, 45.234],
};

const command = new CalculateRouteCommand(input);

const response = await client.send(command);
```

Berechnen einer Route

Sie können Amazon Location Service verwenden, um Routen zwischen einem Ursprung und einem Ziel mit bis zu 23 Routenpunkten entlang der Route für verschiedene Arten von Transporten, Abwägungen und Verkehrsbedingungen zu berechnen.

Note

Sie müssen zunächst eine Routenrechner-Ressource erstellen und die Authentifizierung für Ihre Anfragen an Amazon Location einrichten. Weitere Informationen finden Sie unter [Voraussetzungen](#).

Beginnen Sie mit der Berechnung von Routen

Senden Sie eine einfache Anfrage mithilfe der [-CalculateRoute](#) Operation. Eine einfache -Anforderung enthält die folgenden Pflichtfelder:

- **DeparturePosition** – Die Startposition, aus der die Route berechnet werden soll. Definiert als [longitude, latitude]
- **DestinationPosition** – Die Endposition, an der die Route berechnet werden soll. Definiert als [longitude, latitude].

Note

Wenn Sie eine Verlassen- oder Zielposition angeben, die sich nicht auf einer Straßen befindet, [verschiebt Amazon Location die Position auf die nächste Straßen](#).

Optional können Sie [in Ihrer Anfrage Wegpunkte](#), eine [Verlassenszeit](#) und einen [Reisemodus](#) angeben.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine `-CalculateRoute` Anforderung mit der Routenrechner-Ressource `ExampleCalculator`. Die Anforderung gibt die Berechnung einer Route von einer Ausgangsposition `[-122.7565, 49.0021]` zu einer Zielposition `[-122.3394, 47.6159]` an.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159]
}
```

AWS CLI

Das folgende Beispiel ist ein `calculate-route` Befehl, der die Routenrechner-Ressource verwendet `ExampleCalculator`. Die Anforderung gibt die Berechnung einer Route von einer Ausgangsposition `[-122.7565, 49.0021]` zu einer Zielposition `[-122.3394, 47.6159]` an.

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

Standardmäßig gibt die Antwort `Distance` in Kilometern zurück. Sie können die Maßeinheit mit dem folgenden optionalen Parameter in Fuß ändern:

- `DistanceUnit` – Gibt das Einheitensystem an, das für die Abstandsergebnisse verwendet werden soll.

Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

Festlegen von Wegpunkten

Bei der Berechnung einer Route können Sie mithilfe von Wegpunktpositionen bis zu 23 Zwischen-Stopover-Punkten zwischen der Ausgangsposition und der Zielposition angeben.

- `WaypointPositions` – Gibt eine geordnete Liste von Zwischenpositionen an, die entlang einer Route zwischen der Ausgangsposition und der Zielposition aufgenommen werden sollen.

Note

Wenn Sie eine Wegpunktposition angeben, die sich nicht auf einer Straße befindet, verschiebt Amazon Location die Position auf die nächste Straße.

Example

Die folgende [CalculateRoute](#) Anforderung berechnet eine Route mit 2 Wegpunkten:

- Die Startposition ist [-122.7565, 49.0021], und die Zielposition ist [-122.3394, 47.6159].
- Für den Anforderungsparameter `WaypointPositions`:
 - Der erste Stopp über Position ist [-122.1884, 48,0936].
 - Der zweite Stopp über Position ist [-122.3493, 47,6205].
- Setzen Sie den folgenden optionalen Parameter auf „*true*“, um die Geometrie des Trichter-Linestrings zwischen diesen beiden Wegpunkten einzuschließen:
 - `IncludeLegGeometry` – Schließt die Geometrie jedes Pfads zwischen einem Paar von Positionen in die Antwort ein.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ],
  "IncludeLegGeometry": true
}
```

Beispielantwort

Im Folgenden finden Sie eine Beispielanforderung mit der entsprechenden Antwort beim Aufrufen der [-CalculateRoute](#) Operation von der Amazon Location Routes API mit dem `IncludeLegGeometry` Wert „*true*“, der die Linestring-Geometrie jedes Pfads zwischen einem Paar von Positionen in der Antwort enthält.

Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "IncludeLegGeometry": true
}
```

Example response

```
{
  "Legs": [
    {
      "Distance": 178.5,
      "DurationSeconds": 6480,
      "EndPosition": [-122.3394,47.6159],
      "Geometry": {
        "LineString": [
          [-122.7565,49.0021],
          [-122.3394,47.6159]
        ]
      },
      "StartPosition": [-122.7565,49.0021],
      "Steps": [
        {
          "Distance": 178.5,
          "DurationSeconds": 6480,
          "EndPosition": [-122.3394,47.6159],
          "GeometryOffset": 0,
          "StartPosition": [-122.7565,49.0021]
        }
      ]
    }
  ],
  "Summary": {
    "DataSource": "Esri",
    "Distance": 178.5,
    "DistanceUnit": "Kilometers",
    "DurationSeconds": 6480,
    "RouteBBox": [
      -122.7565,49.0021,
      -122.3394,47.6159
    ]
  ]
}
```

Routenplanung mit einer Routenmatrix

Sie können Amazon Location Service verwenden, um Eingaben für Ihre Routenplanungs- und Optimierungssoftware zu erstellen. Sie können Routenergebnisse, einschließlich der Zeit und

Entfernung, für Routen zwischen einer Reihe von Abstiegspositionen und einer Reihe von Zielpositionen erstellen.

Angenommen, es handelt sich um die Abgangspositionen A und B sowie die Zielpositionen X und Y. Amazon Location Service gibt die Zeit und Entfernung für Routen von A zu X, A zu Y, B zu X und B zu Y zurück.

Sie können die Routen mit unterschiedlichen Transportarten, Abstammungen und Verkehrsbedingungen berechnen. Sie können beispielsweise angeben, dass das Fahrzeug ein Lkw mit einer Länge von 35 Fuß ist, und die berechnete Route verwendet diese Einschränkungen, um die Fahrtzeit und die Entfernung zu bestimmen.

Die Anzahl der zurückgegebenen Ergebnisse (und berechneten Routen) ist die Anzahl der Abstiegspositionen multipliziert mit der Anzahl der Zielpositionen. Ihnen wird jede berechnete Route in Rechnung gestellt, nicht jede Anforderung an den Service. Daher wird eine Routenmatrix mit 10 Abgängen und 10 Zielen als 100 Routen abgerechnet.

Berechnen einer Routenmatrix

Sie können eine Matrix von Routen zwischen einer Reihe von Abstiegspositionen und einer Reihe von Zielpositionen berechnen. Die Routenergebnisse beinhalten die Reisezeit und die Entfernung.

Voraussetzung

- Sie müssen zunächst eine Routenrechner-Ressource erstellen und die Authentifizierung für Ihre Anfragen an Amazon Location einrichten. Weitere Informationen finden Sie unter [Voraussetzungen](#).

Senden Sie eine -Anforderung mithilfe der [-CalculateRouteMatrix](#) Operation. Eine minimale Anforderung enthält die folgenden Pflichtfelder:

- `DeparturePositions` – Der Satz von Startpositionen, für die die Routen berechnet werden sollen. Definiert als Array von `[longitude, latitude]`
- `DestinationPositions` – Der Satz von Endpositionen, für die die Routen berechnet werden sollen. Definiert als Array von `[longitude, latitude]`.

Note

Wenn Sie eine Verlassen- oder Zielposition angeben, die sich nicht auf einer Straßen befindet, [verschiebt Amazon Location die Position auf die nächste Straßen](#).

Sie können optional eine [Ankunftszeit](#) und einen [Reisemodus](#) in Ihrer Anfrage angeben.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine `-CalculateRouteMatrix`Anforderung mit der Routenrechner-Ressource `ExampleCalculator`. Die Anforderung gibt die Berechnung der Matrix der Routen von den Abgangspositionen `[-122.7565, 49.0021]` und `[-122.2014, 47.6101]` zu den Zielpositionen `[-122.3394, 47.6159]` und `[-122.4813, 48.7511]` an.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565, 49.0021],
    [-122.2014, 47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813, 48.7511]
  ]
}
```

AWS CLI

Das folgende Beispiel ist ein `calculate-route-matrix` Befehl, der die Routenrechner-Ressource verwendet `ExampleCalculator`. Die Anforderung gibt die Berechnung der Matrix der Routen von den Abgangspositionen `[-122.7565, 49.0021]` und `[-122.2014, 47.6101]` zu den Zielpositionen `[-122.3394, 47.6159]` und `[-122.4813, 48.7511]` an.

```
aws location \
  calculate-route-matrix \
    --calculator-name ExampleCalculator \
    --departure-positions "[[-122.7565, 49.0021], [-122.2014, 47.6101]]" \
```

```
--destination-positions "[[-122.3394,47.6159],[-122.4813,48.7511]]"
```

Standardmäßig gibt die Antwort Distance in Kilometern zurück. Mit dem folgenden optionalen Parameter können Sie die Maßeinheit in Fuß ändern:

- `DistanceUnit` – Gibt das Einheitensystem an, das für die Abstandsergebnisse verwendet werden soll.

Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ],
  "DistanceUnit": "Miles"
}
```

Einschränkungen bei der Ankunft und bei Zielpositionen

Bei der Berechnung einer Routenmatrix gibt es Einschränkungen in Bezug auf die Ausgangs- und Zielpositionen. Diese Einschränkungen variieren je nach Anbieter, der von der `RouteCalculator` Ressource verwendet wird.

Einschränkung	Esri	Grab	Hier
Anzahl der Positionen	Bis zu 10 Ausgangspositionen und 10 Zielpositionen.	Bis zu 350 Ausgangspositionen und 350 Zielpositionen.	Bis zu 350 Ausgangspositionen und 350 Zielpositionen. Für längere Routen gelten zusätzliche Einschränkungen.

Einschränkung	Esri	Grab	Hier
			<p>Weitere Informationen finden Sie im Abschnitt.</p>
Abstand zwischen Positionen	<p>Jedes Paar von Verlassen- und Zielpositionen muss sich innerhalb von 400 km voneinander befinden (40 km für Walking-Routen).</p>		<p>Alle Eingangs- und Zielpositionen müssen innerhalb eines 180 km langen Trichterkreises liegen.</p> <p>Für längere Routen gelten zusätzliche Einschränkungen. Weitere Informationen finden Sie im Abschnitt.</p>
Routenlänge	<p>Routen werden nicht abgeschlossen, wenn die Gesamtreisezeit für die Route mehr als 400 Minuten beträgt.</p>		<p>Routen, die mehr als 10 km außerhalb eines Kreises um die Ausgangspunkte und Zielpunkte herum abweichen, werden nicht berechnet.</p> <p>Für längere Routen gelten zusätzliche Einschränkungen. Weitere Informationen finden Sie im Abschnitt.</p>

Einschränkung	Esri	Grab	Hier
Regionen	Die Berechnung einer Routenmatrix wird in Korea nicht unterstützt.	Verfügbar in Südostasien. Eine Liste der unterstützten Länder/Regionen und weitere Informationen finden Sie unter Länder/Regionen und abgedecktes Gebiet .	Keine zusätzlichen Einschränkungen.

Längere Routenplanung

Die Berechnung einer Matrix von Routenergebnissen ist für eine effiziente Routenplanung nützlich, aber die Berechnung kann einige Zeit in Anspruch nehmen. Alle Amazon Location Service-Datenanbieter setzen Einschränkungen für die Anzahl der Routen oder die Entfernung der Routen fest, die berechnet werden können. HERE ermöglicht beispielsweise das Erstellen von Routen zwischen 350 Eingangs- und Zielpositionen, diese Positionen müssen jedoch innerhalb eines 180-km-Bereichs liegen. Was ist, wenn Sie mit längeren Routen planen möchten?

Sie können eine Matrix von Routen mit uneingeschränkten Längen für eine geringere Anzahl von Routen berechnen, indem Sie einen `RouteCalculator` mit HERE als Datenanbieter verwenden. Dies ändert nichts daran, wie Sie die [CalculateRouteMatrix](#) API aufrufen. Amazon Location lässt einfach längere Routen zu, wenn Sie die Anforderungen erfüllen.

Die Anforderungen für Routenberechnungen mit längerer Länge sind:

- Der `RouteCalculator` muss den HERE-Datenanbieter verwenden.
- Die Anzahl der Abstiegspositionen darf nicht größer als 15 sein.
- Die Gesamtzahl der zu berechnenden Routen darf nicht größer als 100 sein.
- Langdistanz-Routing ist für Lkw-Routing mit Gebührenvermeidungen nicht zulässig, wenn die Routen größer als 1.000 km sind. Diese Kombination ist langsamer zu berechnen und kann dazu führen, dass der Aufruf eine Zeitüberschreitung hat. Sie können diese Routen mit der [-CalculateRoute](#) Operation einzeln berechnen.

Wenn Ihr Aufruf diese Anforderungen nicht erfüllt (z. B. wenn Sie 150 Routenberechnungen in einem einzigen Aufruf anfordern), `CalculateRouteMatrix` kehrt dazu zurück, nur die kürzeren

Routenregeln zuzulassen. Anschließend können Sie die Routen berechnen, solange sich die Positionen in einem 180-km-Kreis befinden.

Beachten Sie bei der Berechnung längerer Routen die folgenden Punkte:

- Längere Routen können länger dauern als die maximale Zeit für Amazon Location APIs . Wenn Sie häufige Timeouts mit bestimmten Routen erhalten, können Sie bei jedem Aufruf von eine geringere Anzahl von Routen ausprobieren `CalculateRouteMatrix`.
- Wenn Sie Ihrer `CalculateRouteMatrix` Anfrage weitere Ziel- oder Ausgangspositionen hinzufügen, kann der Vorgang in den eingeschränkteren Modus wechseln und Sie können eine Fehlermeldung für eine Route erhalten, die problemlos berechnet werden kann, wenn weniger Routen erstellt werden müssen. Reduzieren Sie in diesem Fall die Anzahl der Ziel- oder Ausgangspositionen und stellen Sie mehrere Anfragen, um den gesamten Satz von Routenberechnungen zu erhalten, die Sie benötigen.

Beispielantwort

Im Folgenden finden Sie eine Beispielanforderung mit der entsprechenden Antwort beim Aufrufen der [-CalculateRouteMatrix](#) Operation von der Amazon Location Routes API aus.

Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ]
}
```

Example response

```
{
  "RouteMatrix": [
    [
```



```

    {
      "Distance": 178.764,
      "DurationSeconds": 7565
    },
    {
      "Distance": 39.795,
      "DurationSeconds": 1955
    }
  ],
  [
    {
      "Distance": 15.31,
      "DurationSeconds": 1217
    },
    {
      "Distance": 142.506,
      "DurationSeconds": 6279
    }
  ]
],
"Summary": {
  "DataSource": "Here",
  "RouteCount": 4,
  "ErrorCount": 0,
  "DistanceUnit": "Kilometers"
}
}

```

Positionen, die sich nicht auf einer Straßen befinden

Wenn Sie `CalculateRoute` oder `verwendenCalculateRouteMatrix` verwenden, verschiebt Amazon Location die Position in eine nahegelegene Straße, wenn Sie eine Position für Hin- und Rückfahrt, Ziel oder Wegpunkt angeben, die sich nicht auf einer Straßen befindet.

Die folgende [CalculateRoute](#) Anforderung gibt eine Verlassen- und Zielposition an, die sich nicht auf einer Straßen befindet:

```

POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-123.128014, 49.298472],
  "DestinationPosition": [-123.134701, 49.294315]
}

```

```
}
```

Die resultierende Antwort gibt eine Position zurück, die auf eine nahegelegene Straßen eingefangen wurde:

```
{
  "Legs": [
    {
      "StartPosition": [-123.12815, 49.29717],
      "EndPosition": [-123.13375, 49.2926],
      "Distance": 4.223,
      "DurationSeconds": 697,
      "Steps": [
        {
          "StartPosition": [ -123.12815, 49.29717 ],
          "EndPosition": [ -123.12806, 49.29707 ],
          "Distance": 0.013,
          "DurationSeconds": 8
        },
        {
          "StartPosition": [ -123.12806, 49.29707 ],
          "EndPosition": [ -123.1288, 49.29659 ],
          "Distance": 0.082,
          "DurationSeconds": 36
        },
        {
          "StartPosition": [ -123.1288, 49.29659 ],
          "EndPosition": [ -123.12021, 49.29853 ],
          "Distance": 0.742,
          "DurationSeconds": 128
        },
        {
          "StartPosition": [ -123.12021, 49.29853 ],
          "EndPosition": [ -123.1201, 49.29959 ],
          "Distance": 0.131,
          "DurationSeconds": 26
        },
        {
          "StartPosition": [ -123.1201, 49.29959 ],
          "EndPosition": [ -123.13562, 49.30681 ],
          "Distance": 1.47,
          "DurationSeconds": 238
        },
      ],
    },
  ],
}
```

```

    {
      "StartPosition": [ -123.13562, 49.30681 ],
      "EndPosition": [ -123.13693, 49.30615 ],
      "Distance": 0.121,
      "DurationSeconds": 28
    },
    {
      "StartPosition": [ -123.13693, 49.30615 ],
      "EndPosition": [ -123.13598, 49.29755 ],
      "Distance": 0.97,
      "DurationSeconds": 156
    },
    {
      "StartPosition": [ -123.13598, 49.29755 ],
      "EndPosition": [ -123.13688, 49.29717 ],
      "Distance": 0.085,
      "DurationSeconds": 15
    },
    {
      "StartPosition": [ -123.13688, 49.29717 ],
      "EndPosition": [ -123.13375, 49.2926 ],
      "Distance": 0.609,
      "DurationSeconds": 62
    }
  ]
}
],
"Summary": {
  "RouteBBox": [ -123.13693, 49.2926, -123.1201, 49.30681 ],
  "DataSource": "Here",
  "Distance": 4.223,
  "DurationSeconds": 697,
  "DistanceUnit": "Kilometers"
}
}

```

Zeit für die Warteschlange

Wenn Sie `CalculateRoute` oder aufrufen `CalculateRouteMatrix` und in der Anforderung keine Abgangszeit angeben, spiegeln die berechneten Routen standardmäßig die optimalen Datenverkehrsbedingungen wider.

Sie können eine bestimmte Abgangszeit festlegen, um Live- und prädiktive Datenverkehrsbedingungen von Ihrem ausgewählten Datenanbieter zu verwenden, indem Sie eine der folgenden Optionen verwenden:

- `DepartNow` – Wenn der Wert auf „*true*“ gesetzt ist, werden Live-Datenverkehrsbedingungen verwendet, um den schnellsten Weg zu berechnen.
- `DepartureTime` – Wenn angegeben, werden prädiktive und bekannte Datenverkehrsbedingungen für die angeforderte Zeit verwendet. Definiert im folgenden [Format](#): `YYYY-MM-DDThh:mm:ss.sssZ`.

Example

Die folgende [CalculateRoute](#) Anfrage legt die Abreisezeit auf den 2. Juli 2024 um 12:15:20 Uhr UTC fest.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ]
  "IncludeLegGeometry": true,
  "DepartureTime": 2024-07-02T12:15:20.000Z,
}
```

Reisemodus

Sie können einen Reisemodus festlegen, wenn Sie `CalculateRoute` oder `CalculateRouteMatrix` verwenden. Die Art der Reise wirkt sich auf die Geschwindigkeit der Reise und die Straßenkompatibilität aus. Während der Standardreisemodus auf dem Auto basiert, können Sie mit dem folgenden optionalen Parameter angeben, welchen Reisemodus Sie während der Reise entlang einer Route verwenden:

- `TravelMode` – Gibt den Transportmodus bei der Berechnung einer Route an, z. B.: *Bicycle*, *Car*, *MotorcycleTruck*, oder *Walking*.

Einschränkungen

- Wenn Sie `Walking` für den Reisemodus angeben und Ihr Datenanbieter Esri ist, müssen Start und Ziel innerhalb von 40 km liegen.
- `Bicycle` oder `Motorcycle` sind nur verfügbar, wenn Grab als Datenanbieter verwendet wird.
- Grab bietet nur `-Bicycle` und `-Walking`Routen in bestimmten Städten. Weitere Informationen finden Sie unter [Länder/Regionen und abgedecktes Gebiet](#).
- `Truck` ist nicht verfügbar, wenn Grab als Datenanbieter verwendet wird.

Zusätzliche Einstellungen

Wenn Sie einen `TravelMode` von `Car` angeben, können Sie zusätzliche Routeneinstellungen mit dem folgenden optionalen Parameter angeben:

- `CarModeOptions` – Gibt Routeneinstellungen an, wenn Sie in einem Auto unterwegs sind, z. B. `AvoidFerries` oder `AvoidTolls`.

Wenn Sie einen `TravelMode` von `Truck` angeben, können Sie zusätzliche Routeneinstellungen mit dem folgenden optionalen Parameter angeben:

- `TruckModeOptions` – Gibt Routeneinstellungen bei Fahrten in einem Lkw an, z. B. `AvoidFerries` oder `AvoidTolls`, zusätzlich zur Angabe von Routen, die für die `TruckDimensions` und geeignet sind `TruckWeight`.

Example

Die folgende [CalculateRoute](#) Anforderung gibt `Truck` als Reisemodus an. Zu den zusätzlichen Routenbeschränkungen gehören: Vermeidung von Routen, die Seeflotten verwenden, und Vermeidung von Straßen, die die Lkw-Dimensionen und das Gewicht nicht berücksichtigen können.

```
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DepartNow": true,
  "TravelMode": "Truck",
  "TruckModeOptions": {
    "AvoidFerries": true,
    "AvoidTolls": false,
```

```
    "Dimensions": {
      "Height": 4.5,
      "Length": 15.5,
      "Unit": "Meters",
      "Width": 4.5
    },
    "Weight": {
      "Total": 7500,
      "Unit": "Pounds"
    }
  }
}
```

Verwalten Ihrer Routenrechner-Ressourcen

Sie können Ihre Routenrechner-Ressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten Ihrer Routenrechner-Ressourcen

Sie können Ihre Routenrechnerliste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der Routenrechner mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Zeigen Sie die Details des Routenrechners unter Meine Routenrechner an.

API

Verwenden Sie die [ListRouteCalculators](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Routenrechnern im AWS Konto.

```
POST /routes/v0/list-calculators
```

Im Folgenden finden Sie eine Beispielantwort für [ListRouteCalculators](#):

```
{
  "Entries": [
    {
      "CalculatorName": "ExampleCalculator",
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "DataSource": "Esri",
      "Description": "string",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den `list-route-calculators`-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste der Routenrechner im AWS Konto abzurufen.

```
aws location list-route-calculators
```

Abrufen von Routenrechnerdetails

Sie können Details zu jeder Routenrechner-Ressource in Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI, oder der Amazon Location APIs abrufen:

Console

So zeigen Sie die Details eines Routenrechners mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie unter Meine Routenrechner den Namenslink des Zielroutenrechners aus.

API

Verwenden Sie die [DescribeRouteCalculator](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Routenrechner-Details für *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeRouteCalculator](#):

```
{
  "CalculatorArn": "arn:aws:geo:us-west-2:123456789012:route-
calculator/ExampleCalculator",
  "CalculatorName": "ExampleCalculator",
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

Verwenden Sie den `describe-route-calculator`-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Routenrechner-Details für *ExampleCalculator* abzurufen.


```
aws location describe-route-calculator \
  --calculator-name "ExampleCalculator"
```

Löschen eines Routenrechners

Sie können einen Routenrechner aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen:

Console

So löschen Sie einen Routenrechner mithilfe der Amazon Location-Konsole

 Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie unter Meine Routenrechner den Zielroutenrechner aus.
4. Wählen Sie Routenrechner löschen aus.

API

Verwenden Sie die [DeleteRouteCalculator](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Geofence-Sammlung *ExampleCalculator*.

```
DELETE /routes/v0/calculators/ExampleCalculator
```

Im Folgenden finden Sie eine Beispielantwort für [DeleteRouteCalculator](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den `delete-route-calculator`-Befehl.

Das folgende Beispiel ist ein `-AWS CLI` Befehl zum Löschen der Geofence-Sammlung *ExampleCalculator*.

```
aws location delete-route-calculator \
```

```
--calculator-name "ExampleCalculator"
```

Geofencing eines Interessengebiets mithilfe von Amazon Location

Eine Geofencing-Anwendung bewertet die Position eines verfolgten Geräts im Verhältnis zu zuvor registrierten Interessengebieten. Auf diese Weise können auf der Grundlage von Positionsaktualisierungen Maßnahmen ergriffen werden. Sie können beispielsweise ein Ereignis initiieren, das eine Benachrichtigung auslöst, wenn sich ein Kunde, der in seiner mobilen App Kaffee bestellt hat, in der Nähe eines Geschäfts befindet.

Note

Einen Überblick über Geofencing- und Tracker-Konzepte finden Sie unter [Geofences und Tracker](#)

Dieser Abschnitt des Handbuchs enthält step-by-step Anweisungen zur Erstellung einer Geofencing-Anwendung mithilfe von Amazon Location Service.

Übersicht über die Schritte

1. Fügen Sie Geofences rund um Bereiche von Interesse hinzu und speichern Sie sie in einer Ressource zur Erfassung von Geofences.
2. Beginnen Sie mit der Verfolgung Ihrer Zielgeräte und speichern Sie den Gerätestandortverlauf in einer Tracker-Ressource.
3. Verknüpfen Sie Ihre Tracker-Ressource mit Ihrer Geofence-Erfassungsressource, sodass Aktualisierungen des Gerätestandorts automatisch mit all Ihren Geofences verglichen werden.
4. Sie können Gerätepositionen direkt mit Ihren Geofence-Erfassungsressourcen vergleichen, wenn Sie Amazon Location Trackers nicht verwenden möchten, um den Standortverlauf Ihrer Geräte zu speichern.

Nachdem Sie Ihre Geofencing-Lösung implementiert haben, sendet Ihre Ressource zur Erfassung von Geofences die folgenden Ereignisse aus:

- ENTER— Ein getracktes Gerät betritt einen Geofence innerhalb einer Geofence-Sammlung.
- EXIT— Ein getracktes Gerät verlässt einen Geofence innerhalb einer Geofence-Sammlung.

Sie können Amazon verwenden, um auf Ereignisse EventBridge zu reagieren, indem Sie sie an einen anderen Ort weiterleiten.

Als Alternative zum Senden von Updates über die Amazon Location Service APIs von jedem Gerät aus können Sie MQTT verwenden, um Geräteaktualisierungen zu senden.

In den folgenden Themen werden diese Schritte und Alternativen ausführlich beschrieben.

Themen

- [Fügen Sie Geofences hinzu](#)
- [Starten Sie die Nachverfolgung](#)
- [Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung](#)
- [Vergleichen Sie die Gerätepositionen anhand von Geofences](#)
- [Überprüfen Sie die Gerätepositionen](#)
- [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#)
- [Nachverfolgung mit AWS IoT und MQTT mit Amazon Location Service](#)
- [Verwalten Ihrer Geofence-Sammlungsressourcen](#)
- [Verwalten Ihrer Tracker-Ressourcen](#)
- [Beispiel für eine mobile Geofencing- und Tracking-Anwendung](#)

Fügen Sie Geofences hinzu

Geofences enthalten Punkte und Scheitelpunkte, die eine geschlossene Grenze bilden, die ein Interessengebiet definiert. Geofence-Sammlungen speichern und verwalten einen oder mehrere Geofences.

Amazon Location Geofence Collections speichert Geofences, die mithilfe eines standardmäßigen Geodatenformats namens [GeoJSON](#) (RFC 7946) definiert wurden. Sie können Tools wie [geojson.io](#) kostenlos verwenden, um Ihre Geofences grafisch zu zeichnen und die ausgegebene GeoJSON-Datei zu speichern.

Note

Amazon Location unterstützt keine Polygone mit Löchern, Multipolygonen, Polygonen im Uhrzeigersinn und Geofences, die den Antimeridian durchqueren.

Erstellen Sie eine Geofence-Sammlung

Erstellen Sie eine Geofence-Sammlung zum Speichern und Verwalten von Geofences mithilfe der Amazon Location-Konsole AWS CLI, der oder der Amazon Location APIs.

Console

So erstellen Sie eine Geofence-Sammlung mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location Service Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie Geofence-Sammlung erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name — Geben Sie einen eindeutigen Namen ein. Zum Beispiel *ExampleGeofenceCollection*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung — Geben Sie optional eine Beschreibung ein, um Ihre Ressourcen zu differenzieren.
5. Unter EventBridge CloudWatch Regel mit Ziel können Sie eine optionale EventBridge Regel erstellen, mit der Sie sofort auf [Geofence-Ereignisse reagieren](#) können. Dadurch kann Amazon Location Ereignisse in Amazon CloudWatch Logs veröffentlichen.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird Ihrer neuen Geofence-Sammlung ein Tag hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Amazon Location Service-Ressourcen](#).
7. (Optional) Unter Verschlüsselung mit vom Kunden verwaltetem Schlüssel können Sie wählen, ob Sie einen vom Kunden verwalteten Schlüssel hinzufügen möchten. Dadurch wird ein symmetrischer, vom Kunden verwalteter Schlüssel hinzugefügt, den Sie erstellen, besitzen und verwalten, statt der standardmäßigen AWS Verschlüsselung. Weitere Informationen finden Sie unter [Daten im Ruhezustand verschlüsseln](#).
8. Wählen Sie Geofence-Sammlung erstellen aus.

API

So erstellen Sie eine Geofence-Sammlung mithilfe der Amazon Location APIs

Verwenden Sie den [CreateGeofenceCollection](#) Vorgang über die Amazon Location Geofences-APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, um eine Geofence-Sammlung namens zu erstellen. *ExampleGeofenceCollection* Die Geofence-Erfassung ist mit einem vom [Kunden verwalteten AWS KMS Schlüssel verknüpft, um Kundendaten zu verschlüsseln](#).

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "ExampleGeofenceCollection",
  "Description": "Geofence collection 1 for shopping center",
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

Um eine Geofence-Sammlung mithilfe von Befehlen zu erstellen AWS CLI

Verwenden Sie den [create-geofence-collection](#)-Befehl.

Im folgenden Beispiel wird eine verwendet AWS CLI , um eine Geofence-Sammlung namens zu erstellen. *ExampleGeofenceCollection* Die Geofence-Erfassung ist mit einem vom [Kunden verwalteten AWS KMS Schlüssel zur Verschlüsselung von Kundendaten](#) verknüpft.

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab" \
  --tags Tag1=Value1
```

Note

Die Abrechnung hängt von Ihrer Nutzung ab. Für die Nutzung anderer AWS Dienste können Gebühren anfallen. Weitere Informationen finden Sie unter [Amazon Location Service — Preise](#).

Zeichne Geofences

Nachdem Sie Ihre Geofence-Sammlung erstellt haben, können Sie Ihre Geofences definieren. Geofences werden entweder als Polygon oder als Kreis definiert. [Um einen Polygon-Geofence zu zeichnen, können Sie ein GeoJSON-Bearbeitungswerkzeug wie `geojson.io` verwenden.](#)

Um einen Geofence als Kreis zu erstellen, müssen Sie den Mittelpunkt des Kreises und den Radius definieren. Wenn Sie beispielsweise einen Geofence erstellen möchten, um benachrichtigt zu werden, wenn sich ein Gerät in einem Umkreis von 50 Metern um einen bestimmten Standort befindet, würden Sie den Breiten- und Längengrad dieses Standorts verwenden und den Radius als 50 Meter angeben.

Mithilfe der Amazon Location Service APIs können Sie Ihrem Geofence auch Metadaten in Form von Schlüssel-Wert-Paaren hinzufügen. Diese können nützlich sein, um Informationen über den Geofence, wie seinen Typ, oder andere für Ihre Anwendung spezifische Informationen zu speichern. Sie können diese Metadaten verwenden, wenn. [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#)

Polygon-Geofences hinzufügen

In diesem Abschnitt wird die Erstellung von Polygon-Geofences beschrieben

Zeichnen Sie Geofences mit einem GeoJSON-Tool

[Nachdem Sie Ihre Geofence-Sammlung erstellt haben, können Sie Ihre Geofences mithilfe eines GeoJSON-Bearbeitungswerkzeugs wie `geojson.io` definieren.](#)

Um eine GeoJSON-Datei zu erstellen

1. Öffnen Sie ein GeoJSON-Bearbeitungswerkzeug. Zum Beispiel `geojson.io`.
2. Wählen Sie das Symbol Polygon zeichnen und zeichnen Sie Ihren Interessenbereich.
3. Wählen Sie Speichern und anschließend GeoJSON aus dem Drop-down-Menü aus.

Fügen Sie GeoJSON-Geofences in eine Geofence-Sammlung ein

Sie können die resultierende GeoJSON-Datei verwenden, um Ihre Geofences mithilfe der Amazon Location Service Service-Konsole AWS CLI, der oder der Amazon Location APIs hochzuladen:

Console

So fügen Sie mithilfe der Amazon Location Service Service-Konsole einen Geofence zu einer Geofence-Sammlung hinzu

1. Öffnen Sie die Amazon Location Service Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie in der Liste der Geofence-Sammlungen den Namenslink für die Geofence-Zielsammlung aus.
4. Wählen Sie unter Geofences die Option Geofences erstellen aus.
5. Ziehen Sie im Fenster Geofences hinzufügen Ihre GeoJSON-Datei per Drag & Drop in das Fenster.
6. Wählen Sie Geofences hinzufügen aus.

API

So fügen Sie Geofences mithilfe der Amazon Location APIs hinzu

Verwenden Sie den [PutGeofence](#) Vorgang über die Amazon Location Geofences-APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, um einer aufgerufenen Geofence-Sammlung einen Geofence mit der ID **GEOFENCE-EXAMPLE1** hinzuzufügen. *ExampleGeofenceCollection* Es gibt auch eine einzelne Geofence-Metadateneigenschaft mit dem Schlüssel und dem Wert an. Type loadingArea

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE1
Content-type: application/json

{
  "GeofenceProperties": {
    "Type" : "loadingArea"
  },
  "Geometry": {
```

```

    "Polygon": [
      [
        [-5.716667, -15.933333],
        [-14.416667, -7.933333],
        [-12.316667, -37.066667],
        [-5.716667, -15.933333]
      ]
    ]
  }
}

```

Alternativ können Sie mithilfe der Operation mehr als einen Geofence hinzufügen.

[BatchPutGeofence](#)

```

POST /geofencing/v0/collections/ExampleGeofenceCollection/put-geofences
Content-type: application/json

```

```

{
  "Entries": [
    {
      "GeofenceProperties": {
        "Type" : "loadingArea"
      },
      "GeofenceId": "GEOFENCE-EXAMPLE1",
      "Geometry": {
        "Polygon": [
          [
            [-5.716667, -15.933333],
            [-14.416667, -7.933333],
            [-12.316667, -37.066667],
            [-5.716667, -15.933333]
          ]
        ]
      }
    }
  ]
}

```

AWS CLI

Um einer Geofence-Sammlung mithilfe von Befehlen einen Geofence hinzuzufügen AWS CLI

Verwenden Sie den [put-geofence](#)-Befehl.

Im folgenden Beispiel wird ein verwendet, AWS CLI um einer Geofence-Sammlung einen Geofence hinzuzufügen, der aufgerufen wird. *ExampleGeofenceCollection*

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceTriangle \
    --geofence-properties '{"Type": "loadingArea"}' \
    --geometry 'Polygon=[[-5.716667, -15.933333],[-14.416667, -7.933333],
[-12.316667, -37.066667],[-5.716667, -15.933333]]]'
{
  "CreateTime": "2020-11-11T00:16:14.487000+00:00",
  "GeofenceId": "ExampleGeofenceTriangle",
  "UpdateTime": "2020-11-11T00:19:59.894000+00:00"
}
```

Hinzufügen von kreisförmigen Geofences

In diesem Abschnitt wird die Erstellung kreisförmiger Geofences beschrieben. Sie müssen den Breiten- und Längengrad des Punkts, der den Mittelpunkt des Kreises bilden soll, sowie den Radius des Kreises in Metern kennen. Sie können kreisförmige Geofences mit den Amazon Location APIs oder den erstellen. AWS CLI

API

Um kreisförmige Geofences mithilfe der Amazon Location APIs hinzuzufügen

Verwenden Sie den [PutGeofence](#) Vorgang über die Amazon Location Geofences-APIs.

Das folgende Beispiel verwendet eine API-Anfrage, um einen Geofence mit der ID GEOFENCE-EXAMPLE2 zu einer *Geofence-Sammlung* mit dem Namen hinzuzufügen: *ExampleGeofenceCollection*

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}
```

```

    }
  }
}

```

AWS CLI

Um einer Geofence-Sammlung mithilfe von Befehlen einen kreisförmigen Geofence hinzuzufügen
AWS CLI

Verwenden Sie den [put-geofence](#)-Befehl.

Im folgenden Beispiel wird ein verwendet, AWS CLI um einer Geofence-Sammlung einen Geofence hinzuzufügen, der aufgerufen wird. *ExampleGeofenceCollection*

```

$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'

```

Note

Sie können JSON für komplexe Geometrie auch in einer eigenen Datei ablegen, wie im folgenden Beispiel gezeigt.

```

$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry file:circle.json

```

Im Beispiel enthält die Datei circle.json JSON für die Kreisgeometrie.

```

{
  "Circle": {
    "Center": [-74.006975, 40.717127],
    "Radius": 287.7897969218057
  }
}

```

Starten Sie die Nachverfolgung

Dieser Abschnitt führt Sie durch die Erstellung einer Tracking-Anwendung, die Gerätestandorte erfasst.

Erstellen Sie einen Tracker

Erstellen Sie eine Tracker-Ressource, um Positionsaktualisierungen von Ihren Geräten zu speichern und zu verarbeiten. Sie können die Amazon Location Service Service-Konsole AWS CLI, die oder die Amazon Location APIs verwenden.

Jede in Ihren Tracker-Ressourcen gespeicherte Positionsaktualisierung kann ein Maß für die Positionsgenauigkeit und bis zu drei Felder mit Metadaten über die Position oder das Gerät enthalten, die Sie speichern möchten. Die Metadaten werden als Schlüssel-Wert-Paare gespeichert und können Informationen wie Geschwindigkeit, Richtung, Reifendruck oder Motortemperatur speichern.

Tracker filtern Positionsaktualisierungen, sobald sie empfangen werden. Dadurch wird das visuelle Rauschen in Ihren Gerätepfaden reduziert (als Jitter bezeichnet) und die Anzahl falscher Geofence-Eingangs- und Austrittsereignisse wird reduziert. Dies trägt auch zur Kostenkontrolle bei, da die Anzahl der eingeleiteten Geofence-Evaluierungen reduziert wird.

Tracker bieten drei Optionen zur Positionsfilterung, um die Kosten im Griff zu behalten und Störungen bei Ihren Standortaktualisierungen zu reduzieren.

- Genauigkeitsbasiert — Kann mit jedem Gerät verwendet werden, das Genauigkeitsmessungen ermöglicht. Die meisten Mobilgeräte stellen diese Informationen bereit. Die Genauigkeit jeder Positionsmessung wird von vielen Umweltfaktoren beeinflusst, darunter dem GPS-Satellitenempfang, der Landschaft und der Nähe von WLAN- und Bluetooth-Geräten. Die meisten Geräte, einschließlich der meisten Mobilgeräte, können zusammen mit der Messung eine Schätzung der Genauigkeit der Messung liefern. Bei der AccuracyBased Filterung ignoriert Amazon Location Standortaktualisierungen, wenn sich das Gerät weniger als die gemessene Genauigkeit bewegt hat. Wenn beispielsweise zwei aufeinanderfolgende Updates von einem Gerät einen Genauigkeitsbereich von 5 m und 10 m haben, ignoriert Amazon Location das zweite Update, wenn sich das Gerät weniger als 15 m bewegt hat. Amazon Location wertet ignorierte Updates weder anhand von Geofences aus, noch speichert sie.

Wenn keine Genauigkeit angegeben ist, wird sie als Null behandelt, und die Messung wird als absolut genau angesehen.

Note

Sie können auch genauigkeitsbasierte Filterung verwenden, um alle Filter zu entfernen. Wenn Sie die genauigkeitsbasierte Filterung auswählen, aber alle Genauigkeitsdaten auf Null überschreiben oder die Genauigkeit ganz weglassen, filtert Amazon Location keine Aktualisierungen heraus.

- **Entfernungsbasiert** — Verwenden Sie diese Option, wenn Ihre Geräte keine Genauigkeitsmessung bieten, Sie aber dennoch die Vorteile der Filterung nutzen möchten, um Jitter zu reduzieren und die Kosten zu senken. `DistanceBased` Beim Filtern werden Standortaktualisierungen ignoriert, bei denen sich Geräte weniger als 30 m (98,4 ft) bewegt haben. Wenn Sie die `DistanceBased` Positionsfilterung verwenden, wertet Amazon Location diese ignorierten Aktualisierungen weder anhand von Geofences aus, noch speichert Amazon Location die Aktualisierungen.

Die Genauigkeit der meisten Mobilgeräte, einschließlich der durchschnittlichen Genauigkeit von iOS- und Android-Geräten, liegt innerhalb von 15 m. In den meisten Anwendungen können durch `DistanceBased` Filterung die Auswirkungen von Standortungenauigkeiten bei der Anzeige der Geräteflugbahn auf einer Karte sowie die Abprallwirkung mehrerer aufeinanderfolgender Ein- und Austrittsereignisse verringert werden, wenn sich Geräte in der Nähe der Grenze eines Geofences befinden. Es kann auch dazu beitragen, die Kosten Ihrer Anwendung zu senken, da weniger Aufrufe zur Auswertung verknüpfter Geofences oder zum Abrufen von Gerätepositionen getätigt werden.

- **Zeitbasiert** — (Standard) Verwenden Sie diese Option, wenn Ihre Geräte sehr häufig Positionsaktualisierungen senden (mehr als einmal alle 30 Sekunden) und Sie Geofence-Auswertungen nahezu in Echtzeit durchführen möchten, ohne jedes Update zu speichern. Bei der `TimeBased` Filterung wird jede Standortaktualisierung anhand verknüpfter Geofence-Sammlungen bewertet, aber nicht jede Standortaktualisierung wird gespeichert. Wenn Ihre Aktualisierungsfrequenz häufiger als 30 Sekunden beträgt, wird nur ein Update pro 30 Sekunden für jede eindeutige Geräte-ID gespeichert.

Note

Denken Sie bei der Auswahl Ihrer Filtermethode und der Häufigkeit von Positionsaktualisierungen an die Kosten Ihrer Tracking-Anwendung. Ihnen wird jede Standortaktualisierung und einmal die Auswertung der Positionsaktualisierung anhand jeder verknüpften Geofence-Erfassung in Rechnung gestellt. Wenn Sie beispielsweise

zeitbasierte Filterung verwenden und Ihr Tracker mit zwei Geofence-Sammlungen verknüpft ist, zählt jede Positionsaktualisierung als eine Anfrage zur Standortaktualisierung und als zwei Auswertungen der Geofence-Erfassung. Wenn Sie Positionsaktualisierungen für Ihre Geräte alle 5 Sekunden melden und zeitbasierte Filterung verwenden, werden Ihnen 720 Standortaktualisierungen und 1.440 Geofence-Bewertungen pro Stunde für jedes Gerät in Rechnung gestellt.

Ihre Rechnung wird nicht von der Anzahl der Geofences in jeder Sammlung beeinflusst. Da jede Geofence-Sammlung bis zu 50.000 Geofences enthalten kann, sollten Sie Ihre Geofences nach Möglichkeit in weniger Sammlungen zusammenfassen, um die Kosten für Geofence-Bewertungen zu senken.

Standardmäßig erhalten Sie jedes Mal EventBridge Ereignisse, wenn ein getracktes Gerät einen verknüpften Geofence betritt oder verlässt. Weitere Informationen finden Sie unter [Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung](#).

Sie können Ereignisse für alle gefilterten Positionsaktualisierungen für eine Tracker-Ressource aktivieren. Weitere Informationen finden Sie unter [Aktivieren von Aktualisierungsereignissen für einen Tracker](#).

Note


Wenn Sie Ihre Daten mit Ihrem eigenen, vom AWS KMS Kunden verwalteten Schlüssel verschlüsseln möchten, ist die Funktion Bounding Polygon Queries standardmäßig deaktiviert. Dies liegt daran, dass durch die Verwendung dieser Funktion für Bounding Polygon Queries eine Darstellung Ihrer Gerätepositionen nicht mit dem von Ihnen verwalteten Schlüssel verschlüsselt wird. AWS KMS Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

Sie können sich für die Funktion Bounding Polygon Queries entscheiden, indem Sie den `KmsKeyEnableGeospatialQueries` Parameter bei der Erstellung oder Aktualisierung eines Trackers auf `true` setzen.

Console

So erstellen Sie einen Tracker mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location Service Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Trackers aus.
3. Wählen Sie Tracker erstellen.
4. Füllen Sie die folgenden Felder aus:
 - Name — Geben Sie einen eindeutigen Namen ein. Zum Beispiel *ExampleTracker*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung — Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie unter Positionsfilterung die Option aus, die am besten zu Ihrer geplanten Nutzung Ihrer Tracker-Ressource passt. Wenn Sie die Positionsfilterung nicht einrichten, ist die Standardeinstellung `TimeBased`. Weitere Informationen finden Sie [Tracker](#) in diesem Handbuch und [PositionFiltering](#) in der Amazon Location Service Trackers API-Referenz.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird Ihrer neuen Geofence-Sammlung ein Tag hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. (Optional) Unter Verschlüsselung mit vom Kunden verwaltetem Schlüssel können Sie wählen, ob Sie einen vom Kunden verwalteten Schlüssel hinzufügen möchten. Dadurch wird ein symmetrischer, vom Kunden verwalteter Schlüssel hinzugefügt, den Sie erstellen, besitzen und verwalten, statt der standardmäßigen AWS Verschlüsselung. Weitere Informationen finden Sie unter [Daten im Ruhezustand verschlüsseln](#).
8. (Optional) Unter können Sie wählen `KmsKeyEnableGeospatialQueries`, ob Geodatenabfragen aktiviert werden sollen. Auf diese Weise können Sie die Funktion `Bounding Polygon Queries` verwenden und gleichzeitig Ihre Daten mit einem von AWS KMS verwalteten Kundenschlüssel verschlüsseln.

 Note

Wenn Sie die Funktion `Bounding Polygon Queries` verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit dem von Ihnen verwalteten Schlüssel verschlüsselt. AWS KMS Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

9. (Optional) Unter EventBridge Konfiguration können Sie festlegen, ob EventBridge Ereignisse für gefilterte Positionsaktualisierungen aktiviert werden sollen. Dadurch wird jedes Mal ein Ereignis gesendet, wenn eine Positionsaktualisierung für ein Gerät in diesem Tracker der Bewertung der Positionsfilterung entspricht.
10. Wählen Sie Tracker erstellen.

API

So erstellen Sie einen Tracker mithilfe der Amazon Location APIs

Verwenden Sie den [CreateTracker](#) Vorgang über die Amazon Location Trackers APIs.

Das folgende Beispiel verwendet eine API-Anfrage, um einen Tracker namens *ExampleTracker* zu erstellen. Die Tracker-Ressource ist mit einem vom [Kunden verwalteten AWS KMS Schlüssel zur Verschlüsselung von Kundendaten](#) verknüpft und [ermöglicht keine Positionsaktualisierungen in EventBridge](#).

```
POST /tracking/v0/trackers
Content-type: application/json

{
  "TrackerName": "ExampleTracker",
  "Description": "string",
  "KmsKeyEnableGeospatialQueries": false,
  "EventBridgeEnabled": false,
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "PositionFiltering": "AccuracyBased",
  "Tags": {
    "string" : "string"
  }
}
```

Erstelle einen Tracker mit aktivierter Option **KmsKeyEnableGeospatialQueries**

Im folgenden Beispiel ist der Parameter auf `true` `KmsKeyEnableGeospatialQueries` gesetzt. Auf diese Weise können Sie die Funktion Bounding Polygon Queries verwenden und gleichzeitig Ihre Daten mit einem vom Kunden AWS KMS verwalteten Schlüssel verschlüsseln.

Informationen zur Verwendung der Funktion Bounding Polygon Queries finden Sie unter [???](#)

Note

Wenn Sie die Funktion Bounding Polygon Queries verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit dem von Ihnen verwalteten Schlüssel verschlüsselt. AWS KMS Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

```
POST /tracking/v0/trackers
Content-type: application/json

{
  "TrackerName": "ExampleTracker",
  "Description": "string",
  "KmsKeyEnableGeospatialQueries": true,
  "EventBridgeEnabled": false,
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "PositionFiltering": "AccuracyBased",
  "Tags": {
    "string" : "string"
  }
}
```

AWS CLI

Um einen Tracker mithilfe von AWS CLI Befehlen zu erstellen

Verwenden Sie den [create-tracker](#)-Befehl.

Das folgende Beispiel verwendet AWS CLI , um einen Tracker namens zu erstellen *ExampleTracker*. Die Tracker-Ressource ist mit einem vom [Kunden verwalteten AWS KMS Schlüssel zur Verschlüsselung von Kundendaten](#) verknüpft und [ermöglicht keine Positionsaktualisierungen in EventBridge](#).

```
aws location \
  create-tracker \
  --tracker-name "ExampleTracker" \
  --position-filtering "AccuracyBased" \
  --event-bridge-enabled false \
  --kms-key-enable-geospatial-queries false \
```



```
--kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Erstelle einen Tracker mit aktivierter Option **KmsKeyEnableGeospatialQueries**

Im folgenden Beispiel ist der Parameter auf `true` `KmsKeyEnableGeospatialQueries` gesetzt. Auf diese Weise können Sie die Funktion `Bounding Polygon Queries` verwenden und gleichzeitig Ihre Daten mit einem vom Kunden AWS KMS verwalteten Schlüssel verschlüsseln.

Informationen zur Verwendung der Funktion `Bounding Polygon Queries` finden Sie unter [???](#)

Note

Wenn Sie die Funktion `Bounding Polygon Queries` verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit dem von Ihnen verwalteten Schlüssel verschlüsselt. AWS KMS Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

```
aws location \  
  create-tracker \  
    --tracker-name "ExampleTracker" \  
    --position-filtering "AccuracyBased" \  
    --event-bridge-enabled false \  
    --kms-key-enable-geospatial-queries true \  
    --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Note

Die Abrechnung hängt von Ihrer Nutzung ab. Für die Nutzung anderer AWS Dienste können Gebühren anfallen. Weitere Informationen finden Sie unter [Amazon Location Service — Preise](#).

Sie können die Beschreibung, die Positionsfilterung und die EventBridge Konfiguration bearbeiten, nachdem der Tracker erstellt wurde, indem Sie `Tracker bearbeiten` wählen.

Authentifizieren Sie Ihre Anfragen

Sobald Sie eine Tracker-Ressource erstellt haben und bereit sind, Gerätepositionen anhand von Geofences zu bewerten, wählen Sie aus, wie Sie Ihre Anfragen authentifizieren möchten:

- Informationen darüber, wie Sie auf die Dienste zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie Gerätepositionen mit nicht authentifizierten Anfragen veröffentlichen möchten, sollten Sie Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines Amazon Cognito Cognito-Identitätspools für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon-Standort [JavaScript Authentifizierungshelfer](#).

```
import { LocationClient, BatchUpdateDevicePositionCommand } from "@aws-sdk/client-location";
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

// Unauthenticated identity pool you created
const identityPoolId = "us-east-1:1234abcd-5678-9012-abcd-sample-id";

// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "us-east-1", // The region containing both the identity pool and tracker resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make requests to Amazon Location
});

const input = {
  TrackerName: "ExampleTracker",
  Updates: [
    {
      DeviceId: "ExampleDevice-1",
      Position: [-123.4567, 45.6789],
      SampleTime: new Date("2020-10-02T19:09:07.327Z"),
    },
    {
```

```
    DeviceId: "ExampleDevice-2",
    Position: [-123.123, 45.123],
    SampleTime: new Date("2020-10-02T19:10:32Z"),
  },
],
};

const command = new BatchUpdateDevicePositionCommand(input);

// Send device position updates
const response = await client.send(command);
```

Aktualisieren Sie Ihren Tracker mit einer Geräteposition

Um deine Geräte zu verfolgen, kannst du Aktualisierungen der Geräteposition auf deinem Tracker posten. Sie können diese Gerätepositionen oder den Verlauf der Gerätepositionen später von Ihrer Tracker-Ressource abrufen.

Jede Positionsaktualisierung muss die Geräte-ID, einen Zeitstempel und eine Position enthalten. Sie können optional weitere Metadaten hinzufügen, darunter Genauigkeit und bis zu 3 Schlüssel-Wert-Paare für Ihren eigenen Gebrauch.

Wenn Ihr Tracker mit einer oder mehreren Geofence-Sammlungen verknüpft ist, werden Aktualisierungen anhand dieser Geofences bewertet (gemäß den Filterregeln, die Sie für den Tracker angegeben haben). Wenn ein Gerät in einen abgegrenzten Bereich eindringt (indem es sich von innerhalb des Bereichs nach außen bewegt oder umgekehrt), erhalten Sie Ereignisse in. EventBridge Zu diesen ENTER EXIT Ereignissen gehören Informationen zur Positionsaktualisierung, einschließlich der Geräte-ID, des Zeitstempels und aller zugehörigen Metadaten.

Note

Weitere Informationen zur Positionsfilterung finden Sie unter [Erstellen Sie einen Tracker](#). Weitere Informationen zu Geofence-Ereignissen finden Sie unter. [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#)

Verwenden Sie eine der folgenden Methoden, um Geräteaktualisierungen zu senden:

- [Senden Sie MQTT-Updates](#) an eine AWS IoT Core-Ressource und verknüpfen Sie sie mit Ihrer Tracker-Ressource.

- Senden Sie Standortaktualisierungen mithilfe der Amazon Location Trackers API, mithilfe der AWS CLI, oder der Amazon Location APIs. Sie können die [AWS SDKs](#) verwenden, um die APIs von Ihrer iOS- oder Android-Anwendung aus aufzurufen.

API

Um eine Positionsaktualisierung mithilfe der Amazon Location APIs zu senden

Verwenden Sie den [BatchUpdateDevicePosition](#) Vorgang über die Amazon Location Trackers APIs.

Das folgende Beispiel verwendet eine API-Anfrage, um eine Aktualisierung der Geräteposition für einen Tracker *ExampleDeviceExampleTracker* zu posten.

```
POST /tracking/v0/trackers/ExampleTracker/positions
Content-type: application/json
{
  "Updates": [
    {
      "DeviceId": "1",
      "Position": [
        -123.12245146162303, 49.27521118043802
      ],
      "SampleTime": "2022-10-24T19:09:07.327Z",
      "PositionProperties": {
        "name" : "device1"
      },
      "Accuracy": {
        "Horizontal": 10
      }
    },
    {
      "DeviceId": "2",
      "Position": [
        -123.1230104928471, 49.27752402723152
      ],
      "SampleTime": "2022-10-02T19:09:07.327Z"
    },
    {
      "DeviceId": "3",
      "Position": [
```

```

-123.12325592118916, 49.27340530543111
],
"SampleTime": "2022-10-02T19:09:07.327Z"
},
{
"DeviceId": "4",
"Position": [
-123.11958813096311, 49.27774641063121
],
"SampleTime": "2022-10-02T19:09:07.327Z"
},
{
"DeviceId": "5",
"Position": [
-123.1277418058896, 49.2765989015285
],
"SampleTime": "2022-10-02T19:09:07.327Z"
},
{
"DeviceId": "6",
"Position": [
-123.11964267059481, 49.274188155916534
],
"SampleTime": "2022-10-02T19:09:07.327Z"
}
]
}

```

AWS CLI

Um eine Positionsaktualisierung mithilfe von AWS CLI Befehlen zu senden

Verwenden Sie den [batch-update-device-position](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI , um eine Aktualisierung der Geräteposition für *ExampleDevice-1* und *ExampleDevice-2* an einen Tracker zu senden *ExampleTracker*.

```

aws location batch-update-device-position \
--tracker-name ExampleTracker \
--updates '[{"DeviceId":"ExampleDevice-1","Position":
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z"},
{"DeviceId":"ExampleDevice-2","Position":

```

```
[ -123.123, 47.123 ], "SampleTime": "2021-11-30T21:47:25.149Z", "Accuracy":
{"Horizontal": 10.30 }, "PositionProperties": { "field1": "value1", "field2": "value2" } ] ]'
```

Ruft den Standortverlauf eines Geräts von einem Tracker ab

Ihre Amazon Location Tracker-Ressource speichert den Standortverlauf all Ihrer verfolgten Geräte für einen Zeitraum von 30 Tagen. Sie können den Standortverlauf Ihres Geräts, einschließlich aller zugehörigen Metadaten, von Ihrer Tracker-Ressource abrufen. In den folgenden Beispielen werden AWS CLI die oder die Amazon Location APIs verwendet.

API

Um den Gerätestandortverlauf von einem Tracker mithilfe der Amazon Location APIs abzurufen

Verwenden Sie den [GetDevicePositionHistory](#) Vorgang über die Amazon Location Trackers APIs.

Das folgende Beispiel verwendet eine API-URI-Anforderung, um den Gerätestandortverlauf *ExampleDevice* von einem Tracker abzurufen, *ExampleTracker* der von 19:05:07 (einschließlich) aufgerufen wird und am 19:20:07 (exklusiv) endet. 2020-10-02

```
POST /tracking/v0/trackers/ExampleTracker/devices/ExampleDevice/list-positions
Content-type: application/json
{
  "StartTimeInclusive": "2020-10-02T19:05:07.327Z",
  "EndTimeExclusive": "2020-10-02T19:20:07.327Z"
}
```

AWS CLI

Um den Gerätestandortverlauf mithilfe von AWS CLI Befehlen von einem Tracker abzurufen

Verwenden Sie den [get-device-position-history](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI , um den Gerätestandortverlauf *ExampleDevice* von einem Tracker abzurufen, der von 19:05:07 (einschließlich) *ExampleTracker* beginnt und bei 19:20:07 (exklusiv) endet 2020-10-02.

```
aws location \
```

```
get-device-position-history \  
  --device-id "ExampleDevice" \  
  --start-time-inclusive "2020-10-02T19:05:07.327Z" \  
  --end-time-exclusive "2020-10-02T19:20:07.327Z" \  
  --tracker-name "ExampleTracker"
```

Listet die Positionen eurer Geräte auf

Sie können eine Liste der Gerätepositionen für einen Tracker mithilfe der AWS CLI oder der Amazon Location APIs mit der ListDevicePositions API anzeigen. Wenn Sie die ListDevicePositions API aufrufen, wird eine Liste der neuesten Positionen für alle Geräte zurückgegeben, die mit einem bestimmten Tracker verknüpft sind. Standardmäßig gibt diese API 100 der neuesten Gerätepositionen pro Ergebnisseite für einen bestimmten Tracker zurück. Um nur Geräte innerhalb einer bestimmten Region zurückzugeben, verwenden Sie den FilterGeometry Parameter, um eine Bounding Polygon Query zu erstellen. Auf diese Weise werden beim Aufrufen nur Geräte ListDevicePositions zurückgegeben, die sich innerhalb des Polygons befinden.

Note

Wenn Sie Ihre Daten mit Ihrem eigenen, vom AWS KMS Kunden verwalteten Schlüssel verschlüsseln möchten, ist die Funktion Bounding Polygon Queries standardmäßig deaktiviert. Dies liegt daran, dass durch die Verwendung dieser Funktion eine Darstellung Ihrer Gerätepositionen nicht mit Ihrem AWS KMS verwalteten Schlüssel verschlüsselt wird. Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

Sie können sich für die Funktion Bounding Polygon Queries entscheiden. Dazu setzen Sie den KmsKeyEnableGeospatialQueries Parameter bei der Erstellung oder Aktualisierung eines Trackers auf true.

API

Verwenden Sie den [ListDevicePositions](#) Vorgang über die Amazon Location Trackers APIs.

Das folgende Beispiel ist eine API-Anfrage zum Abrufen einer Liste von Gerätepositionen im polygonalen Bereich unter Verwendung des optionalen Parameters. [FilterGeometry](#) Das Beispiel gibt 3 Gerätestandorte zurück, die in dem durch das Polygon Array definierten Bereich vorhanden sind.

```
POST /tracking/v0/trackers/TrackerName/list-positions HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "FilterGeometry": {
    "Polygon": [
      [
        [
          -123.12003339442259,
          49.27425121147397
        ],
        [
          -123.1176984148229,
          49.277063620879744
        ],
        [
          -123.12389509145294,
          49.277954183760926
        ],
        [
          -123.12755921328647,
          49.27554025235713
        ],
        [
          -123.12330236586217,
          49.27211836076236
        ],
        [
          -123.12003339442259,
          49.27425121147397
        ]
      ]
    ]
  },
  "MaxResults": 3,
  "NextToken": "1234-5678-9012"
}
```

Im Folgenden finden Sie eine Beispielantwort für [ListDevicePositions](#):

```
{
  "Entries": [
    {
```



```
    "DeviceId": "1",
    "SampleTime": "2022-10-24T19:09:07.327Z",
    "Position": [
      -123.12245146162303,
      49.27521118043802
    ],
    "Accuracy": {
      "Horizontal": 10
    },
    "PositionProperties": {
      "name": "device1"
    }
  },
  {
    "DeviceId": "3",
    "SampleTime": "2022-10-02T19:09:07.327Z",
    "Position": [
      -123.12325592118916,
      49.27340530543111
    ]
  },
  {
    "DeviceId": "2",
    "SampleTime": "2022-10-02T19:09:07.327Z",
    "Position": [
      -123.1230104928471,
      49.27752402723152
    ]
  }
],
"NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-trackers](#)-Befehl.

Das folgende Beispiel dient AWS CLI dazu, eine Liste von Geräten in einem polygonalen Gebiet abzurufen.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung

Nachdem Sie nun über eine Geofence-Sammlung und einen Tracker verfügen, können Sie sie miteinander verknüpfen, sodass Standortaktualisierungen automatisch anhand all Ihrer Geofences ausgewertet werden. Wenn Sie nicht alle Standortaktualisierungen auswerten möchten oder wenn Sie einige Ihrer Standorte nicht in einer Tracker-Ressource speichern, können Sie bei Bedarf [Gerätepositionen mit Geofences vergleichen](#).

Wenn Gerätepositionen anhand von Geofences bewertet werden, werden Ereignisse generiert. Sie können eine Aktion für diese Ereignisse festlegen. Weitere Informationen zu Aktionen, die Sie für Geofence-Ereignisse festlegen können, finden Sie unter [Reagieren auf Amazon Location Service Service-Ereignisse mit Amazon EventBridge](#).

Ein Amazon-Standortereignis umfasst die Attribute der Gerätepositions-Aktualisierung, die es generiert, und einige Attribute des Geofence-Bereichs, der eingegeben oder verlassen wird. Weitere Informationen zu den in einem Geofence-Ereignis enthaltenen Daten finden Sie unter [Amazon-EventBridge Ereignisbeispiele für Amazon Location Service](#).

In den folgenden Beispielen wird eine Tracker-Ressource mithilfe der Konsole, der oder der Amazon Location APIs mit einer Geofence-Sammlung verknüpft. AWS CLI

Console

So verknüpfen Sie mithilfe der Amazon Location Service Service-Konsole eine Tracker-Ressource mit einer Geofence-Sammlung

1. Öffnen Sie die Amazon Location Service Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Trackers aus.
3. Wählen Sie unter Geräte-Tracker den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen die Option Geofence-Sammlung verknüpfen aus.
5. Wählen Sie im Fenster Verknüpfte Geofence-Sammlung eine Geofence-Sammlung aus dem Drop-down-Menü aus.
6. Wählen Sie Verknüpfen.

Nachdem Sie die Tracker-Ressource verknüpft haben, wird ihr der Status Aktiv zugewiesen.

API

Um eine Tracker-Ressource mithilfe der Amazon Location APIs mit einer Geofence-Sammlung zu verknüpfen

Verwenden Sie den [AssociateTrackerConsumer](#) Vorgang über die Amazon Location Trackers APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, die mithilfe ihres [Amazon-Ressourcennamens](#) (ARN) *ExampleTracker* mit einer Geofence-Sammlung verknüpft wird.

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json

{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

AWS CLI

Um eine Tracker-Ressource mithilfe von Befehlen mit einer Geofence-Sammlung zu verknüpfen
AWS CLI

Verwenden Sie den [associate-tracker-consumer](#)-Befehl.

Im folgenden Beispiel wird eine AWS CLI zur Erstellung einer Geofence-Sammlung namens verwendet. *ExampleGeofenceCollection*

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
    --tracker-name "ExampleTracker"
```

Vergleichen Sie die Gerätepositionen anhand von Geofences

Es gibt zwei Möglichkeiten, Positionen anhand von Geofences abzugleichen, um Geofence-Ereignisse zu generieren:

- Sie können Tracker und Geofence-Sammlungen verknüpfen. Weitere Informationen finden Sie im Abschnitt: [Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung](#)
- Mithilfe der API können Sie eine direkte Anfrage an die Ressource zur Erfassung von Geofence stellen, um eine oder mehrere Positionen zu bewerten. [BatchEvaluateGeofences](#)

Darüber hinaus können Sie eingehende Geofence-Ereignisse für ein Gerät prognostizieren, das einen Geofence betritt, verlässt oder innerhalb eines Geofence-Bereichs inaktiv bleibt. Verwenden Sie die API, um Ereignisse vorherzusagen [ForecastGeofenceEvents](#).

Wenn Sie auch den Standortverlauf Ihres Geräts verfolgen oder Standorte auf einer Karte anzeigen möchten, verknüpfen Sie den Tracker mit einer Geofence-Erfassung. Alternativ möchten Sie möglicherweise nicht alle Standortaktualisierungen auswerten, oder Sie beabsichtigen nicht, Standortdaten in einer Tracker-Ressource zu speichern. Wenn einer der beiden Fälle zutrifft, können Sie eine direkte Anfrage an die Geofence-Erfassung richten und eine oder mehrere Gerätepositionen anhand der zugehörigen Geofences auswerten.

Wenn Gerätepositionen anhand von Geofences bewertet werden, werden Ereignisse generiert. Sie können auf diese Ereignisse reagieren und sie an andere AWS Dienste weiterleiten. Weitere Informationen zu Aktionen, die Sie beim Empfang von Geofence-Ereignissen ergreifen können, finden Sie unter [Reagieren auf Amazon Location Service Service-Ereignisse mit Amazon EventBridge](#)

Ein Amazon-Standortereignis umfasst die Attribute der Gerätepositions-Aktualisierung, mit der es generiert wird, einschließlich Uhrzeit, Position, Genauigkeit und Schlüsselwertmetadaten sowie einige Attribute des Geofence, der eingegeben oder verlassen wird. Weitere Informationen zu den in einem Geofence-Ereignis enthaltenen Daten finden Sie unter: [Amazon- EventBridge Ereignisbeispiele für Amazon Location Service](#)

In den folgenden Beispielen werden AWS CLI die oder die Amazon Location APIs verwendet.

API

Um Gerätepositionen mithilfe der Amazon Location APIs mit der Position von Geofences zu vergleichen

Verwenden Sie den [BatchEvaluateGeofences](#) Vorgang über die Amazon Location Geofences-APIs.

Das folgende Beispiel verwendet eine API-Anfrage, um die Position des Geräts anhand einer zugehörigen *ExampleDevice* Geofence-Erfassung auszuwerten.

ExampleGeofenceCollection Ersetzen Sie diese Werte durch Ihre eigenen Geofence- und Geräte-IDs.

```
POST /geofencing/v0/collections/ExampleGeofenceCollection/positions HTTP/1.1
Content-type: application/json

{
  "DevicePositionUpdates": [
    {
      "DeviceId": "ExampleDevice",
      "Position": [-123.123, 47.123],
      "SampleTime": "2021-11-30T21:47:25.149Z",
      "Accuracy": {
        "Horizontal": 10.30
      },
      "PositionProperties": {
        "field1": "value1",
        "field2": "value2"
      }
    }
  ]
}
```

AWS CLI

Um Gerätepositionen mithilfe von Befehlen mit der Position von Geofences abzugleichen AWS CLI

Verwenden Sie den [batch-evaluate-geofences](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI , um die Position von *ExampleDevice* anhand einer zugehörigen Geofence-Sammlung auszuwerten.

ExampleGeofenceCollection Ersetzen Sie diese Werte durch Ihre eigenen Geofence- und Geräte-IDs.

```
aws location \
  batch-evaluate-geofences \
    --collection-name ExampleGeofenceCollection \
    --device-position-updates '[{"DeviceId": "ExampleDevice", "Position":
[-123.123,47.123], "SampleTime": "2021-11-30T21:47:25.149Z", "Accuracy":
{"Horizontal":10.30}, "PositionProperties": {"field1": "value1", "field2": "value2"}}]'
```

Wenn Gerätepositionen anhand von Geofences bewertet werden, werden Ereignisse generiert. Traditionell können Sie auf Ereignisse reagieren [Amazon EventBridge](#), indem Sie Folgendes verwenden. Mit diesem Verfahren können Sie jedoch erst auf Ereignisse reagieren, nachdem sie eingetreten sind. Wenn Sie voraussehen müssen, wann ein Gerät einen Geofence betritt oder verlässt, z. B. wenn ein Gerät eine Grenze überschreitet und infolgedessen anderen Vorschriften unterliegt, können Sie die [ForecastGeofenceEvents](#)API verwenden, um future Geofence-Ereignisse vorherzusagen.

Die [ForecastGeofenceEvents](#)API verwendet Kriterien wie die Nähe time-to-breach, Geschwindigkeit und Position des Geräts, um Ereignisse vorherzusagen. Dort gibt die API eine zurück `ForecastedBreachTime`, die die geschätzte Zeit angibt, zu der das Geofence-Ereignis eintreten wird.

Das folgende Beispiel verwendet die Amazon Location APIs.

API

Um Geofence-Ereignisse mithilfe der Amazon Location APIs zu prognostizieren

Verwenden Sie den [ForecastGeofenceEvents](#) Vorgang über die Amazon Location Geofences-APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, um Geofence-Ereignisse für einen *ExampleDevice* Verwandten eines zu prognostizieren. *ExampleGeofence* Ersetzen Sie diese Werte durch Ihre eigenen Geofence- und Geräte-IDs.

```
POST /geofencing/v0/collections/CollectionName/forecast-geofence-events HTTP/1.1
Content-type: application/json

{
  "DeviceState": {
    "Position": [ number ],
    "Speed": number
  },
  "DistanceUnit": "string",
  "MaxResults": number,
  "NextToken": "string",
  "SpeedUnit": "string",
  "TimeHorizonMinutes": number
}
```

Überprüfen Sie die Gerätepositionen

Verwenden Sie die [VerifyDevicePosition](#) API, um die Integrität einer Geräteposition zu überprüfen. Diese API gibt Informationen über die Integrität der Geräteposition zurück, indem sie Eigenschaften wie das Mobilfunksignal des Geräts, den Wi-Fi-Zugangspunkt, die IPv4-Adresse und die Verwendung eines Proxys auswertet.

Voraussetzungen

Bevor Sie die aufgelisteten APIs zur Geräteverifizierung verwenden können, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Sie haben einen Tracker für das Gerät oder die Geräte erstellt, die Sie überprüfen möchten. Weitere Informationen finden Sie unter [Starten Sie die Nachverfolgung](#).

Das folgende Beispiel zeigt eine Anfrage für die Amazon Location [VerifyDevicePosition](#) API.

API

Um Gerätepositionen mithilfe der Amazon Location APIs zu verifizieren

Verwenden Sie den [VerifyDevicePosition](#) Vorgang über die Amazon Location Tracking APIs.

Das folgende Beispiel zeigt eine API-Anfrage zur Bewertung der Integrität der Position eines Geräts. Ersetzen Sie diese Werte durch Ihre eigenen Geräte-IDs.

```
POST /tracking/v0/trackers/TrackerName/positions/verify HTTP/1.1
Content-type: application/json

{
  "DeviceState": {
    "Accuracy": {
      "Horizontal": number
    },
    "CellSignals": {
      "LteCellDetails": [
        {
          "CellId": number,
          "LocalId": {
            "Earfcn": number,
            "Pci": number
          }
        }
      ]
    }
  }
}
```

```
    },
    "Mcc": number,
    "Mnc": number,
    "NetworkMeasurements": [
      {
        "CellId": number,
        "Earfcn": number,
        "Pci": number,
        "Rsrp": number,
        "Rsrq": number
      }
    ],
    "NrCapable": boolean,
    "Rsrp": number,
    "Rsrq": number,
    "Tac": number,
    "TimingAdvance": number
  }
]
},
"DeviceId": "ExampleDevice",
"Ipv4Address": "string",
"Position": [ number ],
"SampleTime": "string",
"WiFiAccessPoints": [
  {
    "MacAddress": "string",
    "Rss": number
  }
]
},
"DistanceUnit": "string"
}
```

Note

Das Location Integrity SDK bietet erweiterte Funktionen im Zusammenhang mit der Geräteverifizierung und kann auf Anfrage verwendet werden. Um Zugriff auf das SDK zu erhalten, wenden Sie sich an den [Vertriebssupport](#).

Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge

Amazon EventBridge ist ein Serverless Event Bus, der Anwendungen mithilfe von Daten aus - AWS Services wie Amazon Location. EventBridge receives Events von Amazon Location effizient miteinander verbindet und diese Daten an Ziele wie weiterleitet AWS Lambda. Sie können Routing-Regeln einrichten, um zu bestimmen, wohin Ihre Daten gesendet werden sollen, um Anwendungsarchitekturen zu erstellen, die in Echtzeit reagieren.

Standardmäßig werden nur Geofence-Ereignisse (ENTER- und -EXIT Ereignisse, wenn Geräte in die gefensterten Bereiche gelangen oder diese verlassen) an gesendet EventBridge . Sie können auch alle gefilterten Positionsaktualisierungsereignisse für eine Tracker-Ressource aktivieren. Weitere Informationen finden Sie unter [Aktivieren von Aktualisierungsereignissen für einen Tracker](#).

Weitere Informationen finden Sie [unter Ereignisse und Ereignismuster](#) im Amazon- EventBridge Benutzerhandbuch.

Themen

- [Aktivieren von Aktualisierungsereignissen für einen Tracker](#)
- [Erstellen von Ereignisregeln für Amazon Location](#)
- [Amazon- EventBridge Ereignisbeispiele für Amazon Location Service](#)

Aktivieren von Aktualisierungsereignissen für einen Tracker

Standardmäßig sendet Amazon Location nur - ENTER und EXIT Geofence-Ereignisse an EventBridge. Sie können aktivieren, dass alle gefilterten UPDATE Positionereignisse für einen Tracker an gesendet werden EventBridge. Sie können dies tun, wenn Sie einen Tracker [erstellen](#) oder [aktualisieren](#).

Um beispielsweise einen vorhandenen Tracker mithilfe der zu aktualisieren AWS CLI, können Sie den folgenden Befehl verwenden (verwenden Sie den Namen Ihrer Tracker-Ressource anstelle von *MyTracker*).

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Um Positionereignisse für einen Tracker zu deaktivieren, müssen Sie die -API oder die Amazon Location Service-Konsole verwenden.

Erstellen von Ereignisregeln für Amazon Location

Sie können [bis zu 300 Regeln pro Event Bus](#) in erstellen EventBridge , um Aktionen zu konfigurieren, die als Reaktion auf ein Amazon Location-Ereignis ausgeführt werden.

Sie können beispielsweise eine Regel für Geofence-Ereignisse erstellen, bei denen eine Push-Benachrichtigung gesendet wird, wenn ein Telefon innerhalb einer Geofenced-Grenze erkannt wird.

So erstellen Sie eine Regel für Amazon-Location-Ereignisse

Erstellen Sie mit den folgenden Werten [eine - EventBridge Regel](#) basierend auf Amazon Location-Ereignissen:

- Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
- Fügen Sie im Feld Ereignismuster das folgende Muster hinzu:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
}
```

Um eine Regel für Tracker-Positionsaktualisierungen zu erstellen, können Sie stattdessen das folgende Muster verwenden:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

Sie können optional nur - ENTER oder -EXIT Ereignisse angeben, indem Sie ein detail Tag hinzufügen (wenn Ihre Regel für Tracker-Positionsaktualisierungen bestimmt ist, gibt es nur ein einziges EventType, sodass Sie es nicht filtern müssen):

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

```
}
```

Sie können optional auch nach Eigenschaften der Position oder des Geofence filtern:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

- Wählen Sie unter Ziele auswählen die Zielaktion aus, die ausgeführt werden soll, wenn ein Ereignis vom Amazon Location Service empfangen wird.

Verwenden Sie beispielsweise ein Amazon Simple Notification Service (SNS)-Thema, um eine E-Mail oder Textnachricht zu senden, wenn ein Ereignis eintritt. Sie müssen zuerst mit der Amazon-SNS-Konsole ein Amazon-SNS-Thema erstellen. Weitere Informationen finden Sie unter [Verwenden von Amazon SNS für Benutzerbenachrichtigungen](#).

Warning

Es hat sich bewährt zu bestätigen, dass die Ereignisregel erfolgreich angewendet wurde oder Ihre automatisierte Aktion möglicherweise nicht wie erwartet ausgelöst wird. Um Ihre Ereignisregel zu überprüfen, initiieren Sie Bedingungen für die Ereignisregel. Simulieren Sie beispielsweise ein Gerät, das in einen Geofenced-Bereich gelangt.

Sie können auch alle Ereignisse von Amazon Location erfassen, indem Sie einfach den `detail-type` Abschnitt ausschließen. Beispielsweise:

```
{
  "source": [
    "aws.geo"
```

```
]
}
```

Note

Das gleiche Ereignis kann mehr als einmal zugestellt werden. Sie können die Ereignis-ID verwenden, um die Ereignisse, die Sie erhalten, zu deduplizieren.

Amazon- EventBridge Ereignisbeispiele für Amazon Location Service

Im Folgenden finden Sie ein Beispiel für ein Ereignis für die Eingabe eines Geofence, das durch Aufrufen von `BatchUpdateDevicePosition` initiiert wird.

```
{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",
  "source": "aws.geo",
  "account": "636103698109",
  "time": "2020-11-10T23:43:37Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-GeofenceCollection_EXAMPLE",
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "ENTER",
    "GeofenceId": "polygon_14",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:43:37.531Z",
    "Position": [
      -123.12390073297821,
      49.23433613216247
    ],
    "Accuracy": {
      "Horizontal": 15.3
    },
    "GeofenceProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}
```

```

    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}

```

Im Folgenden finden Sie ein Beispiel für ein Ereignis zum Beenden eines Geofence, das durch Aufrufen von `initiiert` wird `BatchUpdateDevicePosition`.

```

{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",
  "source": "aws.geo",
  "account": "123456789012",
  "time": "2020-11-10T23:41:44Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-GeofenceCollection_EXAMPLE",
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "EXIT",
    "GeofenceId": "polygon_10",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:41:43.826Z",
    "Position": [
      -123.08569321875426,
      49.23766166742559
    ],
    "Accuracy": {
      "Horizontal": 15.3
    },
    "GeofenceProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}

```

```
}  
}  
}
```

Im Folgenden finden Sie ein Beispiel für ein Ereignis für eine Positionsaktualisierung, das durch Aufrufen von `initiiert wirdBatchUpdateDevicePosition`.

```
{  
  "version": "0",  
  "id": "aa11aa22-33a-4a4a-aaa5-example",  
  "detail-type": "Location Device Position Event",  
  "source": "aws.geo",  
  "account": "123456789012",  
  "time": "2020-11-10T23:41:44Z",  
  "region": "eu-west-1",  
  "resources": [  
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"  
  ],  
  "detail": {  
    "EventType": "UPDATE",  
    "TrackerName": "tracker_2",  
    "DeviceId": "Device1-EXAMPLE",  
    "SampleTime": "2020-11-10T23:41:43.826Z",  
    "ReceivedTime": "2020-11-10T23:41:39.235Z",  
    "Position": [  
      -123.08569321875426,  
      49.23766166742559  
    ],  
    "Accuracy": {  
      "Horizontal": 15.3  
    },  
    "PositionProperties": {  
      "ExampleKey1": "ExampleField1",  
      "ExampleKey2": "ExampleField2"  
    }  
  }  
}
```

Nachverfolgung mit AWS IoT und MQTT mit Amazon Location Service

[MQTT](#) ist ein leichtgewichtiges und weit verbreitetes Messaging-Protokoll, das für eingeschränkte Geräte entwickelt wurde. AWS IoT Core unterstützt Geräteverbindungen, die das MQTT-Protokoll und das MQTT over WebSocket Secure (WSS)-Protokoll verwenden.

[AWS IoT Core](#) verbindet Geräte mit AWS und ermöglicht es Ihnen, Nachrichten zwischen ihnen zu senden und zu empfangen. Die AWS IoT Core Regel-Engine speichert Abfragen zu den Nachrichtenthemen Ihrer Geräte und ermöglicht es Ihnen, Aktionen zum Senden von Nachrichten an andere -AWS-Services wie Amazon Location Service zu definieren. Geräte, die ihren Standort als Koordinaten kennen, können ihre Standorte über die Regel-Engine an Amazon Location weiterleiten.

Note

Geräte können ihre eigene Position kennen, z. B. über integriertes GPS. unterstützt AWS IoT auch die Standortverfolgung von Geräten von Drittanbietern. Weitere Informationen finden Sie unter [AWS IoT Core Device Location](#) im AWS IoT Core-Entwicklerhandbuch.

In der folgenden Anleitung wird die Nachverfolgung mithilfe von -AWS IoT Core-Regeln beschrieben. Sie können die Geräteinformationen auch an Ihre eigene AWS Lambda Funktion senden, wenn Sie sie vor dem Senden an Amazon Location verarbeiten müssen. Weitere Informationen zur Verwendung von Lambda zur Verarbeitung Ihrer Gerätestandorte finden Sie unter [Verwenden von AWS Lambda mit MQTT](#).

Themen

- [Voraussetzung](#)
- [Erstellen einer AWS IoT Core-Regel](#)
- [Testen Ihrer AWS IoT Core Regel in der Konsole](#)
- [Verwenden von AWS Lambda mit MQTT](#)

Voraussetzung

Bevor Sie mit der Nachverfolgung beginnen können, müssen Sie die folgenden Voraussetzungen erfüllen:

- [Erstellen Sie eine Tracker-Ressource](#), an die Sie die Gerätestandortdaten senden.

- [Erstellen Sie eine IAM-Rolle](#), um AWS IoT Core Zugriff auf Ihren Tracker zu gewähren.

Wenn Sie diese Schritte ausführen, verwenden Sie die folgende Richtlinie, um Zugriff auf Ihren Tracker zu gewähren:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

Erstellen einer AWS IoT Core-Regel

Erstellen Sie als Nächstes eine -AWS IoT Core-Regel, um die Positionstelemetrie Ihrer Geräte an Amazon Location Service weiterzuleiten. Weitere Informationen zum Erstellen von Regeln finden Sie in den folgenden Themen im AWS IoT Core Entwicklerhandbuch für :

- [Erstellen einer -AWS IoT-Regel](#) für Informationen zum Erstellen einer neuen Regel.
- [Standortaktion](#) für Informationen zum Erstellen einer Regel für die Veröffentlichung in Amazon Location

Testen Ihrer AWS IoT Core Regel in der Konsole

Wenn derzeit keine Geräte Telemetrie veröffentlichen, die den Standort enthält, können Sie Ihre Regel mit der AWS IoT Core Konsole testen. Die Konsole verfügt über einen Testclient, über den Sie eine Beispielnachrichtis veröffentlichen können, um die Ergebnisse der Lösung zu überprüfen.

1. Melden Sie sich bei der -AWS IoT Core-Konsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation Test und wählen Sie MQTT-Testclient aus.
3. Legen Sie unter In einem Thema veröffentlichen den Themennamen auf *iot/topic* (oder den Namen des Themas, das Sie in Ihrer AWS IoT Core Regel eingerichtet haben, falls unterschiedlich) fest und geben Sie Folgendes für die Nachrichtennutzlast an. Ersetzen Sie den

Zeitstempel **1604940328** durch einen gültigen Zeitstempel innerhalb der letzten 30 Tage (alle Zeitstempel, die älter als 30 Tage sind, werden von Amazon Location Service-Trackern ignoriert).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Wählen Sie Im Thema veröffentlichen, um die Testnachricht zu senden.
5. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob die Nachricht vom Amazon Location Service empfangen wurde. Wenn Sie ihn während der Einrichtung geändert haben, ersetzen Sie den Tracker-Namen durch den von Ihnen verwendeten.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Verwenden von AWS Lambda mit MQTT

Während die Verwendung von nicht mehr erforderlich AWS Lambda ist, wenn Sie Gerätestandortdaten zur Nachverfolgung an Amazon Location senden, sollten Sie in einigen Fällen trotzdem Lambda verwenden. Wenn Sie beispielsweise Ihre Gerätestandortdaten selbst verarbeiten möchten, bevor Sie sie an Amazon Location senden. In den folgenden Themen wird beschrieben, wie Sie mit Lambda Nachrichten verarbeiten, bevor Sie sie an Ihren Tracker senden. Weitere Informationen zu diesem Muster finden Sie in der [Referenzarchitektur](#).

Themen

- [Voraussetzung](#)
- [Erstellen einer Lambda-Funktion](#)
- [Erstellen einer AWS IoT Core-Regel](#)
- [Testen Ihrer AWS IoT Core Regel in der Konsole](#)

Voraussetzung

Bevor Sie mit der Nachverfolgung beginnen können, müssen Sie [eine Tracker-Ressource erstellen](#). Um eine Tracker-Ressource zu erstellen, können Sie die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs verwenden.

Im folgenden Beispiel wird die Amazon Location Service-Konsole verwendet, um die Tracker-Ressource zu erstellen:

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie Tracker erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen eindeutigen Namen mit maximal 100 Zeichen ein. Gültige Einträge sind alphanumerische Zeichen, Bindestriche und Unterstriche. Zum Beispiel *MyTracker*.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Zum Beispiel *Tracker zum Speichern von AWS IoT Core Gerätepositionen*.
 - Positionsfilerung – Wählen Sie die Filterung aus, die Sie für Positionsaktualisierungen verwenden möchten. Zum Beispiel Genauigkeitsbasierte Filterung.
5. Wählen Sie Tracker erstellen aus.

Erstellen einer Lambda-Funktion

Um eine Verbindung zwischen AWS IoT Core und Amazon Location Service herzustellen, benötigen Sie eine -AWS Lambda-Funktion, um Nachrichten zu verarbeiten, die von weitergeleitet werden AWS IoT Core. Diese Funktion extrahiert alle Positionsdaten, formatiert sie für Amazon Location Service und sendet sie über die Amazon Location Tracker API. Sie können diese Funktion über die AWS Lambda Konsole erstellen oder die AWS Command Line Interface (AWS CLI) oder die AWS Lambda APIs verwenden.

So erstellen Sie eine Lambda-Funktion, die Positionsaktualisierungen in Amazon Location über die Konsole veröffentlicht:

1. Öffnen Sie die AWS Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie in der linken Navigation Funktionen aus.

3. Wählen Sie Funktion erstellen und stellen Sie sicher, dass Von Grund auf neu erstellen ausgewählt ist.
4. Füllen Sie die folgenden Felder aus:
 - Funktionsname – Geben Sie einen eindeutigen Namen für Ihre Funktion ein. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche und Unterstriche ohne Leerzeichen. Zum Beispiel *MyLambda*.
 - Laufzeit – Wählen Sie *Python 3.8* aus.
5. Wählen Sie Funktion erstellen.
6. Wählen Sie die Registerkarte Code, um den Editor zu öffnen.
7. Überschreiben Sie den Platzhaltercode in `lambda_function.py` mit dem folgenden und ersetzen Sie den Wert, der dem zugewiesen ist, durch `TRACKER_NAME` den Namen des Trackers, den Sie als [Voraussetzung](#) erstellt haben.

```
from datetime import datetime
import json
import os

import boto3

# Update this to match the name of your Tracker resource
TRACKER_NAME = "MyTracker"

"""
This Lambda function receives a payload from AWS IoT Core and publishes device
updates to
Amazon Location Service via the BatchUpdateDevicePosition API.

Parameter 'event' is the payload delivered from AWS IoT Core.

In this sample, we assume that the payload has a single top-level key 'payload' and
a nested key
'location' with keys 'lat' and 'long'. We also assume that the name of the device
is nested in
the payload as 'deviceid'. Finally, the timestamp of the payload is present as
'timestamp'. For
example:

>>> event
{ 'payload': { 'deviceid': 'thing123', 'timestamp': 1604940328,
```

```
'location': { 'lat': 49.2819, 'long': -123.1187 },
'accuracy': {'Horizontal': 20.5 },
'positionProperties': {'field1':'value1','field2':'value2'} }
}
```

If your data doesn't match this schema, you can either use the AWS IoT Core rules engine to format the data before delivering it to this Lambda function, or you can modify the code below to match it.

```
"""
def lambda_handler(event, context):
    update = {
        "DeviceId": event["payload"]["deviceid"],
        "SampleTime": datetime.fromtimestamp(event["payload"]
["timestamp"]).strftime("%Y-%m-%dT%H:%M:%SZ"),
        "Position": [
            event["payload"]["location"]["long"],
            event["payload"]["location"]["lat"]
        ]
    }
    if "accuracy" in event["payload"]:
        update["Accuracy"] = event["payload"]['accuracy']
    if "positionProperties" in event["payload"]:
        update["PositionProperties"] = event["payload"]['positionProperties']

    client = boto3.client("location")
    response = client.batch_update_device_position(TrackerName=TRACKER_NAME,
Updates=[update])

    return {
        "statusCode": 200,
        "body": json.dumps(response)
    }
```

8. Wählen Sie Bereitstellen, um die aktualisierte Funktion zu speichern.
9. Wählen Sie die Registerkarte Konfiguration aus.
10. Wählen Sie im Abschnitt Berechtigungen den Hyperlink Rollennamen aus, um Amazon Location Service Berechtigungen für Ihre Lambda-Funktion zu erteilen.
11. Wählen Sie auf der Übersichtsseite Ihrer Rolle die Option Berechtigungen hinzufügen und dann aus der Dropdown-Liste die Option Inline-Richtlinie erstellen aus.

- Wählen Sie die Registerkarte JSON und überschreiben Sie die Richtlinie mit dem folgenden Dokument. Auf diese Weise kann Ihre Lambda-Funktion Gerätepositionen aktualisieren, die von allen Tracker-Ressourcen in allen Regionen verwaltet werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

- Wählen Sie Richtlinie prüfen.
- Geben Sie den Namen einer Richtlinie ein. Zum Beispiel *AmazonLocationTrackerWriteOnly*.
- Wählen Sie Richtlinie erstellen aus.

Sie können diesen Funktionscode nach Bedarf ändern, um ihn an Ihr eigenes Gerätenachrichtenschema anzupassen.

Erstellen einer AWS IoT Core-Regel

Erstellen Sie als Nächstes eine AWS IoT Core Regel, um die Positionstelemetrie Ihrer Geräte zur Transformation und Veröffentlichung an Amazon Location Service an die AWS Lambda Funktion weiterzuleiten. Die bereitgestellte Beispielregel geht davon aus, dass jede erforderliche Transformation von Gerätenutzlasten von Ihrer Lambda-Funktion durchgeführt wird. Sie können diese Regel über die AWS IoT Core-Konsole, die AWS Command Line Interface (AWS CLI) oder die AWS IoT Core APIs erstellen.

Note

Während die AWS IoT-Konsole die Berechtigung übernimmt AWS IoT Core, die zum Aufrufen Ihrer Lambda-Funktion erforderlich ist, müssen Sie eine Richtlinie konfigurieren AWS CLI, um die Berechtigung zu erteilen. [AWS IoT](#)

So erstellen Sie eine AWS IoT Core mithilfe der Konsole

1. Melden Sie sich bei der -AWS IoT CoreKonsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation die Option Act und wählen Sie Rules aus.
3. Wählen Sie Regel erstellen, um den neuen Regelassistenten zu starten.
4. Geben Sie für Ihre Regel einen Namen und eine Beschreibung ein.
5. Aktualisieren Sie für die Regelabfrageanweisung das Attribut so, dass es auf ein Thema verweist, in dem mindestens ein Gerät Telemetrie veröffentlicht, die den Standort enthält. FROM
Wenn Sie die Lösung testen, sind keine Änderungen erforderlich.

```
SELECT * FROM 'iot/topic'
```

6. Wählen Sie unter Eine oder mehrere Aktionen festlegen die Option Aktion hinzufügen aus.
7. Wählen Sie Nachricht an eine Lambda-Funktion senden aus.
8. Wählen Sie Configure action.
9. Suchen und wählen Sie Ihre Lambda-Funktion aus der Liste aus.
10. Wählen Sie Aktion hinzufügen aus.
11. Wählen Sie Regel erstellen aus.

Testen Ihrer AWS IoT Core Regel in der Konsole

Wenn derzeit keine Geräte Telemetrie veröffentlichen, die den Standort enthält, können Sie Ihre Regel und diese Lösung mithilfe der AWS IoT Core Konsole testen. Die Konsole verfügt über einen Testclient, über den Sie eine Beispielnachricht veröffentlichen können, um die Ergebnisse der Lösung zu überprüfen.

1. Melden Sie sich bei der -AWS IoT CoreKonsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation Test und wählen Sie MQTT-Testclient aus.
3. Legen Sie unter In einem Thema veröffentlichen den Themennamen auf *iot/topic* (oder den Namen des Themas, das Sie in Ihrer AWS IoT Core Regel eingerichtet haben, falls unterschiedlich) fest und geben Sie Folgendes für die Nachrichtennutzlast an. Ersetzen Sie den Zeitstempel *1604940328* durch einen gültigen Zeitstempel innerhalb der letzten 30 Tage (alle Zeitstempel, die älter als 30 Tage sind, werden ignoriert).

```
{
```

```
"payload": {
  "deviceid": "thing123",
  "timestamp": 1604940328,
  "location": { "lat": 49.2819, "long": -123.1187 },
  "accuracy": { "Horizontal": 20.5 },
  "positionProperties": { "field1": "value1", "field2": "value2" }
}
```

4. Wählen Sie Im Thema veröffentlichen, um die Testnachricht zu senden.
5. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob die Nachricht vom Amazon Location Service empfangen wurde. Wenn Sie sie während der Einrichtung geändert haben, ersetzen Sie den Tracker-Namen und die Geräte-ID durch die von Ihnen verwendeten.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Verwalten Ihrer Geofence-Sammlungsressourcen

Verwalten Sie Ihre Geofence-Sammlungen mithilfe der Amazon Location-Konsole, der AWS CLI, oder der Amazon Location APIs.

Auflisten Ihrer Geofence-Sammlungsressourcen

Sie können Ihre Geofence-Sammlungsliste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste von Geofence-Sammlungen mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Zeigen Sie eine Liste Ihrer Geofence-Sammlungen unter Meine Geofence-Sammlungen an.

API

Verwenden Sie die [ListGeofenceCollections](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Geofence-Sammlungen im AWS Konto.

```
POST /geofencing/v0/list-collections
```

Im Folgenden finden Sie eine Beispielantwort für `ListGeofenceCollections`:

```
{
  "Entries": [
    {
      "CollectionName": "ExampleCollection",
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "Description": "string",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    },
    "NextToken": "1234-5678-9012"
  ]
}
```

CLI

Verwenden Sie den [list-geofence-collections](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Geofence-Sammlungen im AWS Konto abzurufen.

```
aws location list-geofence-collections
```

Details zur Geofence-Sammlung abrufen

Sie können Details zu jeder Geofence-Sammlungsressource in Ihrem AWS Konto über die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs abrufen:

Console

So zeigen Sie die Details einer Geofence-Sammlung mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlungen den Namenslink der Ziel-Geofence-Sammlung aus.

API

Verwenden Sie die [DescribeGeofenceCollection](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Details der Geofence-Sammlung für *ExampleCollection*.

```
GET /geofencing/v0/collections/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für DescribeGeofenceCollection:

```
{
  "CollectionArn": "arn:aws:geo:us-west-2:123456789012:geofence-collection/
GeofenceCollection",
  "CollectionName": "ExampleCollection",
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "Description": "string",
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

Verwenden Sie den [describe-geofence-collection](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details der Geofence-Sammlung für abzurufen *ExampleCollection*.

```
aws location describe-geofence-collection \
  --collection-name "ExampleCollection"
```

Löschen einer Geofence-Sammlung

Sie können eine Geofence-Sammlung aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Console

So löschen Sie eine Geofence-Sammlung mit der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung die Ziel-Geofence-Sammlung aus.
4. Wählen Sie Geofence-Sammlung löschen aus.

API

Verwenden Sie die [DeleteGeofenceCollection](#) Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Geofence-Sammlung *ExampleCollection*.

```
DELETE /geofencing/v0/collections/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für DeleteGeofenceCollection:

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-geofence-collection](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Geofence-Sammlung *ExampleCollection*.

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

Gespeicherte Geofences auflisten

Sie können Geofences auflisten, die in einer bestimmten Geofence-Sammlung gespeichert sind, indem Sie die Amazon Location-Konsole, die AWS CLI, oder die Amazon Location APIs verwenden.

Console

So zeigen Sie eine Liste von Geofences mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Geofences in der Geofence-Sammlung unter Geofences anzeigen

API

Verwenden Sie die [ListGeofences](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Geofences, die in der Geofence-Sammlung gespeichert sind *ExampleCollection*.

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

Im Folgenden finden Sie eine Beispielantwort für ListGeofences:

```
{
  "Entries": [
    {
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "GeofenceId": "geofence-1",
      "Geometry": {
        "Polygon": [
          [-5.716667, -15.933333,
          [-14.416667, -7.933333],
          [-12.316667, -37.066667],
          [-5.716667, -15.933333]
        ]
      },
      "Status": "ACTIVE",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    }
  ]
}
```

```
    }  
  ],  
  "NextToken": "1234-5678-9012"  
}
```

CLI

Verwenden Sie den [list-geofences](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Geofences abzurufen, die in der Geofence-Sammlung gespeichert sind *ExampleCollection*.

```
aws location list-geofences \  
  --collection-name "ExampleCollection"
```

Geofence-Details abrufen

Sie können die Details eines bestimmten Geofence, z. B. Erstellungszeit, Aktualisierungszeit, Geometrie und Status, aus einer Geofence-Sammlung mithilfe der Amazon Location-Konsole, AWS CLI oder der Amazon Location APIs abrufen.

Console

So zeigen Sie den Status eines Geofence mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Unter Geofences können Sie den Status Ihrer Geofences anzeigen.

API

Verwenden Sie die [-GetGeofence](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Geofence-Details aus einer Geofence-Sammlung *ExampleCollection*.

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

Im Folgenden finden Sie eine Beispielantwort für GetGeofence:

```
{
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "GeofenceId": "ExampleGeofence1",
  "Geometry": {
    "Polygon": [
      [-1,-1],
      [1,-1],
      [0,1],
      [-1,-1]
    ]
  },
  "Status": "ACTIVE",
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

Verwenden Sie den [get-geofence](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details der Geofence-Sammlung für abzurufen *ExampleCollection*.

```
aws location get-geofence \
  --collection-name "ExampleCollection" \
  --geofence-id "ExampleGeofence1"
```

Geofences löschen

Sie können Geofences aus einer Geofence-Sammlung mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Console

So löschen Sie ein Geofence mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Wählen Sie unter Geofences den Ziel-Geofence aus.
5. Wählen Sie Geofence löschen aus.

API

Verwenden Sie die [BatchDeleteGeofence](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen von Geofences aus der Geofence-Sammlung *ExampleCollection*.

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [BatchDeleteGeofence](#).

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [batch-delete-geofence](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen von Geofences aus der Geofence-Sammlung *ExampleCollection*.

```
aws location batch-delete-geofence \
  --collection-name "ExampleCollection" \
  --geofence-ids "ExampleGeofence11"
```

Verwalten Ihrer Tracker-Ressourcen

Sie können Ihre Tracker mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten Ihrer Tracker

Sie können Ihre Tracker-Liste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der vorhandenen Tracker mithilfe der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie in der linken Navigation Tracker aus.
3. Zeigen Sie eine Liste Ihrer Tracker-Ressourcen unter Meine Tracker an.

API

Verwenden Sie die [ListTrackers](#) Operation aus den Amazon Location Trackers APIs .

Das folgende Beispiel ist eine API-Anfrage zum Abrufen einer Liste von Trackern in Ihrem AWS Konto.

```
POST /tracking/v0/list-trackers
```

Im Folgenden finden Sie eine Beispielantwort für [ListTrackers](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-02T19:09:07.327Z,
      "Description": "string",
      "TrackerName": "ExampleTracker",
      "UpdateTime": 2020-10-02T19:10:07.327Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-trackers](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Trackern in Ihrem AWS Konto abzurufen.

```
aws location list-trackers
```

Trennen eines Trackers von einer Geofence-Sammlung

Sie können einen Tracker von einer Geofence-Sammlung trennen, indem Sie die Amazon Location-Konsole AWS CLI, die oder die Amazon Location APIs verwenden:

Console

So heben Sie die Zuordnung eines Trackers zu einer zugeordneten Geofence-Sammlung mithilfe der Amazon-Standortkonsole auf

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie unter Meine Tracker den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen eine Geofence-Sammlung mit dem Status Verknüpft aus.
5. Wählen Sie Verknüpfung aufheben aus.

API

Verwenden Sie die [DisassociateTrackerConsumer](#) Operation aus den Amazon Location Trackers APIs .

Das folgende Beispiel ist eine API-Anforderung zum Trennen eines Trackers von einer zugeordneten Geofence-Sammlung.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für [DisassociateTrackerConsumer](#):


```
HTTP/1.1 200
```

CLI

Verwenden Sie den [disassociate-tracker-consumer](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl, um die Zuordnung eines Trackers zu einer zugeordneten Geofence-Sammlung aufzuheben.

```
aws location disassociate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/  
ExampleCollection" \  
  --tracker-name "ExampleTracker"
```

Abrufen von Tracker-Details

Sie können Details zu jedem Tracker in Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs abrufen.

Console

So zeigen Sie Tracker-Details mithilfe der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie in der linken Navigation Tracker aus.
3. Wählen Sie unter Meine Tracker den Namenslink des Ziel-Trackers aus.
4. Zeigen Sie die Tracker-Details unter Informationen an.

API

Verwenden Sie die [DescribeTracker](#) Operation aus den Amazon Location Tracker-APIs.

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Tracker-Details für *ExampleTracker*.

```
GET /tracking/v0/trackers/ExampleTracker
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeTracker](#):

```
{
  "CreateTime": 2020-10-02T19:09:07.327Z,
  "Description": "string",
  "EventBridgeEnabled": false,
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "PositionFiltering": "TimeBased",
  "Tags": {
    "Tag1" : "Value1"
  },
  "TrackerArn": "arn:aws:geo:us-west-2:123456789012:tracker/ExampleTracker",
  "TrackerName": "ExampleTracker",
  "UpdateTime": 2020-10-02T19:10:07.327Z
}
```

CLI

Verwenden Sie den [describe-tracker](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Abrufen von Tracker-Details für *ExampleTracker*.

```
aws location describe-tracker \
  --tracker-name "ExampleTracker"
```

Löschen eines Trackers

Sie können einen Tracker aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen:

Console

So löschen Sie eine vorhandene Kartenressource mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft. Wenn die Tracker-Ressource verwendet wird, kann ein Fehler auftreten. Stellen Sie sicher, dass die Zielressource keine Abhängigkeit für Ihre Anwendungen ist.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.

2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie unter Meine Tracker den Ziel-Tracker aus.
4. Wählen Sie Tracker löschen aus.

API

Verwenden Sie die [DeleteTracker](#) Operation aus den Amazon Location Tracker-APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen des Trackers *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

Im Folgenden finden Sie eine Beispielantwort für [DeleteTracker](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-tracker](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen des Trackers *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

Beispiel für eine mobile Geofencing- und Tracking-Anwendung

In diesem Thema werden Tutorials behandelt, die die wichtigsten Funktionen der Verwendung von Amazon Location Geofences und Trackern in einer mobilen Anwendung demonstrieren sollen. Die Anwendungen demonstrieren, wie ein Tracker und Geofence mithilfe einer Kombination aus Lambda AWS IoT - und Amazon Location-Funktionen interagieren. Es sind zwei Tutorials verfügbar.

- [Beispiel für eine Tracking- und Geofencing-Anwendung für Android](#). Sie können die Projektdateien klonen von GitHub: <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications>
- [Beispiel für eine Tracking- und Geofencing-Anwendung für iOS](#), und Sie können die Projektdateien klonen von GitHub: <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>

Beispielverfolgungs- und Geofence-Anwendung für Android

In diesem Thema wird das Android-Tutorial behandelt, mit dem die wichtigsten Funktionen der Verwendung von Amazon Location Geofences und Trackern in einer mobilen Anwendung demonstriert werden sollen. Die Anwendungen demonstrieren, wie ein Tracker und ein Geofence mithilfe einer Kombination aus Lambda AWS IoT - und Amazon Location-Funktionen interagieren.

Themen

- [Erstellen Sie Amazon-Standortressourcen für Ihre App](#)
- [Erstellen Sie eine Geofence-Sammlung](#)
- [Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung](#)
- [AWS Lambda mit MQTT verwenden](#)
- [Richten Sie den Beispiel-App-Code ein](#)
- [Verwenden Sie die Beispiel-App](#)

Erstellen Sie Amazon-Standortressourcen für Ihre App

Zu Beginn müssen Sie die erforderlichen Amazon-Standortressourcen erstellen. Diese Ressourcen sind für die Funktionalität der Anwendung und die Ausführung der bereitgestellten Codefragmente unerlässlich.

Note

Wenn Sie noch kein AWS Konto erstellt haben, folgen Sie den Anweisungen im Benutzerleitfaden zur [AWS Kontoverwaltung](#).

Zunächst müssen Sie eine Amazon Cognito Identity Pool-ID erstellen. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie die [Amazon Cognito Cognito-Konsole](#) und wählen Sie im Menü auf der linken Seite Identitätspools und dann Identitätspool erstellen aus.
2. Vergewissern Sie sich, dass Gastzugriff aktiviert ist, und klicken Sie auf Weiter, um fortzufahren.
3. Erstellen Sie als Nächstes eine neue IAM-Rolle oder Verwenden Sie eine vorhandene IAM-Rolle.
4. Geben Sie einen Namen für den Identitätspool ein und stellen Sie sicher, dass Identity Pool Zugriff auf (geo) Amazon-Standortressourcen für die Karte und den Tracker hat, die Sie im nächsten Verfahren erstellen werden.

Als Nächstes müssen Sie in der AWS Amazon Location-Konsole eine Karte erstellen und gestalten. Gehen Sie dazu wie folgt vor:

1. Navigieren Sie in der Amazon Location-Konsole zum [Bereich Maps](#) und wählen Sie Create Map aus.
2. Geben Sie der neuen Karte einen Namen und eine Beschreibung. Notieren Sie sich den Namen, den Sie vergeben, so wie er später im Tutorial verwendet wird.
3. Beachten Sie bei der Auswahl eines Kartenstils den Kartendatenanbieter. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).
4. Akzeptieren Sie die [Allgemeinen Geschäftsbedingungen von Amazon Location und](#) wählen Sie dann Create Map aus, um die Kartenerstellung abzuschließen.

Als Nächstes müssen Sie in der Amazon Location-Konsole einen Tracker erstellen. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie den [Bereich Maps](#) in der Amazon Location-Konsole.
2. Wählen Sie Tracker erstellen.
3. Füllen Sie die erforderlichen Felder aus. Notieren Sie sich den Namen des Trackers, auf den in diesem Tutorial verwiesen wird.
4. Wählen Sie im Feld Positionsfilerung die Option aus, die am besten zu Ihrer geplanten Nutzung Ihrer Tracker-Ressource passt. Wenn Sie die Positionsfilerung nicht einrichten, ist die Standardeinstellung TimeBased. Weitere Informationen finden Sie unter [Trackers](#) und [PositionFiltering](#) in der Amazon Location API-Referenz.
5. Wählen Sie Tracker erstellen, um die Erstellung des Trackers abzuschließen.

Erstellen Sie eine Geofence-Sammlung

Jetzt erstellen Sie eine Geofence-Sammlung. Sie können entweder die Konsole, die API oder die CLI verwenden. Die folgenden Verfahren führen Sie durch die einzelnen Optionen.

- Erstellen Sie mit der Amazon Location-Konsole eine Geofence-Sammlung:
 1. Öffnen Sie den Bereich [Geofence Collections](#) in der Amazon Location-Konsole.
 2. Wählen Sie Geofence-Sammlung erstellen.
 3. Geben Sie einen Namen und eine Beschreibung für die Sammlung an.

4. Unter der EventBridge Regel mit Amazon CloudWatch als Ziel können Sie eine optionale EventBridge Regel erstellen, mit der Sie sofort auf Geofence-Ereignisse reagieren können. Dadurch kann Amazon Location Ereignisse veröffentlichen Amazon CloudWatch Logs.
 5. Klicken Sie auf Geofence-Sammlung erstellen, um die Erstellung der Sammlung abzuschließen.
- Erstellen Sie mit der Amazon Location API eine Geofence-Sammlung:

Verwenden Sie den [CreateGeofenceCollection](#) Vorgang über die Amazon Location Geofences-APIs. Im folgenden Beispiel wird eine API-Anfrage verwendet, um eine Geofence-Sammlung namens zu erstellen. *GEOCOLLECTION_NAME*

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Erstellen Sie eine Geofence-Sammlung mit AWS CLI-Befehlen:

Verwenden Sie den `create-geofence-collection`-Befehl. Das folgende Beispiel verwendet eine AWS CLI, um eine Geofence-Sammlung namens zu erstellen. *GEOCOLLECTION_NAME*
Weitere Informationen zur Verwendung der AWS CLI finden Sie in der [Dokumentation zur AWS Befehlszeilenschnittstelle](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung

Um einen Tracker mit einer Geofence-Sammlung zu verknüpfen, können Sie entweder die Konsole, die API oder die CLI verwenden. Die folgenden Verfahren führen Sie durch die einzelnen Optionen.

Verknüpfen Sie mithilfe der Amazon Location Service Service-Konsole eine Tracker-Ressource mit einer Geofence-Sammlung:

1. Öffnen Sie die Amazon Location-Konsole.
2. Wählen Sie im linken Navigationsbereich Trackers aus.
3. Wählen Sie unter Device Trackers den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen die Option Geofence-Sammlung verknüpfen aus.
5. Wählen Sie im Fenster Verknüpfte Geofence-Sammlung eine Geofence-Sammlung aus dem Drop-down-Menü aus.
6. Wählen Sie Verknüpfen.
7. Nachdem Sie die Tracker-Ressource verknüpft haben, wird ihr der Status Aktiv zugewiesen.

Verknüpfen Sie mithilfe der Amazon Location APIs eine Tracker-Ressource mit einer Geofence-Sammlung:

Verwenden Sie den `AssociateTrackerConsumer` Vorgang über die Amazon Location Trackers APIs. Im folgenden Beispiel wird eine API-Anfrage verwendet, die eine `ExampleTracker` mithilfe ihres Amazon-Ressourcennamens (ARN) einer Geofence-Sammlung zuordnet.

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
}
```

Verknüpfen Sie eine Tracker-Ressource mithilfe von AWS CLI-Befehlen mit einer Geofence-Sammlung:

Verwenden Sie den `associate-tracker-consumer` -Befehl. Das folgende Beispiel verwendet eine AWS CLI, um eine Geofence-Sammlung namens zu erstellen. *GOECOLLECTION_NAME*

```
aws location \
```

```
associate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GOECOLLECTION_NAME" \  
  --tracker-name "ExampleTracker"
```

AWS Lambda mit MQTT verwenden

Um eine Verbindung zwischen AWS IoT und Amazon Location herzustellen, benötigen Sie eine Lambda-Funktion zur Verarbeitung von Nachrichten, die durch EventBridge CloudWatch Ereignisse weitergeleitet werden. Diese Funktion extrahiert alle Positionsdaten, formatiert sie für Amazon Location und übermittelt sie über die Amazon Location Tracker-API.

Das folgende Verfahren zeigt Ihnen, wie Sie diese Funktion über die Lambda-Konsole erstellen:

1. Öffnen Sie die [-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste Funktionen aus.
3. Wählen Sie dann „Funktion erstellen“ und stellen Sie sicher, dass die Option „Von Grund auf neu erstellen“ ausgewählt ist.
4. geben Sie einen Funktionsnamen ein, und wählen Sie für die Runtime-Option Node.js 16.x aus.
5. Wählen Sie Funktion erstellen.
6. Öffnen Sie die Registerkarte Code, um auf den Editor zuzugreifen.
7. Überschreiben Sie den Platzhaltercode in der `index.js` Datei mit dem Folgenden:

```
const AWS = require('aws-sdk')  
const iot = new AWS.Iot();  
exports.handler = function(event) {  
  console.log("event===>>>", JSON.stringify(event));  
  var param = {  
    endpointType: "iot:Data-ATS"  
  };  
  iot.describeEndpoint(param, function(err, data) {  
    if (err) {  
      console.log("error===>>>", err, err.stack); // an error occurred  
    } else {  
      var endp = data['endpointAddress'];  
      const iotdata = new AWS.IotData({endpoint: endp});  
      const trackerEvent = event["detail"]["EventType"];
```



```

const src = event["source"];
const time = event["time"];
const gfId = event["detail"]["GeofenceId"];
const resources = event["resources"][0];
const splitResources = resources.split(".");
const geofenceCollection = splitResources[splitResources.length -
1];

const coordinates = event["detail"]["Position"];

const deviceId = event["detail"]["DeviceId"];
console.log("deviceId===>>>", deviceId);
const msg = {
  "trackerEventType" : trackerEvent,
  "source" : src,
  "eventTime" : time,
  "geofenceId" : gfId,
  "coordinates": coordinates,
  "geofenceCollection": geofenceCollection
};
const params = {
  topic: `${deviceId}/tracker`,
  payload: JSON.stringify(msg),
  qos: 0
};
iotdata.publish(params, function(err, data) {
  if (err) {
    console.log("error===>>>", err, err.stack); // an error
occurred

  } else {
    console.log("Ladmbda triggered===>>>", trackerEvent); //
successful response
  }
});
}
});
}

```

8. Drücken Sie auf Deploy, um die aktualisierte Funktion zu speichern.
9. Öffnen Sie als Nächstes die Registerkarte Konfiguration.
10. Klicken Sie im Bereich Trigger auf die Schaltfläche Trigger hinzufügen.
11. Wählen Sie im Feld Quelle EventBridge (CloudWatch Ereignisse) aus.
12. Wählen Sie die Option Bestehende Regeln aus.

13. Geben Sie beispielsweise den Regelnamen ein `AmazonLocationMonitor-GEOFENCECOLLECTION_NAME`.
14. Drücken Sie die Schaltfläche Hinzufügen.
15. Dadurch werden auch ressourcenbasierte Richtlinienenerklärungen auf der Registerkarte „Berechtigungen“ angehängt

Jetzt richten Sie den MQTT-Testclient mithilfe des folgenden AWS IoT Verfahrens ein:

1. Öffnen Sie die Datei <https://console.aws.amazon.com/iot/>.
2. Wählen Sie im linken Navigationsbereich den MQTT-Testclient aus.
3. Sie sehen einen Abschnitt mit dem Titel MQTT-Testclient, in dem Sie Ihre MQTT-Verbindung konfigurieren können.
4. Nachdem Sie die erforderlichen Einstellungen konfiguriert haben, klicken Sie auf die Schaltfläche Connect, um mithilfe der bereitgestellten Parameter eine Verbindung zum MQTT-Broker herzustellen.
5. Notieren Sie den Endpunkt, so wie er später im Tutorial verwendet wird.

Sobald Sie mit dem Testclient verbunden sind, können Sie MQTT-Themen abonnieren oder Nachrichten zu Themen veröffentlichen, indem Sie die entsprechenden Eingabefelder verwenden, die in der MQTT-Testclient-Oberfläche bereitgestellt werden. Als Nächstes erstellen Sie eine AWS IoT Richtlinie.

6. Erweitern Sie im Menü auf der linken Seite unter Verwalten die Option Sicherheit und klicken Sie auf Richtlinien.
7. Klicken Sie auf die Schaltfläche Richtlinie erstellen.
8. Geben Sie den Namen einer Richtlinie ein.
9. Wählen Sie im Richtliniendokument die Registerkarte JSON aus.
10. Kopieren Sie die unten gezeigte Richtlinie und fügen Sie sie ein. Achten Sie jedoch darauf, alle Elemente mit Ihrem **REGION** und zu aktualisieren **ACCOUNT_ID**:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:Connect",
        "iot:Publish",
```

```

        "iot:Subscribe",
        "iot:Receive"
    ],
    "Resource": [
        "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-identity.amazonaws.com:sub}/*",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}/tracker"
    ],
    "Effect": "Allow"
}
]
}

```

11. Wählen Sie die Schaltfläche Erstellen, um den Vorgang abzuschließen.

Nachdem Sie das vorherige Verfahren abgeschlossen haben, aktualisieren Sie nun die Berechtigungen für die Gastrolle wie folgt:

1. Navigieren Sie zu Amazon Cognito und öffnen Sie Ihren Identitätspool. Fahren Sie dann mit dem Benutzerzugriff fort und wählen Sie die Gastrolle aus.
2. Klicken Sie auf Berechtigungsrichtlinien, um die Bearbeitung zu aktivieren.

```

{
  'Version': '2012-10-17',
  'Statement': [
    {
      'Action': [
        'geo:GetMap*',
        'geo:BatchUpdateDevicePosition',
        'geo:BatchEvaluateGeofences',
        'iot:Subscribe',
        'iot:Publish',
        'iot:Connect',
        'iot:Receive',
        'iot:AttachPrincipalPolicy',
        'iot:AttachPolicy',
        'iot:DetachPrincipalPolicy',

```

```

        'iot:DetachPolicy'
    ],
    'Resource': [
        'arn:aws:geo:us-east-1:{USER_ID}:map/{MAP_NAME}',
        'arn:aws:geo:us-east-1:{USER_ID}:tracker/{TRACKER_NAME}',
        'arn:aws:geo:us-east-1:{USER_ID}:geofence-collection/
{GEOFENCE_COLLECTION_NAME}',
        'arn:aws:iot:us-east-1:{USER_ID}:client/${cognito-
identity.amazonaws.com:sub}',
        'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}',
        'arn:aws:iot:us-east-1:{USER_ID}:topicfilter/${cognito-
identity.amazonaws.com:sub}/*',
        'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}/tracker'
    ],
    'Effect': 'Allow'
},
{
    'Condition': {
        'StringEquals': {
            'cognito-identity.amazonaws.com:sub': '${cognito-
identity.amazonaws.com:sub}'
        }
    },
    'Action': [
        'iot:AttachPolicy',
        'iot:DetachPolicy',
        'iot:AttachPrincipalPolicy',
        'iot:DetachPrincipalPolicy'
    ],
    'Resource': [
        '*'
    ],
    'Effect': 'Allow'
}
]
}

```

3. Mit den oben genannten Richtlinienänderungen sind nun alle erforderlichen AWS Ressourcen für die Anwendung entsprechend konfiguriert.

Richten Sie den Beispiel-App-Code ein

1. Klonen Sie dieses Repository: [https://github.com/aws-geospatial/ amazon-location-samples-android /tree/main/ tracking-with-geofence-notifications](https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications) auf Ihren lokalen Computer.
2. Öffnen Sie das AmazonSampleSDKApp Projekt in Android Studio.
3. Erstellen Sie die App und führen Sie sie auf Ihrem Android-Gerät oder Emulator aus.

Verwenden Sie die Beispiel-App

Gehen Sie wie folgt vor, um das Beispiel zu verwenden:

- Erstellen Sie ein **custom.properties**:

Gehen Sie folgendermaßen vor, um Ihre `custom.properties` Datei zu konfigurieren:

1. Öffnen Sie Ihren bevorzugten Texteditor oder Ihre bevorzugte IDE.
2. Erstellen Sie eine neue Datei.
3. Speichern Sie die Datei mit dem Namen `custom.properties`.
4. Aktualisieren Sie das `custom.properties` mit dem folgenden Codebeispiel und ersetzen Sie `MQTT_END_POINTPOLICY_NAME`, `GEOFENCE_COLLECTION_NAME`, und `TOPIC_TRACKER` durch Ihre Ressourcennamen:

```
MQTT_END_POINT=YOUR_END_POINT.us-east-1.amazonaws.com  
POLICY_NAME=YOUR_POLICY  
GEOFENCE_COLLECTION_NAME=YOUR_GEOFENCE  
TOPIC_TRACKER=YOUR_TRACKER
```

5. Säubern Sie das Projekt und erstellen Sie es neu. Danach können Sie das Projekt ausführen.
- Einloggen:

Gehen Sie wie folgt vor, um sich bei der Anwendung anzumelden:

1. Drücken Sie die Anmeldetaste.
2. Geben Sie eine Identitätspool-ID, einen Tracker-Namen und einen Kartennamen an.
3. Drücken Sie erneut auf Anmelden, um den Vorgang abzuschließen.

- Filter verwalten:

Öffnen Sie den Konfigurationsbildschirm und gehen Sie wie folgt vor:

1. Schalten Sie Filter mithilfe der Switch-Benutzeroberfläche ein oder aus.
 2. Aktualisieren Sie die Zeit- und Entfernungsfiler bei Bedarf.
- Nachverfolgung von Vorgängen:

Öffnen Sie den Tracking-Bildschirm und gehen Sie wie folgt vor:

- Sie können das Tracking im Vordergrund, Hintergrund oder im Energiesparmodus starten und beenden, indem Sie die entsprechenden Tasten drücken.

Beispiel-Tracking- und Geofencing-Anwendung für iOS

In diesem Thema wird das iOS-Tutorial behandelt, mit dem die wichtigsten Funktionen der Verwendung von Amazon Location Geofences und Trackern in einer mobilen Anwendung demonstriert werden sollen. Die Anwendungen demonstrieren, wie ein Tracker und ein Geofence mithilfe einer Kombination aus Lambda AWS IoT - und Amazon Location-Funktionen interagieren.

Themen

- [Erstellen Sie Amazon-Standortressourcen für Ihre App](#)
- [Erstellen Sie eine Geofence-Sammlung](#)
- [Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung](#)
- [AWS Lambda mit MQTT verwenden](#)
- [App-Beispielcode einrichten](#)
- [Verwenden Sie die Beispiel-App](#)

Erstellen Sie Amazon-Standortressourcen für Ihre App

Zu Beginn müssen Sie die erforderlichen Amazon-Standortressourcen erstellen. Diese Ressourcen sind für die Funktionalität der Anwendung und die Ausführung der bereitgestellten Codefragmente unerlässlich.

Note

Wenn Sie noch kein AWS Konto erstellt haben, folgen Sie den Anweisungen im Benutzerleitfaden zur [AWS Kontoverwaltung](#).

Zunächst müssen Sie eine Amazon Cognito Identity Pool-ID erstellen. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie die [Amazon Cognito Cognito-Konsole](#) und wählen Sie im Menü auf der linken Seite Identitätspools und dann Identitätspool erstellen aus.
2. Vergewissern Sie sich, dass Gastzugriff aktiviert ist, und klicken Sie auf Weiter, um fortzufahren.
3. Erstellen Sie als Nächstes eine neue IAM-Rolle oder Verwenden Sie eine vorhandene IAM-Rolle.
4. Geben Sie einen Namen für den Identitätspool ein und stellen Sie sicher, dass Identity Pool Zugriff auf (geo) Amazon-Standortressourcen für die Karte und den Tracker hat, die Sie im nächsten Verfahren erstellen werden.

Als Nächstes müssen Sie in der AWS Amazon Location-Konsole eine Karte erstellen und gestalten. Gehen Sie dazu wie folgt vor:

1. Navigieren Sie in der Amazon Location-Konsole zum [Bereich Maps](#) und wählen Sie Create Map aus.
2. Geben Sie der neuen Karte einen Namen und eine Beschreibung. Notieren Sie sich den Namen, den Sie vergeben, so wie er später im Tutorial verwendet wird.
3. Beachten Sie bei der Auswahl eines Kartenstils den Kartendatenanbieter. Weitere Informationen finden Sie in Abschnitt 82 der [AWS Servicebedingungen](#).
4. Akzeptieren Sie die [Allgemeinen Geschäftsbedingungen von Amazon Location und](#) wählen Sie dann Create Map aus, um die Kartenerstellung abzuschließen.

Als Nächstes müssen Sie in der Amazon Location-Konsole einen Tracker erstellen. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie den [Bereich Maps](#) in der Amazon Location-Konsole.
2. Wählen Sie Tracker erstellen.
3. Füllen Sie die erforderlichen Felder aus. Notieren Sie sich den Namen des Trackers, auf den in diesem Tutorial verwiesen wird.
4. Wählen Sie im Feld Positionsfilerung die Option aus, die am besten zu Ihrer geplanten Nutzung Ihrer Tracker-Ressource passt. Wenn Sie die Positionsfilerung nicht einrichten, ist die Standardeinstellung TimeBased. Weitere Informationen finden Sie unter [Tracking starten](#) und [PositionFilerung](#) in der Amazon Location API-Referenz.
5. Wählen Sie Tracker erstellen, um die Erstellung des Trackers abzuschließen.

Erstellen Sie eine Geofence-Sammlung

Jetzt erstellen Sie eine Geofence-Sammlung. Sie können entweder die Konsole, die API oder die CLI verwenden. Die folgenden Verfahren führen Sie durch die einzelnen Optionen.

- Erstellen Sie mit der Amazon Location-Konsole eine Geofence-Sammlung:
 1. Öffnen Sie den Bereich [Geofence Collections](#) in der Amazon Location-Konsole.
 2. Wählen Sie Geofence-Sammlung erstellen.
 3. Geben Sie einen Namen und eine Beschreibung für die Sammlung an.
 4. Unter der EventBridge Regel mit Amazon CloudWatch als Ziel können Sie eine optionale EventBridge Regel erstellen, mit der Sie sofort auf Geofence-Ereignisse reagieren können. Dadurch kann Amazon Location Ereignisse veröffentlichen Amazon CloudWatch Logs.
 5. Klicken Sie auf Geofence-Sammlung erstellen, um die Erstellung der Sammlung abzuschließen.
- Erstellen Sie mit der Amazon Location API eine Geofence-Sammlung:

Verwenden Sie den [CreateGeofenceCollection](#)Vorgang über die Amazon Location Geofences-APIs. Im folgenden Beispiel wird eine API-Anfrage verwendet, um eine Geofence-Sammlung namens zu erstellen. *GEOCOLLECTION_NAME*

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Erstellen Sie eine Geofence-Sammlung mit AWS CLI-Befehlen:

Verwenden Sie den `create-geofence-collection`-Befehl. Das folgende Beispiel verwendet eine AWS CLI, um eine Geofence-Sammlung namens zu erstellen. *GEOCOLLECTION_NAME*
Weitere Informationen zur Verwendung der AWS CLI finden Sie in der [Dokumentation zur AWS Befehlszeilenschnittstelle](#).


```
aws location \  
  create-geofence-collection \  
  --collection-name "ExampleGeofenceCollection" \  
  --description "Shopping center geofence collection" \  
  --tags Tag1=Value1
```

Verknüpfen Sie einen Tracker mit einer Geofence-Sammlung

Um einen Tracker mit einer Geofence-Sammlung zu verknüpfen, können Sie entweder die Konsole, die API oder die CLI verwenden. Die folgenden Verfahren führen Sie durch die einzelnen Optionen.

Verknüpfen Sie mithilfe der Amazon Location Service Service-Konsole eine Tracker-Ressource mit einer Geofence-Sammlung:

1. Öffnen Sie die Amazon Location-Konsole.
2. Wählen Sie im linken Navigationsbereich Trackers aus.
3. Wählen Sie unter Device Trackers den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen die Option Geofence-Sammlung verknüpfen aus.
5. Wählen Sie im Fenster Verknüpfte Geofence-Sammlung eine Geofence-Sammlung aus dem Drop-down-Menü aus.
6. Wählen Sie Verknüpfen.
7. Nachdem Sie die Tracker-Ressource verknüpft haben, wird ihr der Status Aktiv zugewiesen.

Verknüpfen Sie mithilfe der Amazon Location APIs eine Tracker-Ressource mit einer Geofence-Sammlung:

Verwenden Sie den `AssociateTrackerConsumer` Vorgang über die Amazon Location Trackers APIs. Im folgenden Beispiel wird eine API-Anfrage verwendet, die mithilfe ihres Amazon-Ressourcennamens (ARN) `ExampleTracker` mit einer Geofence-Sammlung verknüpft wird.

```
POST /tracking/v0/trackers/ExampleTracker/consumers  
Content-type: application/json  
{  
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GEOCOLLECTION_NAME"
```

```
}
```

Verknüpfen Sie eine Tracker-Ressource mithilfe von AWS-CLI-Befehlen mit einer Geofence-Sammlung:

Verwenden Sie den `associate-tracker-consumer` -Befehl. Das folgende Beispiel verwendet eine AWS-CLI, um eine Geofence-Sammlung namens zu erstellen. *GEOCOLLECTION_NAME*

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GEOCOLLECTION_NAME" \
    --tracker-name "ExampleTracker"
```

AWS Lambda mit MQTT verwenden

Um eine Verbindung zwischen AWS IoT und Amazon Location herzustellen, benötigen Sie eine Lambda-Funktion zur Verarbeitung von Nachrichten, die durch EventBridge CloudWatch Ereignisse weitergeleitet werden. Diese Funktion extrahiert alle Positionsdaten, formatiert sie für Amazon Location und übermittelt sie über die Amazon Location Tracker-API.

Das folgende Verfahren zeigt Ihnen, wie Sie diese Funktion über die Lambda-Konsole erstellen:

1. Öffnen Sie die [-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste Funktionen aus.
3. Wählen Sie dann „Funktion erstellen“ und stellen Sie sicher, dass die Option „Von Grund auf neu erstellen“ ausgewählt ist.
4. geben Sie einen Funktionsnamen ein, und wählen Sie für die Runtime-Option Node.js 16.x aus.
5. Wählen Sie Funktion erstellen.
6. Öffnen Sie die Registerkarte Code, um auf den Editor zuzugreifen.
7. Überschreiben Sie den Platzhaltercode in der `index.js` Datei mit dem Folgenden:

```
const AWS = require('aws-sdk')
const iot = new AWS.Iot();
exports.handler = function(event) {
  console.log("event===>>>", JSON.stringify(event));
  var param = {
    endpointType: "iot:Data-ATS"
```

```

    };
    iot.describeEndpoint(param, function(err, data) {
      if (err) {
        console.log("error====>>>", err, err.stack); // an error occurred
      } else {
        var endp = data['endpointAddress'];
        const iotdata = new AWS.IotData({endpoint: endp});
        const trackerEvent = event["detail"]["EventType"];
        const src = event["source"];
        const time = event["time"];
        const gfId = event["detail"]["GeofenceId"];
        const resources = event["resources"][0];
        const splitResources = resources.split(".");
        const geofenceCollection = splitResources[splitResources.length -
1];

        const coordinates = event["detail"]["Position"];

        const deviceId = event["detail"]["DeviceId"];
        console.log("deviceId====>>>", deviceId);
        const msg = {
          "trackerEventType" : trackerEvent,
          "source" : src,
          "eventTime" : time,
          "geofenceId" : gfId,
          "coordinates": coordinates,
          "geofenceCollection": geofenceCollection
        };
        const params = {
          topic: `${deviceId}/tracker`,
          payload: JSON.stringify(msg),
          qos: 0
        };
        iotdata.publish(params, function(err, data) {
          if (err) {
            console.log("error====>>>", err, err.stack); // an error
occurred

          } else {
            console.log("Ladmbda triggered====>>>", trackerEvent); //
successful response
          }
        });
      }
    });
  });
}
});

```

```
}
```

8. Drücken Sie auf Deploy, um die aktualisierte Funktion zu speichern.
9. Öffnen Sie als Nächstes die Registerkarte Konfiguration.
10. Klicken Sie im Bereich Trigger auf die Schaltfläche Trigger hinzufügen.
11. Wählen Sie im Feld Quelle EventBridge (CloudWatch Ereignisse) aus.
12. Wählen Sie die Option Bestehende Regeln aus.
13. Geben Sie beispielsweise den Regelnamen einAmazonLocationMonitor-
GEOFENCECOLLECTION_NAME.
14. Drücken Sie die Schaltfläche Hinzufügen.
15. Dadurch werden auch ressourcenbasierte Richtlinienenerklärungen auf der Registerkarte „Berechtigungen“ angehängt

Jetzt richten Sie den AWS IoT MQTT-Testclient ein. Gehen Sie dabei wie folgt vor:

1. Öffnen Sie die Datei <https://console.aws.amazon.com/iot/>.
2. Wählen Sie im linken Navigationsbereich den MQTT-Testclient aus.
3. Sie sehen einen Abschnitt mit dem Titel MQTT-Testclient, in dem Sie Ihre MQTT-Verbindung konfigurieren können.
4. Nachdem Sie die erforderlichen Einstellungen konfiguriert haben, klicken Sie auf die Schaltfläche Connect, um mithilfe der bereitgestellten Parameter eine Verbindung zum MQTT-Broker herzustellen.
5. Notieren Sie den Endpunkt, so wie er später im Tutorial verwendet wird.

Sobald Sie mit dem Testclient verbunden sind, können Sie MQTT-Themen abonnieren oder Nachrichten zu Themen veröffentlichen, indem Sie die entsprechenden Eingabefelder verwenden, die in der MQTT-Testclient-Oberfläche bereitgestellt werden. Als Nächstes erstellen Sie eine AWS IoT Richtlinie.

6. Erweitern Sie im Menü auf der linken Seite unter Verwalten die Option Sicherheit und klicken Sie auf Richtlinien.
7. Klicken Sie auf die Schaltfläche Richtlinie erstellen.
8. Geben Sie den Namen einer Richtlinie ein.
9. Wählen Sie im Richtliniendokument die Registerkarte JSON aus.

10. Kopieren Sie die unten gezeigte Richtlinie und fügen Sie sie ein. Achten Sie jedoch darauf, alle Elemente mit Ihrem *REGION* und zu aktualisieren *ACCOUNT_ID*:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}",
        "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-identity.amazonaws.com:sub}/*",
        "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-identity.amazonaws.com:sub}/tracker"
      ],
      "Effect": "Allow"
    }
  ]
}
```

11. Wählen Sie die Schaltfläche Erstellen, um den Vorgang abzuschließen.

App-Beispielcode einrichten

Um den Beispielcode einzurichten, müssen Sie die folgenden Tools installiert haben:

- Git
- XCode 15.3 oder höher
- iOS Simulator 16 oder höher

Gehen Sie wie folgt vor, um den Beispiel-App-Code einzurichten:

1. Klonen Sie das Git-Repository von dieser URL: <https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications>.
2. Öffnen Sie die Projektdatei `AWSLocationSampleApp.xcodeproj`.
3. Warten Sie, bis der Prozess zur Paketauflösung abgeschlossen ist.
4. Benennen Sie im Projektnavigationsmenü in die folgenden Werte `ConfigTemplate.xcconfig` um `Config.xcconfig` und geben Sie sie ein:

```
IDENTITY_POOL_ID = `YOUR_IDENTITY_POOL_ID`  
MAP_NAME = `YOUR_MAP_NAME`  
TRACKER_NAME = `YOUR_TRACKER_NAME`  
WEBSOCKET_URL = `YOUR_MQTT_TEST_CLIENT_ENDPOINT`  
GEOFENCE_ARN = `YOUR_GEOFENCE_COLLECTION_NAME`
```

Verwenden Sie die Beispiel-App

Nachdem Sie den Beispielcode eingerichtet haben, können Sie die App nun auf einem iOS-Simulator oder einem physischen Gerät ausführen.

1. Erstellen Sie die App und führen Sie sie aus.
2. Die App fragt Sie nach Standort- und Benachrichtigungsberechtigungen. Sie müssen sie zulassen.
3. Drücken Sie die Cognito-Konfigurationstaste.
4. Speichern Sie die Konfiguration.
5. Sie können jetzt die Filteroptionen für Zeit, Entfernung und Genauigkeit sehen. Verwenden Sie sie nach Ihren Bedürfnissen.
6. Gehen Sie in der App zum Tab Tracking und Sie werden die Karte und die Schaltfläche „Tracking starten“ sehen.
7. Wenn Sie die App auf einem Simulator installiert haben, möchten Sie möglicherweise Standortänderungen simulieren. Dies kann unter Funktionen unter der Menüoption Standort erfolgen. Wählen Sie beispielsweise Funktionen, dann Standort und dann Freeway Drive aus.
8. Drücken Sie die Taste „Tracking starten“. Sie sollten die Tracking-Punkte auf der Karte sehen.
9. Die App verfolgt auch die Standorte im Hintergrund. Wenn Sie die App also im Hintergrund bewegen, werden Sie um Erlaubnis gebeten, die Verfolgung im Hintergrundmodus fortzusetzen.
10. Sie können das Tracking beenden, indem Sie auf die Schaltfläche Stop Tracking tippen.

Markieren Ihrer Amazon Location Service-Ressourcen

Verwenden Sie die Ressourcenmarkierung in Amazon Location, um Tags zu erstellen, um Ihre Ressourcen nach Zweck, Eigentümer, Umgebung oder Kriterien zu kategorisieren. Das Markieren Ihrer Ressourcen hilft Ihnen, Ihre Ressourcen zu verwalten, zu identifizieren, zu organisieren, zu suchen und zu filtern.

Mit können Sie beispielsweise Gruppen von AWS Ressourcen erstellen AWS Resource Groups, die auf einem oder mehreren Tags oder Teilen von Tags basieren. Sie können auch Gruppen auf der Grundlage ihres Vorkommens in einem AWS CloudFormation-Batch erstellen. Mit Ressourcengruppen und dem Tag-Editor können Sie Daten für Anwendungen konsolidieren und anzeigen, die aus mehreren Services, Ressourcen und Regionen bestehen. Weitere Informationen zu [allgemeinen Tagging-Strategien](#) finden Sie in der Allgemeinen AWS-Referenz.

Jedes Tag ist eine Bezeichnung, die aus einem Schlüssel und einem Wert besteht, den Sie definieren:

- Tag-Schlüssel – Eine allgemeine Bezeichnung, die die Tag-Werte kategorisiert. Beispiel: `CostCenter`
- Tag-Wert – Eine optionale Beschreibung für die Tag-Schlüsselkategorie. Beispiel: `MobileAssetTrackingResourcesProd`

Dieses Thema hilft Ihnen beim Einstieg in das Tagging, indem Sie die Tagging-Einschränkungen überprüfen. Es zeigt Ihnen auch, wie Sie Tags erstellen und Tags verwenden, um Ihre AWS Kosten für jedes aktive Tag mithilfe von Kostenzuordnungsberichten zu verfolgen.

Themen

- [Tagging-Einschränkungen](#)
- [Gewährt die Berechtigung zum Markieren von Ressourcen](#)
- [Hinzufügen eines Tags zu einer Amazon Location Service-Ressource](#)
- [Verfolgen der Ressourcenkosten nach Tag](#)
- [Steuern des Zugriffs auf Amazon Location Service-Ressourcen mithilfe von Tags](#)
- [Weitere Informationen](#)

Tagging-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Tags pro Ressource – 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.

Note

Wenn Sie einen neuen Tag (Markierung) mit demselben Tag (Markierung)-Schlüssel wie ein bestehender Tag (Markierung) hinzufügen, wird der bestehende Tag (Markierung) vom neuen überschrieben.

- Maximale Schlüssellänge: 128 Unicode-Zeichen in UTF-8
- Maximale Wertlänge: 256 Unicode-Zeichen in UTF-8
- Erlaubte Zeichen in Services sind: Buchstaben, Zahlen und Leerzeichen, die in UTF-8 darstellbar sind, und die folgenden Sonderzeichen: + - = . _ : / @.
- Bei Tag (Markierung)-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.
- Das Präfix `aws :` ist zur Verwendung in AWS reserviert. Wenn der Tag (Markierung) über einen Tag (Markierung)-Schlüssel mit diesem Präfix verfügt, können Sie den Schlüssel oder Wert des Tags (Markierung) nicht bearbeiten oder löschen. Tags mit dem `aws :` Präfix werden nicht auf Ihre Tags pro Ressourcenlimit angerechnet.

Gewährt die Berechtigung zum Markieren von Ressourcen

Sie können IAM-Richtlinien verwenden, um den Zugriff auf Ihre Amazon Location-Ressourcen zu steuern und die Berechtigung zum Markieren einer Ressource bei der Erstellung zu erteilen. Die Richtlinie kann nicht nur die Berechtigung zum Erstellen von Ressourcen erteilen, sondern auch Action Berechtigungen zum Zulassen von Tagging-Operationen enthalten:

- `geo:TagResource` – Ermöglicht es einem Benutzer, einer angegebenen Amazon Location-Ressource ein oder mehrere Tags zuzuweisen.
- `geo:UntagResource` – Ermöglicht es einem Benutzer, ein oder mehrere Tags aus einer angegebenen Amazon Location-Ressource zu entfernen.

- `geo:ListTagsForResource` – Ermöglicht einem Benutzer das Auflisten aller Tags, die einer Amazon Location-Ressource zugewiesen sind.

Im Folgenden finden Sie ein Richtlinienbeispiel, mit dem ein Benutzer eine Geofence-Sammlung erstellen und Ressourcen markieren kann:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
      "Effect": "Allow",
      "Action": [
        "geo:CreateGeofenceCollection",
        "geo:TagResource"
      ],
      "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    }
  ]
}
```

Hinzufügen eines Tags zu einer Amazon Location Service-Ressource

Sie können Tags hinzufügen, wenn Sie Ihre Ressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs erstellen:

- [Erstellen einer Kartenressource](#)
- [Erstellen einer Ortsindex-Ressource](#)
- [Erstellen einer Routenrechner-Ressource](#)
- [Erstellen einer Geofence-Sammlung](#)
- [Erstellen einer Tracker-Ressource](#)

Um vorhandene Ressourcen zu markieren, bearbeiten oder löschen Sie Tags

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich die Ressource aus, die Sie markieren möchten. Zum Beispiel Maps .
3. Wählen Sie eine Ressource aus der Liste aus.

4. Wählen Sie Tags verwalten, um Ihre Tags hinzuzufügen, zu bearbeiten oder zu löschen.

Verfolgen der Ressourcenkosten nach Tag

Sie können Tags für die Kostenzuordnung verwenden, um Ihre AWS Kosten im Detail zu verfolgen. Nachdem Sie die Kostenzuordnungs-Tags aktiviert haben, AWS verwendet die Kostenzuordnungs-Tags, um Ihre Ressourcenfakturierung in Ihrem Kostenzuordnungsbericht zu organisieren. Auf diese Weise können Sie Ihre Nutzungskosten kategorisieren und verfolgen.

Es gibt zwei Arten von Kostenzuordnungs-Tags, die Sie aktivieren können:

- [AWS-generiert](#) – Diese Tags werden von generiertAWS. AWS -Tags verwenden das `aws :` Präfix , z. B. `aws :createdBy`.
- [Benutzerdefiniert](#) – Dies sind benutzerdefinierte Tags, die Sie erstellen. Die benutzerdefinierten Tags verwenden das `user :` Präfix , z. B. `user :CostCenter`.

Sie müssen jeden Tag-Typ einzeln aktivieren. Nachdem Ihre Tags aktiviert wurden, können Sie Ihren monatlichen Kostenzuordnungsbericht [aktivieren AWS Cost Explorer](#) oder anzeigen.

AWS-generated tags

So aktivieren Sie von AWS generierte Tags

1. Öffnen Sie die Konsole für Fakturierung und Kostenmanagement unter <https://console.aws.amazon.com/billing/>.
2. Wählen Sie im linken Navigationsbereich Kostenzuordnungs-Tags aus.
3. Wählen Sie auf der Registerkarte Von AWSgenerierte Kostenzuordnungs-Tags die Tag-Schlüssel aus, die Sie aktivieren möchten.
4. Wählen Sie Activate.

User-defined tags

So aktivieren Sie benutzerdefinierte Tags

1. Öffnen Sie die Konsole für Fakturierung und Kostenmanagement unter <https://console.aws.amazon.com/billing/>.
2. Wählen Sie im linken Navigationsbereich Kostenzuordnungs-Tags aus.

3. Wählen Sie auf der Registerkarte Benutzerdefinierte Kostenzuordnungs-Tags die Tag-Schlüssel aus, die Sie aktivieren möchten.
4. Wählen Sie Activate.

Nachdem Sie Ihre Tags aktiviert haben, AWS generiert einen [monatlichen Kostenzuordnungsbericht](#) für Ihre Ressourcennutzung und -kosten. Dieser Kostenzuordnungsbericht enthält alle Ihre AWS Kosten für jeden Abrechnungszeitraum, einschließlich markierter und nicht markierter Ressourcen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing and Cost Management Leitfadens.

Steuern des Zugriffs auf Amazon Location Service-Ressourcen mithilfe von Tags

AWS Identity and Access Management (IAM)-Richtlinien unterstützen Tag-basierte Bedingungen, mit denen Sie die Autorisierung für Ihre Ressourcen basierend auf bestimmten Tag-Schlüsseln und -Werten verwalten können. Beispielsweise kann eine IAM-Rollenrichtlinie Bedingungen enthalten, um den Zugriff auf bestimmte Umgebungen, wie Entwicklung, Test oder Produktion, basierend auf Tags zu beschränken.

Weitere Informationen finden Sie im Thema zum [Steuern des Ressourcenzugriffs basierend auf Tags](#).

Weitere Informationen

Weitere Informationen über:

- Bewährte Methoden für die Markierung finden Sie unter [Markieren von AWS-Ressourcen](#) in der Allgemeinen AWS-Referenz.
- Informationen zum Verwenden von Tags zum Steuern des Zugriffs auf -AWSRessourcen finden Sie unter [Steuern des Zugriffs auf -AWSRessourcen mithilfe von Tags](#) im AWS Identity and Access Management -Benutzerhandbuch.

Gewähren des Zugriffs auf Amazon Location Service

Um Amazon Location Service verwenden zu können, muss einem Benutzer Zugriff auf die Ressourcen und APIs gewährt werden, aus denen Amazon Location besteht. Es gibt drei Strategien, mit denen Sie Zugriff auf Ihre -Ressourcen gewähren können.

- Verwenden von IAM – Um Benutzern Zugriff zu gewähren, die mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) authentifiziert sind, erstellen Sie eine IAM-Richtlinie, die den Zugriff auf die gewünschten Ressourcen ermöglicht. Weitere Informationen zu IAM und Amazon Location finden Sie unter [Identity and Access Management für Amazon Location Service](#).
- API-Schlüssel verwenden – Um nicht authentifizierten Benutzern Zugriff zu gewähren, können Sie API-Schlüssel erstellen, die schreibgeschützten Zugriff auf Ihre Amazon Location Service-Ressourcen gewähren. Dies ist in Fällen nützlich, in denen Sie nicht jeden Benutzer authentifizieren möchten. Zum Beispiel eine Webanwendung. Weitere Informationen zu API-Schlüsseln finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).
- Amazon Cognito verwenden – Eine Alternative zu API-Schlüsseln ist die Verwendung von Amazon Cognito, um anonymen Zugriff zu gewähren. Mit Amazon Cognito können Sie eine umfassendere Autorisierung mit einer Richtlinie erstellen, um zu definieren, was von nicht authentifizierten Benutzern getan werden kann. Weitere Informationen zur Verwendung von Amazon Cognito finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

Note

Sie können Amazon Cognito auch verwenden, um Ihren eigenen Authentifizierungsprozess zu verwenden oder mehrere Authentifizierungsmethoden mithilfe von Amazon Cognito Federated Identities zu kombinieren. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Themen

- [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#)
- [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#)

Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln

Wenn Sie Amazon Location Service APIs in Ihren Anwendungen aufrufen, führen Sie diesen Aufruf normalerweise als authentifizierter Benutzer aus, der zum Ausführen der API-Aufrufe autorisiert ist. Es gibt jedoch einige Fälle, in denen Sie nicht jeden Benutzer Ihrer Anwendung

authentifizieren möchten. Beispielsweise möchten Sie vielleicht, dass eine Webanwendung, die Ihren Standort anzeigt, jedem, der die Website verwendet, zur Verfügung steht, unabhängig davon, ob er angemeldet ist oder nicht. In diesem Fall besteht eine Alternative darin, API-Schlüssel für die API-Aufrufe zu verwenden.

API-Schlüssel sind ein Schlüsselwert, der bestimmten Amazon Location Service-Ressourcen in Ihrem zugeordnet ist AWS-Konto, sowie bestimmten Aktionen, die Sie für diese Ressourcen ausführen können. Sie können einen API-Schlüssel in Ihrer Anwendung verwenden, um nicht authentifizierte Aufrufe an die Amazon Location APIs für diese Ressourcen zu tätigen. Wenn Sie beispielsweise einen API-Schlüssel mit der Kartenressource myMap und den GetMap* Aktionen verknüpfen, kann eine Anwendung, die diesen API-Schlüssel verwendet, Karten anzeigen, die mit dieser Ressource erstellt wurden, und Ihr Konto wird als jede andere Nutzung von Ihrem Konto belastet. Derselbe API-Schlüssel würde keine Berechtigungen zum Ändern oder Aktualisieren der Kartenressource erteilen – nur die Verwendung der Ressource ist zulässig.

Note

API-Schlüssel können nur mit Karten-, Orts- und Routenressourcen verwendet werden, und Sie können diese Ressourcen nicht ändern oder erstellen. Wenn Ihre Anwendung Zugriff auf andere Ressourcen oder Aktionen für nicht authentifizierte Benutzer benötigt, können Sie Amazon Cognito verwenden, um Zugriff zusammen mit oder anstelle von API-Schlüsseln zu gewähren. Weitere Informationen finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

API-Schlüssel enthalten einen Klartextwert, der Zugriff auf eine oder mehrere Ressourcen in Ihrem gewährt AWS-Konto. Wenn jemand Ihren API-Schlüssel kopiert, kann er auf dieselben Ressourcen zugreifen. Um dies zu vermeiden, können Sie die Domänen angeben, in denen der API-Schlüssel beim Erstellen des Schlüssels verwendet werden kann. Diese Domains werden als Referenten bezeichnet. Bei Bedarf können Sie auch kurzfristige API-Schlüssel erstellen, indem Sie Ablaufzeiten für Ihre API-Schlüssel festlegen.

Themen

- [API-Schlüssel im Vergleich zu Amazon Cognito](#)
- [Erstellen von API-Schlüsseln](#)
- [Verwenden eines API-Schlüssels zum Aufrufen einer Amazon Location API](#)
- [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#)

- [Verwalten der Lebensdauer von API-Schlüsseln](#)

API-Schlüssel im Vergleich zu Amazon Cognito

API-Schlüssel und Amazon Cognito werden auf ähnliche Weise für ähnliche Szenarien verwendet. Warum würden Sie also einen gegenüber dem anderen verwenden? In der folgenden Liste werden einige der Unterschiede zwischen den beiden hervorgehoben.

- API-Schlüssel sind nur für Karten-, Orts- und Routenressourcen sowie nur für bestimmte Aktionen verfügbar. Amazon Cognito kann verwendet werden, um den Zugriff auf die meisten Amazon Location Service APIs zu authentifizieren.
- Die Leistung von Kartenanforderungen mit API-Schlüsseln ist in der Regel schneller als ähnliche Szenarien mit Amazon Cognito. Eine einfachere Authentifizierung bedeutet weniger Roundtrips zum Service und zwischengespeicherte Anforderungen, wenn innerhalb eines kurzen Zeitraums wieder dieselbe Kartenkachel abgerufen wird.
- Mit Amazon Cognito können Sie Ihren eigenen Authentifizierungsprozess verwenden oder mehrere Authentifizierungsmethoden kombinieren, indem Sie Amazon Cognito Federated Identities verwenden. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Erstellen von API-Schlüsseln

Sie können einen API-Schlüssel erstellen und ihn einer oder mehreren Ressourcen in Ihrem zugeordneten AWS-Konto zuordnen.

Sie können einen API-Schlüssel mithilfe der Amazon Location Service-Konsole, der AWS CLI oder der Amazon Location APIs erstellen.

Console

So erstellen Sie einen API-Schlüssel mit der Amazon Location Service-Konsole

1. Wählen Sie in der [Amazon Location-Konsole](#) im linken Menü API-Schlüssel aus.
2. Wählen Sie auf der Seite API-Schlüssel die Option API-Schlüssel erstellen aus.
3. Geben Sie auf der Seite API-Schlüssel erstellen die folgenden Informationen ein:
 - Name – Ein Name für Ihren API-Schlüssel, z. B. MyWebAppKey.

- Beschreibung – Eine optionale Beschreibung für Ihren API-Schlüssel.
 - Ressourcen – Wählen Sie die Amazon Location-Ressourcen aus, auf die Sie mit diesem API-Schlüssel Zugriff gewähren möchten, aus der Dropdown-Liste aus. Sie können mehr als eine Ressource hinzufügen, indem Sie Ressource hinzufügen auswählen.
 - Aktionen – Geben Sie die Aktionen an, die Sie mit diesem API-Schlüssel autorisieren möchten. Sie müssen mindestens eine Aktion auswählen, die jedem ausgewählten Ressourcentyp entspricht. Wenn Sie beispielsweise eine Ortsressource ausgewählt haben, müssen Sie mindestens eine der Optionen unter Ortsaktionen auswählen.
 - Ablaufzeit – Fügen Sie optional ein Ablaufdatum und eine Ablaufzeit für Ihren API-Schlüssel hinzu. Weitere Informationen finden Sie unter [Verwalten der Lebensdauer von API-Schlüsseln](#).
 - Referenten – Fügen Sie optional eine oder mehrere Domains hinzu, in denen Sie den API-Schlüssel verwenden können. Wenn der API-Schlüssel beispielsweise eine Anwendung zulassen soll, die auf der Website ausgeführt wird `example.com`, können Sie `*.example.com/` als zulässigen Referer verwenden.
 - Tags – Fügen Sie dem API-Schlüssel optional Tags hinzu.
4. Wählen Sie API-Schlüssel erstellen, um den API-Schlüssel zu erstellen.
 5. Auf der Detailseite für den API-Schlüssel finden Sie Informationen zu dem von Ihnen erstellten API-Schlüssel. Wählen Sie API-Schlüssel anzeigen, um den Schlüsselwert anzuzeigen, den Sie beim Aufrufen von Amazon Location APIs verwenden. Der Schlüsselwert hat das Format `v1.public.a1b2c3d4...`. Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#).

API

So erstellen Sie einen API-Schlüssel mit den Amazon Location APIs

Verwenden Sie die [CreateKey](#) Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen eines API-Schlüssels *ExampleKey* namens ohne Ablaufdatum und Zugriff auf eine einzelne Kartenressource.

```
POST /metadata/v0/keys HTTP/1.1
Content-type: application/json
```

```
{
  "KeyName": "ExampleKey"
  "Restrictions": {
    "AllowActions": [
      "geo:GetMap*"
    ],
    "AllowResources": [
      "arn:aws:geo:region:map/mapname"
    ]
  },
  "NoExpiry": true
}
```

Die Antwort enthält den API-Schlüsselwert, der beim Zugriff auf Ressourcen in Ihren Anwendungen verwendet werden soll. Der Schlüsselwert hat das Format `v1.public.a1b2c3d4...`. Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#).

Sie können auch die [DescribeKey](#)-API verwenden, um den Schlüsselwert für einen Schlüssel zu einem späteren Zeitpunkt zu finden.

AWS CLI

So erstellen Sie einen API-Schlüssel mit AWS CLI -Befehlen

Verwenden Sie den [create-key](#)-Befehl.

Im folgenden Beispiel wird ein API-Schlüssel namens *ExampleKey* ohne Ablaufdatum und Zugriff auf eine einzelne Kartenressource erstellt.

```
aws location \
  create-key \
  --key-name ExampleKey \
  --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":
["arn:aws:geo:region:map/mapname"]}' \
  --no-expiry
```

Die Antwort enthält den API-Schlüsselwert, der beim Zugriff auf Ressourcen in Ihren Anwendungen verwendet werden soll. Der Schlüsselwert hat das Format

v1.public.a1b2c3d4... Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#). Die Antwort auf create-key sieht wie folgt aus.

```
{
  "Key": "v1.public.a1b2c3d4...",
  "KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",
  "KeyName": "ExampleKey",
  "CreateTime": "2023-02-06T22:33:15.693Z"
}
```

Sie können auch verwendetes describe-key, um den Schlüsselwert zu einem späteren Zeitpunkt zu finden. Das folgende Beispiel zeigt, wie Sie describe-key für einen API-Schlüssel mit dem Namen aufrufen *ExampleKey*.

```
aws location describe-key \
  --key-name ExampleKey
```

Verwenden eines API-Schlüssels zum Aufrufen einer Amazon Location API

Nachdem Sie einen API-Schlüssel erstellt haben, können Sie den Schlüsselwert verwenden, um Aufrufe an Amazon Location APIs in Ihrer Anwendung zu tätigen.

Die APIs, die API-Schlüssel unterstützen, verfügen über einen zusätzlichen Parameter, der den API-Schlüsselwert verwendet. Wenn Sie beispielsweise die GetPlace-API aufrufen, können Sie den [Schlüsselparameter](#) wie folgt ausfüllen:

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyValue
```

Wenn Sie diesen Wert eingeben, müssen Sie den API-Aufruf nicht wie gewohnt mit AWS Sig v4 authentifizieren.

Für JavaScript Entwickler können Sie Amazon Location verwenden, [JavaScript Authentifizierungshelfer](#) um die Authentifizierung von API-Operationen mit API-Schlüsseln zu unterstützen.

Für Entwickler von Mobilgeräten können Sie die folgenden Amazon Location SDKs für die mobile Authentifizierung verwenden:

- [Amazon Location Service Mobile Authentication SDK für iOS](#)
- [Amazon Location Service Mobile Authentication SDK für Android](#)

Wenn Sie den `---keyParameter` verwenden, sollten Sie für AWS CLI Benutzer auch den `---no-sign-requestParameter` verwenden, um das Signieren mit Sig v4 zu vermeiden.

Note

Wenn Sie sowohl eine `key` als auch eine AWS Sig v4-Signatur in einen Aufruf an Amazon Location Service aufnehmen, wird nur der API-Schlüssel verwendet.

Verwenden eines API-Schlüssels zum Rendern einer Zuordnung

Sie können den API-Schlüsselwert verwenden, um eine Zuordnung in Ihrer Anwendung mithilfe von `MapLibre` zu rendern. Dies unterscheidet sich etwas von der Verwendung der API-Schlüssel in anderen Amazon Location APIs, die Sie direkt aufrufen, da diese Aufrufe für Sie `MapLibre` vornimmt.

Der folgende Beispielcode zeigt die Verwendung des `-API-Schlüssels` zum Rendern einer Karte auf einer einfachen Webseite mithilfe der `MapLibre microSD JS-Kartensteuerung`. Damit dieser Code ordnungsgemäß funktioniert, ersetzen Sie die *ExampleMap* Zeichenfolgen `v1.public.your-api-key-value`, `us-east-1` und durch Werte, die Ihrem entsprechen AWS-Konto.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body { margin: 0; }
      #map { height: 100vh; }
    </style>
  </head>
  <body>
    <!-- Map container -->
    <div id="map" />
    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.js"></script>
    <script>
      const apiKey = "v1.public.your-api-key-value"; // API key
```

```
const region = "us-east-1"; // Region
const mapName = "ExampleMap"; // Map name
// URL for style descriptor
const styleUrl = `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/
${mapName}/style-descriptor?key=${apiKey}`;
// Initialize the map
const map = new maplibregl.Map({
  container: "map",
  style: styleUrl,
  center: [-123.1187, 49.2819],
  zoom: 11,
});
map.addControl(new maplibregl.NavigationControl(), "top-left");
</script>
</body>
</html>
```

Verwalten der Lebensdauer von API-Schlüsseln

Sie können API-Schlüssel erstellen, die auf unbestimmte Zeit funktionieren. Wenn Sie jedoch einen temporären API-Schlüssel erstellen, API-Schlüssel regelmäßig rotieren oder einen vorhandenen API-Schlüssel widerrufen möchten, können Sie den Ablauf von API-Schlüsseln verwenden.

Wenn Sie einen neuen API-Schlüssel erstellen oder einen vorhandenen aktualisieren, können Sie die Ablaufzeit für diesen API-Schlüssel festlegen.

- Wenn ein API-Schlüssel seine Ablaufzeit erreicht hat, wird der Schlüssel automatisch deaktiviert. Inaktive Schlüssel können nicht mehr verwendet werden, um Kartenanforderungen zu stellen.
- Sie können einen API-Schlüssel 90 Tage nach der Deaktivierung löschen.
- Wenn Sie einen inaktiven Schlüssel haben, den Sie noch nicht gelöscht haben, können Sie ihn wiederherstellen, indem Sie die Ablaufzeit auf einen zukünftigen Zeitpunkt aktualisieren.
- Um einen permanenten Schlüssel zu erstellen, können Sie die Ablaufzeit entfernen.
- Wenn Sie versuchen, einen API-Schlüssel zu deaktivieren, der innerhalb der letzten 7 Tage verwendet wurde, werden Sie aufgefordert, zu bestätigen, dass Sie die Änderung vornehmen möchten. Wenn Sie die Amazon Location Service API oder die verwenden AWS CLI, erhalten Sie eine Fehlermeldung, es sei denn, Sie setzen den `ForceUpdate` Parameter auf „true“.

Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito

Sie können die Amazon Cognito-Authentifizierung als Alternative zur direkten Verwendung von AWS Identity and Access Management (IAM) sowohl mit Frontend-SDKs als auch mit direkten HTTPS-Anfragen verwenden.

Sie können diese Form der Authentifizierung aus folgenden Gründen verwenden:

- Nicht authentifizierte Benutzer – Wenn Sie eine Website mit anonymen Benutzern haben, können Sie Amazon Cognito-Identitätspools verwenden. Weitere Informationen finden Sie im Abschnitt auf [the section called “Verwenden von Amazon Cognito”](#).
- Ihre eigene Authentifizierung – Wenn Sie Ihren eigenen Authentifizierungsprozess verwenden oder mehrere Authentifizierungsmethoden kombinieren möchten, können Sie Amazon Cognito Federated Identities verwenden. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Amazon Cognito bietet Authentifizierung, Autorisierung und Benutzerverwaltung für Web- und mobile Apps. Sie können nicht authentifizierte Amazon Cognito-Identitätspools mit Amazon Location verwenden, um Anwendungen zu ermöglichen, temporäre, begrenzte AWS Anmeldeinformationen abzurufen.

Weitere Informationen finden Sie unter [Erste Schritte mit Benutzerpools](#) im Amazon Cognito-Entwicklerhandbuch.

Note

Für Entwickler von Mobilgeräten stellt Amazon Location SDKs für die mobile Authentifizierung sowohl für iOS als auch für Android bereit. Weitere Informationen finden Sie in den folgenden Github-Repositorys:

- [Amazon Location Service Mobile Authentication SDK für iOS](#)
- [Amazon Location Service Mobile Authentication SDK für Android](#)

Amazon-Cognito-Identitätspool erstellen

Sie können Amazon Cognito-Identitätspools erstellen, um nicht authentifizierten Gastzugriff auf Ihre Anwendung über die Amazon Cognito-Konsole, die AWS CLI oder die Amazon Cognito-APIs zu ermöglichen.

Important

Der von Ihnen erstellte Pool muss sich in demselben AWS-Konto und derselben AWS Region befinden wie die von Ihnen verwendeten Amazon Location Service-Ressourcen.

Sie können IAM-Richtlinien, die nicht authentifizierten Identitätsrollen zugeordnet sind, mit den folgenden Aktionen verwenden:

- `geo:GetMap*`
- `geo:SearchPlaceIndex*`
- `geo:GetPlace`
- `geo:CalculateRoute*`
- `geo:GetGeofence`
- `geo:ListGeofences`
- `geo:PutGeofence`
- `geo:BatchDeleteGeofence`
- `geo:BatchPutGeofence`
- `geo:BatchEvaluateGeofences`
- `geo:GetDevicePosition*`
- `geo:ListDevicePositions`
- `geo:BatchDeleteDevicePositionHistory`
- `geo:BatchGetDevicePosition`
- `geo:BatchUpdateDevicePosition`

Das Einschließen anderer Amazon Location-Aktionen hat keine Auswirkungen und nicht authentifizierte Identitäten können sie nicht aufrufen.

Example

So erstellen Sie einen Identitäten-Pool mit der Amazon Cognito-Konsole

1. Melden Sie sich bei der [Amazon-Cognito-Konsole](#) an.
2. Klicken Sie auf Manage Identity Pools (Identitäten-Pools verwalten).
3. Wählen Sie Neuen Identitäten-Pool erstellen und geben Sie dann einen Namen für Ihren Identitäten-Pool ein.
4. Wählen Sie im Abschnitt Einklappbare nicht authentifizierte Identitäten die Option Zugriff auf nicht authentifizierte Identitäten aktivieren aus.
5. Wählen Sie Pool erstellen.
6. Wählen Sie aus, welche IAM-Rollen Sie mit Ihrem Identitäten-Pool verwenden möchten.
7. Erweitern Sie Details anzeigen.
8. Geben Sie unter Nicht authentifizierte Identitäten einen Rollennamen ein.
9. Erweitern Sie den Abschnitt Richtliniendokument anzeigen und wählen Sie dann Bearbeiten, um Ihre Richtlinie hinzuzufügen.
10. Fügen Sie Ihre Richtlinie hinzu, um Zugriff auf Ihre -Ressourcen zu gewähren.

Im Folgenden finden Sie Richtlinienbeispiele für Karten, Orte, Tracker und Routen. Um die Beispiele für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die Platzhalter *region* und *accountID*:

Maps policy example

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Kartenressource mit dem Namen *ExampleMap*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapStyleDescriptor",
        "geo:GetMapGlyphs",
        "geo:GetMapSprites",
        "geo:GetMapTile"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
  }
]
}

```

Durch das Hinzufügen einer übereinstimmenden [IAM-Bedingung](#) `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Das folgende Beispiel erlaubt den Zugriff auf eine Kartenressource namens nur `RasterEsriImagery` von der Website `ausexample.com`:

Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetMap*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:map/
RasterEsriImagery",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

Wenn Sie [Tangram verwenden](#), um eine Karte anzuzeigen, verwendet es nicht die Stildeskriptoren, Glyphs oder -Sprites, die von der Maps-API zurückgegeben werden. Stattdessen wird sie konfiguriert, indem auf eine ZIP-Datei verwiesen wird, die Stilregeln und erforderliche Komponenten enthält. Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Kartenressource mit dem Namen *ExampleMap* für die `GetMapTileOperation`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile"
      ],
      "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
    }
  ]
}

```

Places policy example

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Ortsindexressource mit dem Namen *ExamplePlaceIndex* um nach Orten nach Text oder Positionen zu suchen.

```


{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PlacesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndex*",
        "geo:GetPlace"
      ],
      "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
    }
  ]
}

```



```
    }  
  ]  
}
```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Im folgenden Beispiel wird der Zugriff auf eine Ortsindexressource mit dem Namen *ExamplePlaceIndex* von allen verweisenden Websites mit Ausnahme von `verweigertexample.com`.

 Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "geo:*",  
      "Resource": "arn:aws:geo:us-west-2:111122223333:place-  
index/ExamplePlaceIndex",  
      "Condition": {  
        "StringLike": {  
          "aws:referer": [  
            "https://example.com/*",  
            "https://www.example.com/*"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Trackers policy example

Die folgende Richtlinie gewährt Zugriff auf eine Tracker-Ressource mit dem Namen , um Gerätepositionen *ExampleTracker* zu aktualisieren.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePosition",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
    }
  ]
}
```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referrer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Im folgenden Beispiel wird der Zugriff auf eine Tracker-Ressource mit dem Namen *ExampleTracker* von allen verweisenden Websites mit Ausnahme von `verweigertexample.com`.

Warning

Obwohl den Zugriff einschränken `aws:referrer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referrer`-Wert ihrer Wahl bereitstellen. Daher `aws:referrer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#) .


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetDevice*",
      "Resource": "arn:aws:geo:us-
west-2:111122223333:tracker/ExampleTracker",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Routes policy example

Die folgende Richtlinie gewährt Zugriff auf eine Routenrechner-Ressource namens , um eine Route *ExampleCalculator* zu berechnen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:region:accountID:route-
calculator/ExampleCalculator"
    }
  ]
}
```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Das folgende Beispiel verweigert den Zugriff auf einen Routenrechner mit dem Namen *ExampleCalculator* von allen verweisenden Websites mit Ausnahme von `example.com`.

 Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:route-
calculator/ExampleCalculator",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Note

Während nicht authentifizierte Identitätspools für die Offenlegung auf ungesicherten Websites vorgesehen sind, beachten Sie, dass sie gegen standardmäßige, zeitlich begrenzte AWS Anmeldeinformationen ausgetauscht werden.

Es ist wichtig, die IAM-Rollen, die nicht authentifizierten Identitätspools zugeordnet sind, entsprechend einzugrenzen.

11. Wählen Sie Zulassen, um Ihre Identitätspools zu erstellen.

Der resultierende Identitätspool folgt der Syntax `<region>:<GUID>`.

Beispielsweise:

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Weitere Beispiele für Amazon Location finden Sie unter [the section called “Beispiele für identitätsbasierte Richtlinien”](#).

Verwenden der Amazon Cognito-Identitätspools in JavaScript

Im folgenden Beispiel wird der von Ihnen erstellte nicht authentifizierte Identitätspool gegen Anmeldeinformationen ausgetauscht, die dann zum Abrufen des Stildeskriptors für Ihre Kartenressource verwendet werden *ExampleMap*.

```
const AWS = require("aws-sdk");

const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: "<identity pool ID>" // for example, us-east-1:1sample4-5678-90ef-
  aaaa-1234abcd56ef
});

const client = new AWS.Location({
  credentials,
  region: AWS.config.region || "<region>"
});

console.log(await client.getMapStyleDescriptor("ExampleMap").promise());
```

Note

Abrufen von Anmeldeinformationen von nicht authentifizierten Identitäten sind eine Stunde lang gültig.

Im Folgenden finden Sie ein Beispiel für eine Funktion, die Anmeldeinformationen automatisch erneuert, bevor sie ablaufen.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Um diese Arbeit zu vereinfachen, können Sie den Amazon Location verwenden [JavaScript Authentifizierungshelfer](#). Dies ersetzt sowohl das Abrufen der Anmeldeinformationen als auch deren Aktualisierung. In diesem Beispiel wird das AWS SDK für JavaScript v3 verwendet.

```
import { LocationClient, GetMapStyleDescriptorCommand } from "@aws-sdk/client-location";
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

const identityPoolId = "<identity pool ID>"; // for example, us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef

// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "<region>", // The region containing both the identity pool and tracker resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make requests to Amazon Location
});

const input = {
  MapName: "ExampleMap",
};

const command = new GetMapStyleDescriptorCommand(input);
```

```
console.log(await client.send(command));
```

Nächste Schritte

- Um Ihre Rollen zu ändern, wechseln Sie zur [IAM-Konsole](#) .
- Um Ihre Identitätspools zu verwalten, gehen Sie zur [Amazon Cognito-Konsole](#) .

Überwachen von Amazon Location Service

Wenn Sie Amazon Location Service verwenden, können Sie Ihre Nutzung und Ressourcen im Laufe der Zeit überwachen, indem Sie Folgendes verwenden:

- Amazon CloudWatch. Überwacht Ihre Amazon Location Service-Ressourcen und stellt Metriken mit Statistiken nahezu in Echtzeit bereit.
- AWS CloudTrail. Stellt die Ereignisnachverfolgung aller Aufrufe an Amazon Location Service APIs bereit.

Dieser Abschnitt enthält Informationen zur Verwendung dieser Services.

Themen

- [Überwachen von Amazon Location Service mit Amazon CloudWatch](#)
- [Protokollierung und Überwachung mit AWS CloudTrail](#)

Überwachen von Amazon Location Service mit Amazon CloudWatch

Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie ausgeführt werden, nahezu AWS in Echtzeit. Sie können Amazon Location-Ressourcen mit überwachen CloudWatch, das Rohdaten sammelt und Metriken nahezu in Echtzeit in aussagekräftige Statistiken verarbeitet. Sie können historische Informationen für bis zu 15 Monate anzeigen oder Metriken durchsuchen, die Sie in der Amazon- CloudWatch Konsole anzeigen können, um mehr über Ihre Amazon Location-Ressourcen zu erfahren. Sie können Alarmer auch einstellen, indem Sie Schwellenwerte definieren, Benachrichtigungen senden oder Maßnahmen ergreifen, wenn diese Schwellenwerte erreicht werden.

Weitere Informationen finden Sie im [Amazon CloudWatch -Benutzerhandbuch](#).

Themen

- [Metriken von Amazon Location Service, die nach Amazon exportiert werden CloudWatch](#)
- [Anzeigen von Amazon Location Service-Metriken](#)
- [Erstellen von CloudWatch Alarmen für Amazon Location Service-Metriken](#)
- [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#)
- [CloudWatch -Metrikbeispiele für Amazon Location Service](#)

Metriken von Amazon Location Service, die nach Amazon exportiert werden CloudWatch

Metriken sind zeitgeordnete Datenpunkte, die nach exportiert werden CloudWatch. Eine Dimension ist ein Name-Wert-Paar, das die Metrik identifiziert. Weitere Informationen finden Sie unter [Verwenden von CloudWatch Metriken](#) und [CloudWatch](#) Dimensionen im Amazon CloudWatch - Benutzerhandbuch.

Im Folgenden finden Sie Metriken, die Amazon Location Service CloudWatch in im AWS/Location Namespace exportiert.

Metrik	Beschreibung
CallCount	<p>Die Anzahl der Aufrufe an einen bestimmten API-Endpunkt.</p> <p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistiken: Summe</p> <p>Einheiten: Anzahl</p>
ErrorCount	<p>Die Anzahl der Fehlerantworten von Aufrufen an einen bestimmten API-Endpunkt.</p> <p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistiken: Summe</p> <p>Einheiten: Anzahl</p>
SuccessCount	<p>Die Anzahl der erfolgreichen Aufrufe an einen bestimmten API-Endpunkt.</p>

Metrik	Beschreibung
	<p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistiken: Summe</p> <p>Einheiten: Anzahl</p>
CallLatency	<p>Die Zeit, die der Vorgang benötigt, um zu verarbeiten und eine Antwort zurückzugeben, wenn ein Aufruf an einen bestimmten API-Endpunkt erfolgt.</p> <p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistik: Durchschnitt</p> <p>Einheiten: Millisekunden</p>

Anzeigen von Amazon Location Service-Metriken

Sie können Metriken für Amazon Location Service in der Amazon CloudWatch Konsole oder mithilfe der Amazon CloudWatch API anzeigen.

So zeigen Sie Metriken mit der CloudWatch Konsole an

Example

1. Öffnen Sie die - CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie auf der Registerkarte Alle Metriken den Amazon Location-Namespace aus.
4. Wählen Sie den Metriktyp aus, der angezeigt werden soll.
5. Wählen Sie die Metrik aus und fügen Sie sie dem Diagramm hinzu.

Weitere Informationen finden Sie unter [Anzeigen verfügbarer Metriken](#) im Amazon- CloudWatch Benutzerhandbuch.

Erstellen von CloudWatch Alarmen für Amazon Location Service-Metriken

Sie können verwenden CloudWatch , um Alarme für Ihre Amazon Location Service-Metriken festzulegen. Sie können beispielsweise einen Alarm in erstellen, CloudWatch um eine E-Mail zu senden, wenn ein Anstieg der Fehleranzahl auftritt.

Die folgenden Themen geben Ihnen eine grobe Übersicht darüber, wie Sie die Alarme mithilfe von CloudWatch einrichten können. Detaillierte Anweisungen finden Sie unter [Verwenden von Alarmen](#) im Amazon- CloudWatch Benutzerhandbuch.

So richten Sie Alarme mithilfe der CloudWatch Konsole ein

Example

1. Öffnen Sie die - CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Alarm aus.
3. Wählen Sie Alarm erstellen aus.
4. Wählen Sie Select metric (Metrik auswählen) aus.
5. Wählen Sie auf der Registerkarte Alle Metriken den Amazon Location-Namespace aus.
6. Wählen Sie eine Metrikkategorie aus.
7. Suchen Sie die Zeile mit der Metrik, für die Sie einen Alarm erstellen möchten, und aktivieren Sie dann das Kontrollkästchen neben dieser Zeile.
8. Wählen Sie Select metric (Metrik auswählen) aus.
9. Geben Sie unter Metrik die Werte ein.
10. Geben Sie die Alarmbedingungen an.
11. Wählen Sie Weiter aus.
12. Wenn Sie eine Benachrichtigung senden möchten, wenn die Alarmbedingungen erfüllt sind:
 - Wählen Sie unter Alarmstatusauslöser den Alarmstatus aus, um eine Benachrichtigung zum Senden anzufordern.
 - Wählen Sie unter SNS-Thema auswählen die Option Neues Thema erstellen aus, um ein neues Amazon Simple Notification Service (Amazon SNS)-Thema zu erstellen. Geben Sie den Themennamen und die E-Mail-Adresse ein, an die die Benachrichtigung gesendet werden soll.
 - Unter Benachrichtigung senden, um zusätzliche E-Mail-Adressen einzugeben, an die die Benachrichtigung gesendet werden soll.
 - Wählen Sie Add notification (Benachrichtigung hinzufügen) aus. Diese Liste wird gespeichert und erscheint für künftige Alarme in der Liste.

13. Klicken Sie auf Weiter, wenn Sie fertig sind.

14. Geben Sie einen Namen und eine Beschreibung für den Alarm ein und wählen Sie dann Weiter aus.

15. Bestätigen Sie die Alarmdetails und wählen Sie dann Weiter aus.

Note

Wenn Sie ein neues Amazon SNS-Thema erstellen, müssen Sie die E-Mail-Adresse verifizieren, bevor eine Benachrichtigung gesendet werden kann. Wenn die E-Mail nicht verifiziert wurde, wird die Benachrichtigung nicht empfangen, wenn ein Alarm durch eine Statusänderung ausgelöst wird.

Weitere Informationen zum Einrichten von Alarmen mithilfe der CloudWatch Konsole finden Sie unter [Erstellen eines Alarms, der E-Mails sendet](#) im Amazon CloudWatch-Benutzerhandbuch.

Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten

Sie können Amazon- CloudWatch Alarme erstellen, um Sie zu benachrichtigen, wenn Ihre Auslastung eines bestimmten Kontingents einen konfigurierbaren Schwellenwert überschreitet. Auf diese Weise können Sie erkennen, wann Sie sich in der Nähe Ihrer Kontingentlimits befinden, und entweder Ihre Auslastung anpassen, um Kostenüberschreitungen zu vermeiden, oder bei Bedarf eine Kontingenterhöhung beantragen. Informationen zur Verwendung von CloudWatch zur Überwachung von Kontingenten finden Sie unter [Visualisieren Ihrer Servicekontingente und Einstellen von Alarmen](#) im Amazon- CloudWatch Benutzerhandbuch.

CloudWatch -Metrikbeispiele für Amazon Location Service

Sie können die [GetMetricData](#)-API verwenden, um Metriken für Amazon Location abzurufen.

- Sie können beispielsweise einen Alarm für den Fall überwachen `CallCount` und einstellen, dass eine Verringerung der Zahl auftritt.

Die Überwachung der `CallCount` Metriken für `SendDeviceLocation` kann Ihnen helfen, einen Überblick über verfolgte Komponenten zu erhalten. Wenn die `CallCount` abnimmt, bedeutet dies, dass nachverfolgte Komponenten, wie z. B. eine Flotte von Lkw, ihre aktuellen Standorte nicht mehr gesendet haben. Wenn Sie dafür einen Alarm einrichten, können Sie sich benachrichtigen lassen, dass ein Problem aufgetreten ist.

- Ein weiteres Beispiel: Sie können einen Alarm überwachen `ErrorCount` und einstellen, wenn eine Zahlenspitze auftritt.

Tracker müssen mit Geofence-Sammlungen verknüpft sein, damit Gerätestandorte anhand von Geofences ausgewertet werden können. Wenn Sie über eine Geräteflotte verfügen, die kontinuierliche Standortaktualisierungen erfordert, zeigt die für `CallCount` `BatchEvaluateGeofence` oder `BatchPutDevicePosition` der Rückgang auf Null an, dass Updates nicht mehr ausgeführt werden.

Im Folgenden finden Sie eine Beispielausgabe für [GetMetricData](#) mit den Metriken für `CallCount` und `ErrorCount` zum Erstellen von Kartenressourcen.

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/Location",
          "MetricName": "CallCount",
          "Dimensions": [
            {
              "Name": "SendDeviceLocation",
              "Value": "100"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Count"
      }
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/Location",
          "MetricName": "ErrorCount",
          "Dimensions": [
            {
```

```
        "Name": "AssociateTrackerConsumer",
        "Value": "0"
    }
]
},
"Period": 1,
"Stat": "SampleCount",
"Unit": "Count"
}
}
]
```

Protokollierung und Überwachung mit AWS CloudTrail

AWS CloudTrail ist ein Service, der die Aktionen eines Benutzers, einer Rolle oder eines - AWS-Services aufzeichnet. CloudTrail zeichnet alle API-Aufrufe als Ereignisse auf. Sie können Amazon Location Service mit verwenden, CloudTrail um Ihre API-Aufrufe zu überwachen, einschließlich Aufrufen von der Amazon Location Service-Konsole und AWS SDK-Aufrufen der Amazon Location Service API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Location Service. Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse in der CloudTrail Konsole unter Ereignisverlauf anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie die an Amazon Location Service gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

Themen

- [Amazon Location Service-Informationen in CloudTrail](#)
- [Grundlegendes zu Amazon Location Service-Protokolldateieinträgen](#)

Amazon Location Service-Informationen in CloudTrail

CloudTrail wird beim Erstellen des AWS Kontos in Ihrem Konto aktiviert. Wenn eine Aktivität in Amazon Location Service auftritt, wird diese Aktivität in einem - CloudTrail Ereignis zusammen

mit anderen -AWS-Serviceereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail Ereignisverlauf](#).

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich Ereignissen für Amazon Location Service, einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen S3-Bucket. Wenn Sie ein Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien für den von Ihnen angegebenen S3 Bucket bereit. Darüber hinaus können Sie andere -AWS-Services konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren.

Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien aus mehreren Konten](#)

Alle Amazon Location Service-Aktionen werden von protokolliert CloudTrail und sind in den [API-Referenzen zu Amazon Location Service](#) dokumentiert. Aufrufe der DescribeTracker Aktionen CreateTracker, UpdateTracker und erzeugen beispielsweise Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen können Sie feststellen, ob eine Anforderung gestellt wurde:

- Mit Root- oder AWS Identity and Access Management (IAM)-Benutzeranmeldeinformationen.
- Mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer.
- Durch einen anderen AWS-Service.

Weitere Informationen finden Sie unter [CloudTrail userIdentity-Element](#).

Grundlegendes zu Amazon Location Service-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Bereitstellung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen S3-Bucket oder an Amazon CloudWatch Logs ermöglicht. Weitere Informationen finden Sie unter [Arbeiten mit CloudTrail Protokolldateien](#) im AWS CloudTrail - Benutzerhandbuch.

CloudTrail -Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Operation, Anforderungsparameter usw.

Note

CloudTrail -Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge. Um die Reihenfolge der Operationen zu bestimmen, verwenden Sie [eventTime](#).

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag, der die -CreateTrackerOperation demonstriert, die eine Tracker-Ressource erstellt.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
        "accountId": "123456789012",
        "userName": "exampleUser",
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-10-22T16:36:07Z"
      }
    }
  }
}
```

```

    }
  }
},
"eventTime": "2020-10-22T17:43:30Z",
"eventSource": "geo.amazonaws.com",
"eventName": "CreateTracker",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0/24-TEST-NET-1",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.864
Linux/4.14.193-110.317.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.8+10-LTS java/11.0.8
kotlin/1.3.72 vendor/Amazon.com_Inc. exec-env/AWS_Lambda_java11",
"requestParameters": {
  "TrackerName": "ExampleTracker",
  "Description": "Resource description"
},
"responseElements": {
  "TrackerName": "ExampleTracker",
  "Description": "Resource description"
  "TrackerArn": "arn:partition:service:region:account-id:resource-id",
  "CreateTime": "2020-10-22T17:43:30.521Z"
},
"requestID": "557ec619-0674-429d-8e2c-eba0d3f34413",
"eventID": "3192bc9c-3d3d-4976-bbef-ac590fa34f2c",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012",
}

```

Im Folgenden wird ein Protokolleintrag für die `-DescribeTrackerOperation` gezeigt, der die Details einer Tracker-Ressource zurückgibt.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:partition:service:region:account-id:resource-id",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",

```



```

        "arn": "arn:partition:service:region:account-id:resource-id",
        "accountId": "123456789012",
        "userName": "exampleUser",
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-10-22T16:36:07Z"
    }
}
},
"eventTime": "2020-10-22T17:43:33Z",
"eventSource": "geo.amazonaws.com",
"eventName": "DescribeTracker",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0/24-TEST-NET-1",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.864
Linux/4.14.193-110.317.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.8+10-LTS java/11.0.8
kotlin/1.3.72 vendor/Amazon.com_Inc. exec-env/AWS_Lambda_java11",
"requestParameters": {
    "TrackerName": "ExampleTracker"
},
"responseElements": null,
"requestID": "997d5f93-cfef-429a-bbed-daab417ceab4",
"eventID": "d9e0eebe-173c-477d-b0c9-d1d8292da103",
"readOnly": true,
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012",
}

```

Erstellen von Amazon Location Service-Ressourcen mit AWS CloudFormation

Amazon Location Service ist in integriert AWS CloudFormation, einem Service, der Ihnen hilft, Ihre AWS-Ressourcen zu modellieren und einzurichten, sodass Sie weniger Zeit mit der Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur verbringen können. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt (z. B. Amazon Location-Ressourcen) und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

Wenn Sie verwenden AWS CloudFormation, können Sie Ihre Vorlage wiederverwenden, um Ihre Amazon Location-Ressourcen konsistent und wiederholt einzurichten. Sie beschreiben Ihre

Ressourcen dann einmal und können die gleichen Ressourcen dann in mehreren AWS-Konten und -Regionen immer wieder bereitstellen.

Amazon Location und AWS CloudFormation Vorlagen

Um Ressourcen für Amazon Location und zugehörige Services bereitzustellen und zu konfigurieren, müssen Sie [AWS CloudFormation die Vorlagen](#) verstehen. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation-Stacks bereitstellen möchten. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormation Designer verwenden, der den Einstieg in die Arbeit mit AWS CloudFormation-Vorlagen erleichtert. Weitere Informationen finden Sie unter [Was ist AWS CloudFormation-Designer?](#) im AWS CloudFormation-Benutzerhandbuch.

Amazon Location unterstützt das Erstellen der folgenden Ressourcentypen in AWS CloudFormation:

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Weitere Informationen, einschließlich Beispiele für JSON- und YAML-Vorlagen für Amazon Location-Ressourcen, finden Sie in der [Ressourcentypreferenz für Amazon Location Service](#) im AWS CloudFormation -Benutzerhandbuch.

Weitere Informationen zu AWS CloudFormation

Weitere Informationen zu AWS CloudFormation finden Sie in den folgenden Ressourcen.

- [AWS CloudFormation](#)
- [AWS CloudFormation-Benutzerhandbuch](#)
- [AWS CloudFormation API Referenz](#)
- [AWS CloudFormation-Benutzerhandbuch für die Befehlszeilenschnittstelle](#)

Sicherheit bei Amazon Location Service

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) . Weitere Informationen zu den Compliance-Programmen, die für Amazon Location Service gelten, finden Sie unter [AWS Services im Umfang nach Compliance-Programm AWS](#) .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von Amazon Location anwenden können. In den folgenden Themen erfahren Sie, wie Sie Amazon Location konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon-Standortressourcen überwachen und sichern können.

Themen

- [Datenschutz bei Amazon Location Service](#)
- [Identity and Access Management für Amazon Location Service](#)
- [Reaktion auf Vorfälle in Amazon Location Service](#)
- [Konformitätsprüfung für Amazon Location Service](#)
- [Resilienz bei Amazon Location Service](#)
- [Infrastruktursicherheit in Amazon Location Service](#)
- [Konfiguration und Schwachstellenanalyse in Amazon Location](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Bewährte Sicherheitsmethoden für Amazon Location Service](#)

- [Bewährte Methoden für Amazon Location Service](#)

Datenschutz bei Amazon Location Service

Das AWS [Modell](#) der mit gilt für den Datenschutz in Amazon Location Service. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der die gesamte Infrastruktur läuft AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Location oder anderen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenschutz

Mit Amazon Location Service behalten Sie die Kontrolle über die Daten Ihres Unternehmens. Amazon Location anonymisiert alle Anfragen, die an Datenanbieter gesendet werden, indem Kundenmetadaten und Kontoinformationen entfernt werden.

Amazon Location verwendet keine Datenanbieter für Tracking und Geofencing. Das bedeutet, dass Ihre sensiblen Daten in Ihrem AWS Konto verbleiben. Dies trägt dazu bei, vertrauliche Standortinformationen wie Standort, Anlage und Personalstandort vor Dritten zu schützen, die Privatsphäre der Benutzer zu schützen und das Sicherheitsrisiko Ihrer Anwendung zu verringern.

Weitere Informationen finden Sie in den [häufig gestellten Fragen zum AWS Datenschutz](#).

Aufbewahrung von Daten bei Amazon Location

Die folgenden Merkmale beziehen sich darauf, wie Amazon Location Daten für den Service sammelt und speichert:

- Amazon Location Service Trackers — Wenn Sie die Trackers-APIs verwenden, um den Standort von Entitäten zu verfolgen, können deren Koordinaten gespeichert werden. Gerätestandorte werden 30 Tage lang gespeichert, bevor sie vom Service gelöscht werden.
- Amazon Location Service Geofences — Wenn Sie die Geofences-APIs verwenden, um Interessengebiete zu definieren, speichert der Service die von Ihnen angegebenen Geometrien. Sie müssen explizit gelöscht werden.

Note

Wenn Sie Ihr AWS Konto löschen, werden alle darin enthaltenen Ressourcen gelöscht. Weitere Informationen finden Sie in den [häufig gestellten Fragen zum AWS Datenschutz](#).

Datenverschlüsselung im Ruhezustand für Amazon Location Service

Amazon Location Service bietet standardmäßig Verschlüsselung, um vertrauliche Kundendaten im Speicher mithilfe AWS eigener Verschlüsselungsschlüssel zu schützen.

- **AWS eigene Schlüssel** — Amazon Location verwendet diese Schlüssel standardmäßig, um persönlich identifizierbare Daten automatisch zu verschlüsseln. Sie können AWS eigene Schlüssel nicht einsehen, verwalten oder verwenden oder deren Verwendung überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme zum Schutz der Schlüssel ändern, die zur Verschlüsselung Ihrer Daten verwendet werden. Weitere Informationen finden Sie unter [AWS - eigene Schlüssel](#) im AWS Key Management Service -Entwicklerhandbuch.

Die standardmäßige Verschlüsselung von Daten im Ruhezustand trägt dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz vertraulicher Daten verbunden sind. Gleichzeitig können Sie damit sichere Anwendungen erstellen, die strenge Verschlüsselungsvorschriften und gesetzliche Auflagen erfüllen.

Sie können diese Verschlüsselungsebene zwar nicht deaktivieren oder einen anderen Verschlüsselungstyp auswählen, aber Sie können eine zweite Verschlüsselungsebene über den vorhandenen AWS eigenen Verschlüsselungsschlüsseln hinzufügen, indem Sie bei der Erstellung Ihrer Tracker- und Geofence-Collection-Ressourcen einen vom Kunden verwalteten Schlüssel auswählen:

- **Vom Kunden verwaltete Schlüssel** — Amazon Location unterstützt die Verwendung eines symmetrischen, vom Kunden verwalteten Schlüssels, den Sie erstellen, besitzen und verwalten, um eine zweite Verschlüsselungsebene über der vorhandenen AWS eigenen Verschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:
 - Festlegung und Pflege wichtiger Richtlinien
 - Festlegung und Aufrechterhaltung von IAM-Richtlinien und -Zuschüssen
 - Aktivieren und Deaktivieren wichtiger Richtlinien
 - Kryptographisches Material mit rotierendem Schlüssel
 - Hinzufügen von Tags
 - Erstellen von Schlüsselaliasen
 - Schlüssel für das Löschen von Schlüsseln planen

Weitere Informationen finden Sie unter vom [Kunden verwalteter Schlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Die folgende Tabelle fasst zusammen, wie Amazon Location personenbezogene Daten verschlüsselt.

Datentyp	AWS Verschlüsselung mit eigenem Schlüssel	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
Position Eine Punktgeometrie, die die Details zur Geräteposition enthält.	Aktiviert	Aktiviert
PositionProperties Ein Satz von Schlüssel-Wert-Paaren, die mit der Positionsaktualisierung verknüpft sind .	Aktiviert	Aktiviert
GeofenceGeometry Eine Polygon-Geofence-Geometrie , die den abgegrenzten Bereich darstellt.	Aktiviert	Aktiviert
DeviceId Die Geräteerkennung, die beim Hochladen einer Aktualisierung der Geräteposition in eine Tracker-Ressource angegeben wurde.	Aktiviert	Nicht unterstützt
GeofenceId Eine Kennung, die beim Speichern einer Geofence-Geometrie oder eines Stapels von Geofences in einer bestimmten Geofence-Sammlung angegeben wird.	Aktiviert	Nicht unterstützt

Note

Amazon Location aktiviert automatisch die Verschlüsselung im Ruhezustand mithilfe AWS eigener Schlüssel, um personenbezogene Daten kostenlos zu schützen.

Für die Verwendung eines vom Kunden verwalteten Schlüssels fallen jedoch AWS KMS Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter [AWS Key Management Service Preisgestaltung](#).

Weitere Informationen zu AWS KMS finden Sie unter [Was ist AWS Key Management Service?](#)

So verwendet Amazon Location Service Zuschüsse in AWS KMS

Für den Amazon-Standort ist eine [Genehmigung](#) erforderlich, um Ihren vom Kunden verwalteten Schlüssel verwenden zu können.

Wenn Sie eine [Tracker-Ressource](#) oder eine [Geofence-Sammlung](#) erstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, erstellt Amazon Location in Ihrem Namen einen Zuschuss, indem es eine [CreateGrant](#)Anfrage an sendet. AWS KMS Grants in AWS KMS werden verwendet, um Amazon Location Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Amazon Location benötigt den Zuschuss, um Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Operationen zu verwenden:

- Senden Sie [DescribeKey](#)Anfragen an, um AWS KMS zu überprüfen, ob die symmetrische, vom Kunden verwaltete KMS-Schlüssel-ID, die Sie bei der Erstellung einer Tracker- oder Geofence-Sammlung eingegeben haben, gültig ist.
- Senden Sie [GenerateDataKeyWithoutPlaintext](#)Anfragen AWS KMS zur Generierung von Datenschlüsseln, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt sind.
- Senden Sie [Entschlüsselungsanfragen](#) an AWS KMS, um die verschlüsselten Datenschlüssel zu entschlüsseln, sodass sie zur Verschlüsselung Ihrer Daten verwendet werden können.

Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, kann Amazon Location auf keine der mit dem vom Kunden verwalteten Schlüssel verschlüsselten Daten zugreifen, was sich auf Vorgänge auswirkt, die von diesen Daten abhängig sind. Wenn Sie beispielsweise versuchen,

[Gerätepositionen](#) von einem verschlüsselten Tracker abzurufen, auf den Amazon Location nicht zugreifen kann, würde der Vorgang einen `AccessDeniedException` Fehler zurückgeben.

Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS Management Console AWS KMS APIs oder erstellen.

Einen symmetrischen kundenverwalteten Schlüssel erstellen

Folgen Sie den Schritten zum [Erstellen eines symmetrischen kundenverwalteten Schlüssels](#) im Entwicklerhandbuch zum AWS Key Management Service .

Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter [Verwalten des Zugriffs auf kundenverwaltete Schlüssel](#) im Entwicklerhandbuch zum AWS Key Management Service

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren Amazon-Standortressourcen zu verwenden, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zugelassen sein:

- [kms:CreateGrant](#): Fügt einem kundenverwalteten Schlüssel eine Erteilung hinzu. Gewährt Kontrollzugriff auf einen bestimmten KMS-Schlüssel, der den Zugriff auf [Grant-Operationen](#) ermöglicht, die Amazon Location benötigt. Weitere Informationen zur [Verwendung von Grants](#) finden Sie im AWS Key Management Service Developer Guide.

Dadurch kann Amazon Location Folgendes tun:

- `GenerateDataKeyWithoutPlainText` aufrufen, um einen verschlüsselten Datenschlüssel zu generieren und zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.
- `Decrypt` aufrufen, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
- Richten Sie einen Principal ein, der in den Ruhezustand geht, damit der Service dies tun `RetireGrant` kann.

- [kms:DescribeKey](#)— Stellt die vom Kunden verwalteten Schlüsselinformationen bereit, damit Amazon Location den Schlüssel validieren kann.

Im Folgenden finden Sie Beispiele für Richtlinienerklärungen, die Sie für Amazon Location hinzufügen können:

```
"Statement" : [
  {
    "Sid" : "Allow access to principals authorized to use Amazon Location",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [
      "kms:DescribeKey",
      "kms:CreateGrant"
    ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "geo.region.amazonaws.com",
        "kms:CallerAccount" : "111122223333"
      }
    }
  },
  {
    "Sid": "Allow access for key administrators",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action" : [
      "kms:*"
    ],
    "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
  },
  {
    "Sid" : "Allow read-only access to key metadata to the account",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : [
```

```
    "kms:Describe*",
    "kms:Get*",
    "kms:List*",
    "kms:RevokeGrant"
  ],
  "Resource" : "*"
}
```

Weitere Informationen zum [Festlegen von Berechtigungen in einer Richtlinie](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Weitere Informationen zur [Fehlerbehebung beim Schlüsselzugriff](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Angabe eines vom Kunden verwalteten Schlüssels für Amazon Location

Sie können einen vom Kunden verwalteten Schlüssel als zweite Verschlüsselungsebene für die folgenden Ressourcen festlegen:

- [Tracker-Ressource](#)
- [Geofence-Sammlung](#)

Wenn Sie eine Ressource erstellen, können Sie den Datenschlüssel angeben, indem Sie eine KMS-ID eingeben, die Amazon Location verwendet, um die von der Ressource gespeicherten identifizierbaren persönlichen Daten zu verschlüsseln.

- KMS-ID — Eine [Schlüssel-ID](#) für einen vom AWS KMS Kunden verwalteten Schlüssel. Geben Sie eine Schlüssel-ID, einen Schlüssel-ARN, einen Alias-Namen oder einen Alias-ARN ein.

Amazon Location Service Service-Verschlüsselungskontext

Ein [Verschlüsselungskontext](#) ist ein optionaler Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten.

AWS KMS verwendet den Verschlüsselungskontext als [zusätzliche authentifizierte Daten](#), um die [authentifizierte](#) Verschlüsselung zu unterstützen. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zum Verschlüsseln von Daten einbeziehen, wird der Verschlüsselungskontext AWS KMS an die verschlüsselten Daten gebunden. Zur Entschlüsselung von Daten müssen Sie denselben Verschlüsselungskontext in der Anfrage übergeben.

Amazon Location Service Service-Verschlüsselungskontext

Amazon Location verwendet bei allen AWS KMS kryptografischen Vorgängen denselben Verschlüsselungskontext, wobei der Schlüssel `aws:geo:arn` und der Wert die Ressource [Amazon Resource Name](#) (ARN) ist.

Example

```
"encryptionContext": {
  "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

Verwenden des Verschlüsselungskontexts für die Überwachung

Wenn Sie einen symmetrischen, vom Kunden verwalteten Schlüssel verwenden, um Ihren Tracker oder Ihre Geofence-Erfassung zu verschlüsseln, können Sie den Verschlüsselungskontext auch in Prüfaufzeichnungen und Protokollen verwenden, um festzustellen, wie der vom Kunden verwaltete Schlüssel verwendet wird. Der Verschlüsselungskontext erscheint auch in [Protokollen, die von Amazon CloudWatch Logs generiert wurden AWS CloudTrail](#).

Verwendung des Verschlüsselungskontextes zur Steuerung des Zugriffs auf den vom Kunden verwalteten Schlüssel

Sie können den Verschlüsselungskontext in Schlüsselrichtlinien und IAM-Richtlinien als `conditions` verwenden, um den Zugriff auf Ihren symmetrischen, kundenverwalteten Schlüssel zu kontrollieren. Sie können Verschlüsselungskontext-Einschränkungen auch in einer Genehmigung verwenden.

Amazon Location verwendet bei Zuschüssen eine Einschränkung des Verschlüsselungskontextes, um den Zugriff auf den vom Kunden verwalteten Schlüssel in Ihrem Konto oder Ihrer Region zu kontrollieren. Eine Genehmigungseinschränkung erfordert, dass durch die Genehmigung ermöglichte Vorgänge den angegebenen Verschlüsselungskontext verwenden.

Example

Im Folgenden finden Sie Beispiele für Schlüsselrichtlinienanweisungen zur Gewährung des Zugriffs auf einen vom Kunden verwalteten Schlüssel für einen bestimmten Verschlüsselungskontext. Die Bedingung in dieser Richtlinienanweisung setzt voraus, dass die Genehmigungen eine Einschränkung des Verschlüsselungskontextes haben, die den Verschlüsselungskontext spezifiziert.

```
{
```

```

    "Sid": "Enable DescribeKey",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  },
  {
    "Sid": "Enable CreateGrant",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
    },
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-
west-2:111122223333:tracker/SAMPLE-Tracker"
      }
    }
  }
}

```

Überwachung Ihrer Verschlüsselungsschlüssel für Amazon Location Service

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel mit Ihren Amazon Location Service Service-Ressourcen verwenden, können Sie [AWS CloudTrail](#) oder [Amazon CloudWatch Logs](#) verwenden, um Anfragen zu verfolgen, an die Amazon Location sendet AWS KMS.

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für `CreateGrant`, `GenerateDataKeyWithoutPlainText`, und `DescribeKey` zur Überwachung von KMS-Vorgängen `Decrypt`, die von Amazon Location aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

CreateGrant

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel verwenden, um Ihre Tracker- oder Geofence-Sammelressourcen zu verschlüsseln, sendet Amazon Location in Ihrem Namen eine `CreateGrant` Anfrage für den Zugriff auf den KMS-Schlüssel in Ihrem Konto. AWS Der Zuschuss, den Amazon Location gewährt, ist spezifisch für die Ressource, die dem vom AWS KMS Kunden verwalteten Schlüssel zugeordnet ist. Darüber hinaus verwendet Amazon Location

den `RetireGrant` Vorgang, um einen Zuschuss zu entfernen, wenn Sie eine Ressource löschen.

Das folgende Beispiereignis zeichnet den Vorgang `CreateGrant` auf:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-04-22T17:02:00Z"
      }
    },
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:07:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "retiringPrincipal": "geo.region.amazonaws.com",
    "operations": [
      "GenerateDataKeyWithoutPlaintext",
      "Decrypt",
      "DescribeKey"
    ],
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
  }
}
```

```

    "granteePrincipal": "geo.region.amazonaws.com"
  },
  "responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}

```

GenerateDataKeyWithoutPlainText

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für Ihre Tracker- oder Geofence-Sammelressource aktivieren, erstellt Amazon Location einen eindeutigen Tabellenschlüssel. Es sendet eine `GenerateDataKeyWithoutPlainText` Anfrage an AWS KMS, die den vom AWS KMS Kunden verwalteten Schlüssel für die Ressource angibt.

Das folgende Beispielergebnis zeichnet den Vorgang `GenerateDataKeyWithoutPlainText` auf:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:07:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-west-2",

```

```

"sourceIPAddress": "172.12.34.56",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
  "encryptionContext": {
    "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/
SAMPLE-GeofenceCollection"
  },
  "keySpec": "AES_256",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333",
"sharedEventID": "57f5dbec-16da-413e-979f-2c4c6663475e"
}

```

Decrypt

Wenn Sie auf eine verschlüsselte Tracker- oder Geofence-Erfassung zugreifen, ruft Amazon Location den Decrypt Vorgang auf, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf die verschlüsselten Daten zu verwenden.

Das folgende Beispiereignis zeichnet den Vorgang Decrypt auf:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "geo.amazonaws.com"
  }
}

```



```

},
"eventTime": "2021-04-22T17:10:51Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "172.12.34.56",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
  "encryptionContext": {
    "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/
SAMPLE-GeofenceCollection"
  },
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333",
"sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}

```

DescribeKey

Amazon Location verwendet diesen DescribeKey Vorgang, um zu überprüfen, ob der vom AWS KMS Kunden verwaltete Schlüssel, der mit Ihrer Tracker- oder Geofence-Sammlung verknüpft ist, in dem Konto und in der Region vorhanden ist.

Das folgende Beispiereignis zeichnet den Vorgang DescribeKey auf:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-04-22T17:02:00Z"
      }
    },
    "invokedBy": "geo.amazonaws.com"
  },
  "eventTime": "2021-04-22T17:07:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ]
}
```

```
    }  
  ],  
  "eventType": "AwsApiCall",  
  "managementEvent": true,  
  "eventCategory": "Management",  
  "recipientAccountId": "111122223333"  
}
```

Weitere Informationen

Die folgenden Ressourcen enthalten weitere Informationen zur Datenverschlüsselung im Ruhezustand:

- Weitere Informationen zu grundlegenden [AWS Key Management Service -Konzepten](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.
- Weitere Informationen zu [bewährten Sicherheitsmethoden für AWS Key Management Service](#) finden Sie im AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung von Daten bei der Übertragung für Amazon Location Service

Amazon Location schützt Daten während der Übertragung zum und vom Service, indem alle Netzwerkdaten automatisch mit dem Verschlüsselungsprotokoll Transport Layer Security (TLS) 1.2 verschlüsselt werden. Direkte HTTPS-Anfragen, die an die Amazon Location Service APIs gesendet werden, werden mithilfe des [AWS Signature Version 4-Algorithmus](#) signiert, um eine sichere Verbindung herzustellen.

Identity and Access Management für Amazon Location Service

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon-Standortressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert Amazon Location Service mit IAM](#)
- [So funktioniert Amazon Location Service mit nicht authentifizierten Benutzern](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)
- [Fehlerbehebung bei Identität und Zugriff auf Amazon Location Service](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon Location ausführen.

Servicebenutzer — Wenn Sie den Amazon Location Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Funktionen von Amazon Location verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in Amazon Location nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung bei Identität und Zugriff auf Amazon Location Service](#).

Service-Administrator — Wenn Sie in Ihrem Unternehmen für die Ressourcen von Amazon Location verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Location. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon Location Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit Amazon Location verwenden kann, finden Sie unter [So funktioniert Amazon Location Service mit IAM](#).

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon Location schreiben können. Beispiele für identitätsbasierte Richtlinien von Amazon Location, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein neues AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen

und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch](#).

- **Serviceübergreifender Zugriff** — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon EC2 ausgeführt werden** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie

mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze

für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.

- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

So funktioniert Amazon Location Service mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Location zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für Amazon Location verfügbar sind.

IAM-Funktionen, die Sie mit Amazon Location Service verwenden können

IAM-Feature	Unterstützung Amazon Amazon-Standorte
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Nein
Servicerollen	Nein
Serviceverknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie Amazon Location und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für Amazon Location

Unterstützt Richtlinien auf Identitätsbasis.	Ja
--	----

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen

ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon Location

Beispiele für identitätsbasierte Richtlinien von Amazon Location finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)

Ressourcenbasierte Richtlinien innerhalb von Amazon Location

Unterstützt ressourcenbasierte Richtlinien	Nein
--	------

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalidentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto

gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontenübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Politische Maßnahmen für Amazon Location

Unterstützt Richtlinienaktionen

Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Amazon-Standortaktionen finden Sie unter [Von Amazon Location Service definierte Aktionen in der Service Authorization Reference](#).

Richtlinienaktionen in Amazon Location verwenden das folgende Präfix vor der Aktion:

```
geo
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "geo:action1",  
  "geo:action2"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Get` beginnen, einschließlich der folgenden Aktion:

```
"Action": "geo:Get*"
```

Beispiele für identitätsbasierte Richtlinien von Amazon Location finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)

Richtlinienressourcen für Amazon Location

Unterstützt Richtlinienressourcen	Ja
-----------------------------------	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der Ressourcentypen von Amazon Location und deren ARNs finden Sie unter [Von Amazon Location Service definierte Ressourcen in der Service](#) Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien von Amazon Location finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)

Schlüssel zu den Richtlinienbedingungen für Amazon Location

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja
---	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Amazon Location-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Location Service in der Service](#) Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Amazon Location unterstützt Bedingungsschlüssel, mit denen Sie in Ihren Richtlinienerklärungen den Zugriff auf bestimmte Geofences oder Geräte zulassen oder verweigern können. Die folgenden Bedingungsschlüssel sind verfügbar:

- `geo:GeofenceIds` zur Verwendung mit Geofence-Aktionen. Der Typ ist `ArrayOfString`

- `geo:DeviceIds` zur Verwendung mit Tracker-Aktionen. Der Typ ist `ArrayOfString`.

Die folgenden Aktionen können `geo:GeofenceIds` in Ihrer IAM-Richtlinie verwendet werden:

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

Die folgenden Aktionen können zusammen mit Ihrer `geo:DeviceIds` IAM-Richtlinie verwendet werden:

- `BatchDeleteDevicePositionHistory`
- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

Note

Sie können diese Bedingungsschlüssel nicht mit den `ListDevicePosition` Aktionen `BatchEvaluateGeofences` `ListGeofences`, oder verwenden.

Beispiele für identitätsbasierte Richtlinien von Amazon Location finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)

ACLs am Amazon-Standort

Unterstützt ACLs

Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon-Standort

Unterstützt ABAC (Tags in Richtlinien)	Ja
--	----

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen zum Taggen von Amazon-Standortressourcen finden Sie unter [Markieren Ihrer Amazon Location Service-Ressourcen](#).

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter [Steuern Sie den Ressourcenzugriff anhand von Tags](#).

Temporäre Anmeldeinformationen mit Amazon Location verwenden

Unterstützt temporäre Anmeldeinformationen	Ja
--	----

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services](#) , [finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Serviceübergreifende Hauptberechtigungen für Amazon Location


Unterstützt Forward Access Sessions (FAS)	Nein
---	------

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon Location

Unterstützt Servicerollen	Nein
---------------------------	------

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

 Warning

Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon Location beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon Location Sie dazu anleitet.

Servicebezogene Rollen für Amazon Location

Unterstützt serviceverknüpfte Rollen Nein

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

So funktioniert Amazon Location Service mit nicht authentifizierten Benutzern

Viele Szenarien für die Nutzung von Amazon Location Service, einschließlich der Anzeige von Karten im Internet oder in einer mobilen Anwendung, erfordern, dass Benutzern, die sich nicht mit IAM angemeldet haben, Zugriff gewährt wird. Für diese nicht authentifizierten Szenarien haben Sie zwei Optionen.

- API-Schlüssel verwenden — Um nicht authentifizierten Benutzern Zugriff zu gewähren, können Sie API-Schlüssel erstellen, die nur Lesezugriff auf Ihre Amazon Location Service Service-Ressourcen

gewähren. Dies ist nützlich, wenn Sie nicht jeden Benutzer authentifizieren möchten. Zum Beispiel eine Webanwendung. Weitere Hinweise zu API-Schlüsseln finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).

- Amazon Cognito verwenden — Eine Alternative zu API-Schlüsseln besteht darin, Amazon Cognito zu verwenden, um anonymen Zugriff zu gewähren. Amazon Cognito ermöglicht es Ihnen, eine umfassendere Autorisierung mit IAM-Richtlinie zu erstellen, um zu definieren, was die nicht authentifizierten Benutzer tun können. Weitere Informationen zur Verwendung von Amazon Cognito finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

Eine Übersicht über die Bereitstellung des Zugriffs für nicht authentifizierte Benutzer finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#)

Beispiele für identitätsbasierte Richtlinien für Amazon Location Service

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon-Standortressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon Location definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter [Aktionen, Ressourcen und Zustandsschlüssel für Amazon Location Service in der Service](#) Authorization Reference.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Amazon Location-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Verwendung von Amazon Location Service Service-Ressourcen in der Richtlinie](#)
- [Berechtigungen für die Aktualisierung der Gerätepositionen](#)

- [Schreibgeschützte Richtlinie für Tracker-Ressourcen](#)
- [Richtlinie für die Erstellung von Geofences](#)
- [Schreibgeschützte Richtlinie für Geofences](#)
- [Berechtigungen für das Rendern einer Kartenressource](#)
- [Berechtigungen zum Zulassen von Suchvorgängen](#)
- [Schreibgeschützte Richtlinie für Routenrechner](#)
- [Steuert den Ressourcenzugriff auf der Grundlage von Bedingungsschlüsseln](#)
- [Steuern Sie den Ressourcenzugriff anhand von Tags](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon-Standortressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren, verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben,

um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Amazon Location-Konsole

Um auf die Amazon Location Service Service-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon-Standortressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Amazon Location-Konsole verwenden können, fügen Sie den Entitäten die folgende Richtlinie bei. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Die folgende Richtlinie gewährt Zugriff auf die Amazon Location Service Service-Konsole, sodass Sie Details zu Amazon Location-Ressourcen in Ihrem AWS Konto erstellen, löschen, auflisten und anzeigen können.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Alternativ können Sie nur Leseberechtigungen gewähren, um den Nur-Lese-Zugriff zu ermöglichen. Bei schreibgeschützten Berechtigungen wird eine Fehlermeldung angezeigt, wenn der Benutzer versucht, Schreibaktionen wie das Erstellen oder Löschen von Ressourcen durchzuführen. Ein Beispiel finden Sie unter [the section called “Nur-Lese-Richtlinie für Tracker”](#)

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder 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": "*"
  }
]
}

```

Verwendung von Amazon Location Service Service-Ressourcen in der Richtlinie

Amazon Location Service verwendet die folgenden Präfixe für Ressourcen:

Ressourcenpräfix für Amazon-Standorte

Ressource	Ressourcenpräfix
Ressourcen zuordnen	map
Ressourcen platzieren	place-index
Ressourcen weiterleiten	route-calculator
Ressourcen nachverfolgen	tracker
Ressourcen für die Geofence-Sammlung	geofence-collection

Verwenden Sie die folgende ARN-Syntax:

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

Weitere Informationen zum Format von ARNs finden Sie unter [Amazon Resource Names \(ARNs\) und AWS Service Namespaces](#).

Beispiele

- Verwenden Sie den folgenden ARN, um den Zugriff auf eine angegebene Kartenressource zu ermöglichen.

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
```

- Um den Zugriff auf alle map Ressourcen anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/*"
```

- Einige Amazon-Standortaktionen, z. B. zum Erstellen von Ressourcen, können nicht für eine bestimmte Ressource ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Eine Liste der Ressourcentypen von Amazon Location und deren ARNs finden Sie unter [Von Amazon Location Service definierte Ressourcen in der Service](#) Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Berechtigungen für die Aktualisierung der Gerätepositionen

Um die Gerätepositionen für mehrere Tracker zu aktualisieren, sollten Sie einem Benutzer Zugriff auf eine oder mehrere Ihrer Tracker-Ressourcen gewähren. Sie sollten dem Benutzer auch ermöglichen, eine Reihe von Gerätepositionen zu aktualisieren.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf die Tracker1- und Tracker2-Ressourcen, sondern gewährt auch die Erlaubnis, die geo:BatchUpdateDevicePosition Aktion für die Tracker1- und Tracker2-Ressourcen zu verwenden.

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "UpdateDevicePositions",
    "Effect": "Allow",
    "Action": [
      "geo:BatchUpdateDevicePosition"
    ],
    "Resource": [
      "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",
      "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
    ]
  }
]
}

```

Wenn Sie den Benutzer so einschränken möchten, dass er nur Gerätepositionen für ein bestimmtes Gerät aktualisieren kann, können Sie einen Bedingungsschlüssel für diese Geräte-ID hinzufügen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
      ],
      "Condition": {
        "ForAllValues:StringLike": {
          "geo:DeviceIds": [
            "deviceId"
          ]
        }
      }
    }
  ]
}

```

Schreibgeschützte Richtlinie für Tracker-Ressourcen

Um eine Richtlinie mit Lesezugriff für alle Tracker-Ressourcen in Ihrem AWS Konto zu erstellen, müssen Sie Zugriff auf alle Tracker-Ressourcen gewähren. Sie sollten einem Benutzer auch Zugriff auf Aktionen gewähren, mit denen er die Geräteposition für mehrere Geräte abrufen, die Geräteposition von einem einzelnen Gerät abrufen und den Positionsverlauf abrufen kann.

In diesem Beispiel gewährt die folgende Richtlinie die Erlaubnis für die folgenden Aktionen:

- `geo:BatchGetDevicePosition`um die Position mehrerer Geräte abzurufen.
- `geo:GetDevicePosition`um die Position eines einzelnen Geräts abzurufen.
- `geo:GetDevicePositionHistory`um den Positionshistorie eines Geräts abzurufen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchGetDevicePosition",
        "geo:GetDevicePosition",
        "geo:GetDevicePositionHistory"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:tracker/*"
    }
  ]
}
```

Richtlinie für die Erstellung von Geofences

Um eine Richtlinie zu erstellen, die es einem Benutzer ermöglicht, Geofences zu erstellen, müssen Sie Zugriff auf bestimmte Aktionen gewähren, mit denen Benutzer einen oder mehrere Geofences in einer Geofence-Sammlung erstellen können.

Die folgende Richtlinie gewährt die Genehmigung für die folgenden Aktionen bei der Erfassung:

- `geo:BatchPutGeofence`um mehrere Geofences zu erstellen.

- `geo:PutGeofence`um einen einzigen Geofence zu erstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:BatchPutGeofence",
        "geo:PutGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
}
```

Schreibgeschützte Richtlinie für Geofences

Um eine schreibgeschützte Richtlinie für Geofences zu erstellen, die in einer Geofence-Sammlung in Ihrem AWS Konto gespeichert sind, müssen Sie Zugriff auf Aktionen gewähren, die aus der Geofence-Sammlung lesen, in der die Geofences gespeichert sind.

Die unten stehende Richtlinie gewährt die Genehmigung für die folgenden Aktionen bei der Erfassung:

- `geo:ListGeofences`um Geofences in der angegebenen Geofence-Sammlung aufzulisten.
- `geo:GetGeofence`um einen Geofence aus der Geofence-Sammlung abzurufen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:ListGeofences",
        "geo:GetGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
}
```

```
    }  
  ]  
}
```

Berechtigungen für das Rendern einer Kartenressource

Um ausreichende Berechtigungen zum Rendern von Karten zu gewähren, müssen Sie Zugriff auf Kartenkacheln, Sprites, Glyphen und den Style-Deskriptor gewähren:

- `geo:GetMapTiles` ruft Kartenkacheln ab, die zum selektiven Rendern von Features auf einer Karte verwendet werden.
- `geo:GetMapSprites` ruft das PNG-Spritesheet und das entsprechende JSON-Dokument ab, das die darin enthaltenen Offsets beschreibt.
- `geo:GetMapGlyphs` ruft die Glyphen ab, die für die Anzeige von Text verwendet werden.
- `geo:GetMapStyleDescriptor` ruft den Stildeskriptor der Map ab, der die Renderregeln enthält.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "GetTiles",  
      "Effect": "Allow",  
      "Action": [  
        "geo:GetMapTile",  
        "geo:GetMapSprites",  
        "geo:GetMapGlyphs",  
        "geo:GetMapStyleDescriptor"  
      ],  
      "Resource": "arn:aws:geo:us-west-2:account-id:map/Map"  
    }  
  ]  
}
```

Berechtigungen zum Zulassen von Suchvorgängen

Um eine Richtlinie zu erstellen, die Suchvorgänge zulässt, müssen Sie zunächst Zugriff auf die Ortsindex-Ressource in Ihrem AWS Konto gewähren. Sie sollten auch Zugriff auf Aktionen gewähren, mit denen der Benutzer mithilfe von Text durch Geokodierung und mithilfe einer Position durch umgekehrte Geokodierung suchen kann.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf *PlaceIndex*, sondern gewährt auch Berechtigungen für die folgenden Aktionen:

- `geo:SearchPlaceIndexForPosition` ermöglicht es Ihnen, nach Orten oder Sehenswürdigkeiten in der Nähe einer bestimmten Position zu suchen.
- `geo:SearchPlaceIndexForText` ermöglicht es Ihnen, mithilfe von Freiformtext nach einer Adresse, einem Namen, einer Stadt oder einer Region zu suchen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Search",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndexForPosition",
        "geo:SearchPlaceIndexForText"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:place-index/PlaceIndex"
    }
  ]
}
```

Schreibgeschützte Richtlinie für Routenrechner

Sie können eine schreibgeschützte Richtlinie erstellen, um einem Benutzer Zugriff auf eine Routenberechnungsressource zu gewähren, um eine Route zu berechnen.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf *ExampleCalculator*, sondern gewährt auch die Erlaubnis für den folgenden Vorgang:

- `geo:CalculateRoute` berechnet eine Route anhand einer Startposition, einer Zielposition und einer Liste von Wegpunktpositionen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
```

```

    "Effect": "Allow",
    "Action": [
      "geo:CalculateRoute"
    ],
    "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
  }
]
}

```

Steuert den Ressourcenzugriff auf der Grundlage von Bedingungschlüsseln

Wenn Sie eine IAM-Richtlinie erstellen, um Zugriff auf die Nutzung von Geofences oder Gerätepositionen zu gewähren, können Sie [Bedingungsoperatoren](#) verwenden, um genauer zu steuern, auf welche Geofences oder Geräte ein Benutzer zugreifen kann. Sie können dies tun, indem Sie die Geofence-ID oder die Geräte-ID in das Element Ihrer Richtlinie aufnehmen. `Condition`

Die folgende Beispielrichtlinie zeigt, wie Sie eine Richtlinie erstellen können, die es einem Benutzer ermöglicht, Gerätepositionen für ein bestimmtes Gerät zu aktualisieren.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker"
      ],
      "Condition": {
        "ForAllValues:StringLike": {
          "geo:DeviceIds": [
            "deviceId"
          ]
        }
      }
    }
  ]
}

```


Steuern Sie den Ressourcenzugriff anhand von Tags

Wenn Sie eine IAM-Richtlinie erstellen, um Zugriff auf Ihre Amazon-Standortressourcen zu gewähren, können Sie die [attributbasierte Zugriffskontrolle verwenden, um besser kontrollieren](#) zu können, welche Ressourcen ein Benutzer ändern, verwenden oder löschen kann. [Sie können dies tun, indem Sie Tag-Informationen in das Condition Element Ihrer Richtlinie aufnehmen, um den Zugriff anhand Ihrer Ressourcen-Tags zu kontrollieren.](#)

Die folgende Beispielrichtlinie zeigt, wie Sie eine Richtlinie erstellen können, die es einem Benutzer ermöglicht, Geofences zu erstellen. *Dadurch erhalten Sie die Erlaubnis für die folgenden Aktionen, einen oder mehrere Geofences in einer Geofence-Sammlung namens Collection zu erstellen:*

- `geo:BatchPutGeofence`um mehrere Geofences zu erstellen.
- `geo:PutGeofence`um einen einzigen Geofence zu erstellen.

Diese Richtlinie verwendet das Condition Element jedoch nur dann, um die Berechtigung zu erteilen, wenn das *Sammlungs-Tag*`Owner`, den Wert des Benutzernamens dieses Benutzers hat.

- Wenn ein benannter Benutzer beispielsweise `richard-roe` versucht, eine *Amazon-Standortsammlung aufzurufen*, muss die *Sammlung* mit `Owner=richard-roe` oder gekennzeichnet werden `owner=richard-roe`. Andernfalls wird dem Benutzer der Zugriff verweigert.

Note

Der Tag-Schlüssel `Owner` der Bedingung stimmt sowohl mit `Owner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofencesIfOwner",
      "Effect": "Allow",
```

```
"Action": [
  "geo:BatchPutGeofence",
  "geo:PutGeofence"
],
"Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection",
"Condition": {
  "StringEquals": {"geo:ResourceTag/Owner": "${aws:username}"}
}
]
```

Ein [Tutorial zur Definition von Zugriffsberechtigungen für AWS-Ressourcen auf der Grundlage von Tags](#) finden Sie im AWS Identity and Access Management Benutzerhandbuch.

Fehlerbehebung bei Identität und Zugriff auf Amazon Location Service

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Location und IAM auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion in Amazon Location durchzuführen](#)
- [Ich bin nicht berechtigt, iam durchzuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon-Standortressourcen ermöglichen](#)

Ich bin nicht berechtigt, eine Aktion in Amazon Location durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven `my-example-widget`-Ressource anzuzeigen, jedoch nicht über `geo:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
geo:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `geo:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon Location übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Location auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon-Standortressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon Location diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Location Service mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie im IAM-Benutzerhandbuch unter [Kontenübergreifender Ressourcenzugriff in IAM](#).

Reaktion auf Vorfälle in Amazon Location Service

Sicherheit hat bei uns höchste Priorität AWS. AWS verwaltet im Rahmen des [Modells der gemeinsamen Verantwortung](#) in der AWS Cloud ein Rechenzentrum und eine Netzwerkarchitektur, die den Anforderungen der sicherheitssensibelsten Unternehmen gerecht wird. Als AWS Kunde tragen Sie gemeinsam die Verantwortung für die Aufrechterhaltung der Sicherheit in der Cloud. Das bedeutet, dass Sie die Sicherheit, die Sie implementieren möchten, anhand der AWS Tools und Funktionen, auf die Sie Zugriff haben, kontrollieren.

Indem Sie eine Sicherheitsbasis einrichten, die den Zielen Ihrer in der Cloud ausgeführten Anwendungen entspricht, können Sie Abweichungen erkennen, auf die Sie reagieren können. Da die Reaktion auf Sicherheitsvorfälle ein komplexes Thema sein kann, empfehlen wir Ihnen, die folgenden Ressourcen zu lesen, damit Sie besser verstehen, welche Auswirkungen die Reaktion auf Vorfälle (IR) und Ihre Entscheidungen auf Ihre Unternehmensziele haben: [AWS Security Incident Response Guide](#), [AWS Security Best Practices](#) Whitepaper und das [AWS Cloud Adoption Framework \(AWS CAF\)](#).

Protokollierung und Überwachung in Amazon Location Service

Protokollierung und Überwachung sind ein wichtiger Bestandteil der Reaktion auf Vorfälle. Auf diese Weise können Sie eine Sicherheitsbasis festlegen, um Abweichungen zu erkennen, die Sie untersuchen und darauf reagieren können. Durch die Implementierung von Protokollierung und

Überwachung für Amazon Location Service können Sie die Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Projekte und Ressourcen aufrechterhalten.

AWS bietet mehrere Tools, mit denen Sie Daten für die Reaktion auf Vorfälle protokollieren und sammeln können:

AWS CloudTrail

Amazon Location Service ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS Service ausgeführten Aktionen bereitstellt. Dazu gehören Aktionen von der Amazon Location Service Service-Konsole aus und programmatische Aufrufe von Amazon Location API-Operationen. Diese Aufzeichnungen von Aktionen werden als Ereignisse bezeichnet. Weitere Informationen finden Sie unter [Protokollierung und Überwachung von Amazon Location Service mit AWS CloudTrail](#).

Amazon CloudWatch

Sie können Amazon verwenden CloudWatch , um Kennzahlen zu Ihrem Amazon Location Service Service-Konto zu sammeln und zu analysieren. Sie können CloudWatch Alarme aktivieren, um Sie zu benachrichtigen, wenn eine Metrik bestimmte Bedingungen erfüllt und einen bestimmten Schwellenwert erreicht hat. Wenn Sie einen Alarm erstellen, CloudWatch sendet eine Benachrichtigung an einen von Ihnen definierten Amazon Simple Notification Service. Weitere Informationen finden Sie unter [Monitoring Amazon Location Service with Amazon CloudWatch](#).

AWS Health Dashboards

Mithilfe von [AWS Health Dashboards](#) können Sie den Status des Amazon Location Service Service überprüfen. Sie können auch historische Daten zu Ereignissen oder Problemen, die sich auf Ihre AWS Umgebung auswirken könnten, überwachen und einsehen. Weitere Informationen finden Sie im [AWS Health -Benutzerhandbuch](#).


Konformitätsprüfung für Amazon Location Service

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

 Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmapen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#) — Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen

wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.

- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz bei Amazon Location Service

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur bietet Amazon Location mehrere Funktionen, die Ihnen helfen, Ihre Anforderungen an Datenstabilität und Datensicherung zu erfüllen.

Infrastruktursicherheit in Amazon Location Service

Als verwalteter Service ist Amazon Location Service durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Location zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Konfiguration und Schwachstellenanalyse in Amazon Location

Konfiguration und IT-Kontrollen liegen in der gemeinsamen Verantwortung AWS von Ihnen, unserem Kunden. Weitere Informationen finden Sie im [Modell der AWS gemeinsamen Verantwortung](#).

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Amazon Location Service fungiert nicht als Anrufservice in Ihrem Namen für andere AWS Dienste, sodass Sie diese Schutzmaßnahmen in diesem Fall nicht hinzufügen müssen. Weitere Informationen über den verwirrten Stellvertreter finden Sie unter [Das Problem mit dem verwirrten Stellvertreter](#) im AWS Identity and Access Management Benutzerhandbuch.

Bewährte Sicherheitsmethoden für Amazon Location Service

Amazon Location Service bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Bewährte Methoden zur Detektivsicherheit für Amazon Location Service

Die folgenden bewährten Methoden für Amazon Location Service können dabei helfen, Sicherheitsvorfälle zu erkennen:

Implementieren Sie AWS-Überwachungstools

Die Überwachung ist entscheidend für die Reaktion auf Vorfälle und gewährleistet die Zuverlässigkeit und Sicherheit der Ressourcen von Amazon Location Service und Ihrer Lösungen. Sie können Überwachungstools aus den verschiedenen über AWS verfügbaren Tools und Services implementieren, um Ihre Ressourcen und Ihre anderen AWS-Services zu überwachen.

CloudWatch Mit Amazon können Sie beispielsweise Messwerte für Amazon Location Service überwachen und Alarme einrichten, um Sie zu benachrichtigen, wenn eine Metrik bestimmte von Ihnen festgelegte Bedingungen erfüllt und einen von Ihnen definierten Schwellenwert erreicht hat. Wenn Sie einen Alarm erstellen, können Sie festlegen, CloudWatch dass über Amazon Simple Notification Service eine Benachrichtigung gesendet wird. Weitere Informationen finden Sie unter [the section called “Protokollieren und Überwachen”](#).

AWS-Protokollierungstools aktivieren

Die Protokollierung bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon Location Service ausgeführt wurden. Sie können Protokollierungstools implementieren AWS CloudTrail , um beispielsweise Daten über Aktionen zu sammeln, um ungewöhnliche API-Aktivitäten zu erkennen.

Wenn Sie einen Trail erstellen, können Sie ihn so konfigurieren, CloudTrail dass Ereignisse protokolliert werden. Ereignisse sind Aufzeichnungen von Ressourcenoperationen, die auf oder innerhalb einer Ressource ausgeführt wurden, wie z. B. die Anfrage an Amazon Location, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann die Anfrage gestellt wurde, zusammen mit zusätzlichen Daten. Weitere Informationen finden Sie unter [Protokollierung von Datenereignissen für Trails](#) im AWS CloudTrail Benutzerhandbuch.

Bewährte Methoden zur präventiven Sicherheit für Amazon Location Service

Die folgenden bewährten Methoden für Amazon Location Service können dazu beitragen, Sicherheitsvorfälle zu verhindern:

Verwenden Sie sichere Verbindungen

Verwenden Sie immer verschlüsselte Verbindungen, z. B. solche, die mit `https://` beginnen, um vertrauliche Informationen bei der Übertragung zu schützen.

Implementieren Sie den Zugriff auf Ressourcen mit den geringsten Rechten

Wenn Sie benutzerdefinierte Richtlinien für Amazon-Standortressourcen erstellen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Es wird empfohlen, mit einem Mindestsatz an Berechtigungen zu beginnen und bei Bedarf zusätzliche Berechtigungen zu gewähren. Die Implementierung des Zugriffs mit den geringsten Rechten ist unerlässlich, um das Risiko und die Auswirkungen von Fehlern oder böswilligen Angriffen zu verringern. Weitere Informationen finden Sie unter [the section called “Identitäts- und Zugriffsverwaltung”](#).

Verwenden Sie weltweit eindeutige IDs als Geräte-IDs

Verwenden Sie die folgenden Konventionen für Geräte-IDs.

- Geräte-IDs müssen eindeutig sein.
- Geräte-IDs sollten nicht geheim sein, da sie als Fremdschlüssel für andere Systeme verwendet werden können.
- Geräte-IDs sollten keine persönlich identifizierbaren Informationen (PII) wie Telefongeräte-IDs oder E-Mail-Adressen enthalten.
- Geräte-IDs sollten nicht vorhersehbar sein. Undurchsichtige Identifikatoren wie UUIDs werden empfohlen.

Nehmen Sie keine personenbezogenen Daten in die Eigenschaften der Geräteposition auf

Geben Sie beim Senden von Geräteaktualisierungen (z. B. mithilfe von [DevicePositionUpdate](#)) keine personenbezogenen Daten (PII) wie Telefonnummer oder E-Mail-Adresse in das `PositionProperties`.

Bewährte Methoden für Amazon Location Service

In diesem Thema finden Sie bewährte Methoden zur Verwendung von Amazon Location Service. Diese bewährten Methoden können Ihnen zwar dabei helfen, den Amazon Location Service in vollem Umfang zu nutzen, stellen jedoch keine vollständige Lösung dar. Sie sollten nur die Empfehlungen befolgen, die für Ihre Umgebung gelten.

Themen

- [Sicherheit](#)
- [Ressourcenmanagement](#)
- [Fakturierung und Kostenmanagement](#)
- [Kontingente und Nutzung](#)

Sicherheit

Beachten Sie die folgenden bewährten Methoden, um Sicherheitsrisiken zu bewältigen oder sogar zu vermeiden:

- Verwenden Sie Identity Federation und IAM-Rollen, um den Zugriff auf Ihre Amazon- Standortressourcen zu verwalten, zu kontrollieren oder einzuschränken. Weitere Informationen finden Sie unter [Bewährte Methoden für IAM](#) im IAM-Benutzerhandbuch.
- Folgen Sie dem Prinzip der geringsten Rechte, um nur den erforderlichen Mindestzugriff auf Ihre Amazon Location Service Service-Ressourcen zu gewähren. Weitere Informationen finden Sie unter [the section called “Verwalten des Zugriffs mit Richtlinien”](#).
- Beschränken Sie den Zugriff auf Amazon Location Service Service-Ressourcen, die in Webanwendungen verwendet werden, mithilfe einer `aws:referer` IAM-Bedingung und beschränken Sie die Nutzung durch Websites, die nicht in der Zulassungsliste aufgeführt sind.
- Verwenden Sie Überwachungs- und Protokollierungstools, um den Zugriff auf und die Nutzung von Ressourcen zu verfolgen. Weitere Informationen finden Sie unter [the section called “Protokollieren und Überwachen”](#) und [Protokollierung von Datenereignissen für Trails](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie sichere Verbindungen, z. B. solche, die mit `https://` beginnen, um die Sicherheit zu erhöhen und Benutzer vor Angriffen zu schützen, während Daten zwischen dem Server und dem Browser übertragen werden.

Weitere Informationen zu bewährten Methoden für detektive und präventive Sicherheit finden Sie im Thema unter [the section called “Bewährte Methoden für die Gewährleistung der Sicherheit”](#).

Ressourcenmanagement

Beachten Sie die folgenden bewährten Methoden, um Ihre Standortressourcen in Amazon Location Service effektiv zu verwalten:

- Verwenden Sie regionale Endpunkte, die für Ihre erwartete Benutzerbasis von zentraler Bedeutung sind, um deren Benutzererfahrung zu verbessern. Informationen zu regionalen Endpunkten finden Sie unter [Amazon-Standortregionen und Endpunkte](#)
- Beachten Sie bei Ressourcen, die Datenanbieter verwenden, wie Kartenressourcen und Ortsindexressourcen, die Nutzungsbedingungen des jeweiligen Datenanbieters. Weitere Informationen finden Sie unter [Datenanbieter](#).
- Minimiere die Erstellung von Ressourcen, indem du für jede Konfiguration von Karte, Ortsindex oder Routen eine Ressource zur Verfügung hast. Innerhalb einer Region benötigen Sie in der Regel nur eine Ressource pro Datenprovider oder Kartenstil. Die meisten Anwendungen verwenden vorhandene Ressourcen und erstellen zur Laufzeit keine Ressourcen.
- Wenn Sie in einer einzigen Anwendung unterschiedliche Ressourcen verwenden, z. B. eine Kartenressource und einen Routenrechner, verwenden Sie in jeder Ressource denselben Datenanbieter, um sicherzustellen, dass die Daten übereinstimmen. Beispiel: Eine Routengeometrie, die Sie mit Ihrem Routenrechner erstellen, stimmt mit den Straßen auf der Karte überein, die mit der Kartenressource gezeichnet wurde.

Fakturierung und Kostenmanagement

Beachten Sie die folgenden bewährten Methoden, um Ihre Kosten und Ihre Abrechnung besser zu verwalten:

- Verwenden Sie Überwachungstools wie Amazon CloudWatch, um Ihren Ressourcenverbrauch zu verfolgen. Sie können Warnmeldungen einrichten, die Sie benachrichtigen, wenn die Nutzung Ihre angegebenen Grenzwerte bald überschreitet. Weitere Informationen finden Sie unter [Erstellen eines Fakturierungsalarms zur Überwachung Ihrer geschätzten AWS-Gebühren](#) im CloudWatch Amazon-Benutzerhandbuch.

Kontingente und Nutzung

Sie schließen AWS-Konto Kontingente ein, die eine Standardbegrenzung für Ihre Nutzungsmenge festlegen. Sie können Alarmer einrichten, die Sie benachrichtigen, wenn sich Ihre Nutzung Ihrem Limit nähert, und Sie können bei Bedarf eine Erhöhung eines Kontingents beantragen. Informationen zum Arbeiten mit Kontingenten finden Sie in den folgenden Themen.

- [Kontingente für Amazon Location Service](#)
- [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#)

- [Visualisieren Sie Ihre Servicekontingente und stellen Sie Alarme](#) ein im CloudWatch Amazon-Benutzerhandbuch.

Sie können Alarme einrichten, um Sie im Voraus zu warnen, wenn Sie kurz davor sind, Ihre Limits zu überschreiten. Wir empfehlen, Alarme für jedes Kontingent in jedem Standort einzurichten AWS-Region , an dem Sie Amazon Location verwenden. Sie können beispielsweise Ihre Nutzung des SearchPlaceIndexForText Vorgangs überwachen und einen Alarm auslösen, wenn Sie 80 Prozent Ihres aktuellen Kontingents überschreiten.

Wenn Sie eine Alarmwarnung bezüglich Ihres Kontingents erhalten, müssen Sie entscheiden, was zu tun ist. Möglicherweise benötigen Sie zusätzliche Ressourcen, weil Ihr Kundenstamm gewachsen ist. In diesem Fall möchten Sie möglicherweise eine Erhöhung Ihres Kontingents beantragen, z. B. eine 50-prozentige Erhöhung des Kontingents für einen API-Aufruf in dieser Region. Oder vielleicht liegt ein Fehler in Ihrem Service vor, der dazu führt, dass Sie zusätzliche unnötige Anrufe an Amazon Location tätigen. In diesem Fall möchten Sie das Problem in Ihrem Service lösen.

Dokumentverlauf

In der folgenden Tabelle wird die Dokumentation für Amazon Location Service beschrieben. Um Benachrichtigungen über Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Amazon Location Service veröffentlicht ein neues SDK für JavaScript	Um die Entwicklung von Amazon Location-Anwendungen mit in Web-Frontends zu vereinfachen, fügt Amazon Location ein neues Open-Source-SDK hinzu, das AWS SDK for JavaScript v3 unterstützt, wodurch die Authentifizierung und die Verwendung von GeoJSON vereinfacht werden. Weitere Informationen finden Sie unter Amazon Location SDK .	6. Juli 2023
Amazon Location Service veröffentlicht API-Schlüssel für allgemeine Verfügbarkeit	Amazon Location bietet Unterstützung für Ort und Route und kündigt die allgemeine Verfügbarkeit der API-Schlüsselfunktion an. Weitere Informationen finden Sie unter Verwenden von API-Schlüsseln .	6. Juli 2023
Amazon Location Service fügt Amazon- EventBridge Ereignisse für Positionsaktualisierungen hinzu	Amazon Location fügt Unterstützung für das Senden von Tracker-Positionsaktualisierungsereignissen an hinzu EventBridge. Weitere Informationen, einschließlich der Aktivierung der Ereigniss	6. Juli 2023

	e für einen Tracker, finden Sie unter Reagieren auf Ereignisse mit EventBridge .	
Amazon Location fügt Geofences Metadaten hinzu	Mit der Amazon Location API können Sie jetzt Metadaten eigenschaften zu Ihren Geofences hinzufügen. Diese werden mit Ihrem Geofence gespeichert und in Ereignissen im Zusammenhang mit dem Geofence in Amazon aufgenommen EventBridge. Weitere Informationen finden Sie unter Zeichnen von Geofences und Erstellen von Ereignisregeln .	15. Juni 2023
Amazon Location fügt Kategorien für Orte hinzu	Amazon Location fügt Kategorien zu Suchergebnissen hinzu und filtert Ergebnisse nach Kategorien. Weitere Informationen finden Sie unter Kategorien und Filtern von .	15. Juni 2023
Amazon Location führt Polnische Ansichten ein	Amazon Location fügt bestimmten Kartenstilen öffentliche Ansichten hinzu. Weitere Informationen finden Sie unter Politische Ansichten .	23. Mai 2023

[Amazon Location führt neue Demo- und Beispielwebsite ein](#)

Amazon Location kündigt eine neue Website an, die Ihnen Zugriff auf Amazon Location-Demos und -Beispiele bietet. Weitere Informationen finden Sie auf der [Demo-Website von Amazon Location](#).

3. Mai 2023

[Amazon Location führt längere Routen in ein Calculate RouteMatrix](#)

Amazon Location ermöglicht jetzt Routen mit unbegrenzter Länge für Routenmatrixrouten, die mit dem HERE-Date nanbieter erstellt wurden. Weitere Informationen finden Sie unter [Längere Routenplanung](#).

24. April 2023

[Die Amazon Location-Dokumentation fügt Funktionsunterschiede nach Datenanbieter hinzu](#)

Die Dokumentation zu Amazon Location wurde mit Informationen zu den Unterschieden zwischen den einzelnen Datenanbietern in Karten, Ortssuche und Routing aktualisiert. Weitere Informationen finden Sie unter [Funktionen nach Datenanbieter](#).

30. März 2023

[Amazon Location Open Data ordnet die allgemeine Verfügbarkeit zu](#)

Allgemeine Verfügbarkeit des Datenanbieters und -OpenStreetMapstils von Amazon Location Service, basierend auf den Sommerkarten von . Weitere Informationen finden Sie unter [Open Data](#).

7. März 2023

[Amazon Location fügt neue Autorisierungsmethode in der Vorschau hinzu](#)

Amazon Location Service fügt API-Schlüssel als neue Autorisierungsmethode für anonyme Benutzer im Vorschaumodus hinzu. Weitere Informationen finden Sie unter [Zulassen des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).

23. Februar 2023

[Die Dokumentation zu Amazon Location wurde mit den neuesten bewährten IAM-Methoden aktualisiert](#)

Die Dokumentation zu Amazon Location Service wurde aktualisiert, um den neuesten AWS Identity and Access Management bewährten Methoden zu entsprechen. Weitere Informationen finden Sie unter [Sicherheit in Amazon Location Service](#).

26. Januar 2023

[Amazon Location Service fügt GrabMaps als Datenanbieter in Südostasien hinzu](#)

Amazon Location führt GrabMaps als Datenanbieter in Südostasien ein. Weitere Informationen finden Sie unter [GrabMaps](#).

10. Januar 2023

[Open-Data-Übersichten von Amazon Location Service in der Vorschau](#)

Neuer Amazon Location-Datenanbieter und -Stil wurden in der öffentlichen Vorschau hinzugefügt, basierend auf OpenStreetMapden Daylight-Übersichten von . Weitere Informationen finden Sie unter [Open Data \(Vorschau\)](#).

15. Dezember 2022

Neue HERE-Satellitenbilder	Für Karten, die HERE als Datenanbieter verwenden, wurden zwei neue Kartenstile hinzugefügt: HERE microSDe Imagery und HERE Hybrid Map styles. Weitere Informationen finden Sie unter HERE-Kartenstile .	25. Oktober 2022
Einheiten in Adressen	Amazon Location Service unterstützt jetzt Einheiten in Adressen, z. B. „123 Main St, Apartment 3B , Beliebig, USA“.	20. September 2022
Abrufen von Orten nach ID	Amazon Location Service unterstützt jetzt die Suche nach dem genauen Speicherort, den die SearchPlaceIndexForSuggestions Operation mithilfe der GetPlace Operation vorschlägt. Weitere Informationen finden Sie unter Verwenden der automatischen Vervollständigung von .	20. September 2022
Zusätzliche Bedingungsschlüssel für die IAM-Richtlinie	Amazon Location Service unterstützt jetzt zusätzliche Bedingungsschlüssel, mit denen Sie den Zugriff für bestimmte Geofences oder Geräte in der IAM-Richtlinie festlegen können. Siehe Bedingungsschlüssel .	23. August 2022

[Lokale Geofences](#)

Amazon Location Service unterstützt jetzt Geofences, die als Kreis mit einem Mittelpunkt und einem Radius definiert sind, um Ereignisse abzurufen, wenn sich Geräte innerhalb einer bestimmten Entfernung von einem Standort befinden. Weitere Informationen finden Sie unter [Hinzufügen von Zirkelgeofences](#).

11. August 2022

[Kombinierte API-Referenz](#)

Amazon Location Service verfügt jetzt über ein einziges API-Referenzhandbuch und nicht über separate Leitfäden für jeden Subservice. Weitere Informationen zu den APIs finden Sie unter [Amazon Location APIs](#).

7. Juli 2022

[Integration von Service Quotas](#)

Amazon Location ist jetzt [in Service Quotas integriert](#), sodass Sie Ihre Kontingente über die AWS Management Console oder mithilfe der anzeigen und verwalten können AWS CLI.

6. Juli 2022

[Kapitel zur Dokumentation aktualisierter Konzepte](#)

Das [Kapitel Amazon-Standortkonzepte](#) wurde mit weiteren Informationen für Benutzer von Amazon Location aktualisiert.

22. April 2022

Neues Android-Schnellstart-Tutorial	Ein neues Schnellstart-Tutorial für die Android-Entwicklung mit Kotlin wurde hinzugefügt, um Entwicklern einen schnellen Einstieg zu ermöglichen.	15. April 2022
Neue HERE-Map-Stile	Für Karten, die HERE als Datenanbieter verwenden, wurden zwei neue Kartenstile hinzugefügt. Weitere Informationen finden Sie unter HERE-Kartenstile .	15. März 2022
Dokumentation mit zusätzlichen Codebeispielen und Tutorials umstrukturieren	Dieses Entwicklerhandbuch wurde umstrukturiert, um Themen einfacher zu finden, einschließlich der neuen Kapitel Schnellstart und Codebeispiele .	25. Februar 2022
Genauigkeitsbasierte Positionsfiltrierung für Tracker	Sie können jetzt genauigkeitsbasierte Filter verwenden, wenn Sie eine Tracker-Ressource erstellen .	7. Dezember 2021
Automatische Vervollständigung für Ortsindizes	Sie können jetzt die automatische Vervollständigung verwenden, wenn Sie Ortsindizes durchsuchen.	6. Dezember 2021

[Neues Amplify-Tutorial für die Verwendung von Karten](#)

Es ist ein neues Tutorial verfügbar, das zeigt, wie Sie verwenden, AWS Amplify um Karten in einer Webanwendung anzuzeigen. Das Tutorial finden Sie unter [Verwenden der Amplify-Bibliothek mit Amazon Location Service](#).

24. November 2021

[Abfrageerweiterungen platzieren](#)

Amazon Location Service unterstützt jetzt das Festlegen einer bevorzugten Sprache für Ergebnisse bei Geokodierung oder Reverse-Geokodierung und fügt die Zeitzone und andere Informationen zu den Ergebnissen hinzu. Weitere Informationen zur Geokodierung und umgekehrten Geokodierung finden Sie unter [Geokodierung, umgekehrte Geokodierung und Suche nach](#) .

16. November 2021

[Tracker-Positionsfilterung](#)

Amazon Location Service fügt den Trackern eine neue Positionsfilterfunktion hinzu, mit der Sie die Kosten kontrollieren können. Diese Funktion filtert einige Positionsaktualisierungen auf Geräten heraus, bevor die Updates gespeichert oder anhand von Geofences ausgewertet werden. Weitere Informationen zur Positionsfilterung finden Sie unter [Tracker](#).

5. Oktober 2021

[Aktualisierungsvorgänge](#)

Die folgenden Operationen wurden den API-Referenzen für Amazon Location Service hinzugefügt: [UpdateMap](#), [UpdatePlaceIndexUpdateRouteCalculator](#) [UpdateGeofenceCollection](#), und [UpdateTracker](#).

19. Juli 2021

[Tutorial-Update: Amazon Aurora PostgreSQL benutzerdefinierte Funktionen](#)

Ein neues Tutorial zur Verwendung [Amazon Aurora PostgreSQL benutzerdefinierte Funktionen mit Amazon Location](#) zum Validieren, Bereinigen und Anreichern von Geodaten wurde hinzugefügt.

19. Juli 2021

[AWS- CloudFormation Ressourcen](#)

Amazon Location unterstützt jetzt [das Erstellen der folgenden Ressourcentypen in AWS CloudFormation-Ressourcen](#): `AWS::Location::Map` , `AWS::Location::PlaceIndex` , `AWS::Location::RouteCalculator` , `AWS::Location::Tracker` `AWS::Location::TrackerConsumer` , und `AWS::Location::GeofenceCollection` .

7. Juni 2021

[Markieren von Ressourcen](#)

Sie können [Ihren Amazon Location-Ressourcen jetzt Tags hinzufügen](#), um Ihre Ressourcen zu verwalten, zu identifizieren, zu organisieren, zu suchen und zu filtern.

1. Juni 2021

[Allgemeine Verfügbarkeit](#)

Allgemeine Verfügbarkeit der Entwicklerdokumentation zu Amazon Location Service: [Region und Endpunkte](#) und [Service Quotas](#) aktualisiert.

1. Juni 2021

[Esri-Imagery](#)

Amazon Location unterstützt jetzt die Verwendung des Esri-Kartenstils: [Esri Imagery](#) . Weitere Informationen finden Sie unter [Esri World Imagery](#) auf der Esri-Website.

1. Juni 2021

[Berechnen von Routen](#)

Sie können jetzt [Routenrec hner von Amazon Location verwenden](#), um Routen zu berechnen und die Reisezeit auf der Grundlage von up-to-date Straßennetz- und Live-Datenverkehrsinformationen des von Ihnen ausgewählten Datenanbieters zu schätzen.

1. Juni 2021

[AWS KMS Kundenverwaltete Schlüsselverschlüsselung für Daten im Ruhezustand](#)

Amazon Location unterstützt jetzt die Verwendung eines symmetrischen kundenverwalteten Schlüssels, den Sie erstellen, besitzen und verwalten, um [eine zweite Verschlüsselungsebene über die vorhandene -AWSeigene Verschlüsselung hinzuzufügen](#).

1. Juni 2021

[Veröffentlichung der öffentlichen Vorschau](#)

Erstveröffentlichung der öffentlichen Vorschau dokumentation.

16. Dezember 2020

[Tutorial-Update: Anzeigen von Karten](#)

Tutorials zum Anzeigen von Karten mit MapLibre für Android und iOS wurden aktualisiert, um das MapLibre native SDK zu verwenden.

17. März 2020

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.