



Guia do Desenvolvedor

# Amazon Location Service



# Amazon Location Service: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

---

# Table of Contents

Bem-vindo .....	1
O que é o Amazon Location Service? .....	1
Principais atributos .....	2
Serviços relacionados .....	3
Início rápido .....	5
Criar um aplicativo Web .....	5
Criar recursos do .....	6
Configurar autenticação .....	7
Criando HTML .....	9
Adicionar o mapa .....	12
Adicionar pesquisas .....	16
Aplicativo final .....	21
Próximas etapas .....	26
Criar um aplicativo Android .....	26
Criar recursos do Amazon Location para seu aplicativo .....	26
Configurar autenticação .....	28
Crie o aplicativo .....	31
Adicione o mapa .....	32
Adicionar pesquisa .....	35
Adicionar rastreamento .....	45
Próximas etapas .....	54
Criação de um aplicativo para iOS .....	54
Criar recursos do .....	55
Configurar autenticação .....	57
Criar o aplicativo .....	59
Código inicial .....	60
Adicionar um mapa .....	63
Adicionar pesquisa .....	67
Adicionar rastreamento .....	68
Próximas etapas .....	80
Conceitos do Amazon Location .....	81
Visão geral .....	82
Mapas .....	83
Estilos de mapa .....	84

Visões políticas .....	84
Camadas personalizadas .....	85
Renderização de mapas .....	85
Terminologia dos mapas .....	86
Pesquisa de lugares .....	88
Conceitos de geocodificação .....	89
Resultados da pesquisa .....	90
Múltiplos resultados e relevância .....	90
Resultados de endereço .....	91
Armazenamento de resultados de geocodificação .....	93
Terminologia de locais .....	94
Rotas .....	95
Recursos da calculadora de rotas .....	95
Cálculo de uma rota .....	96
Planejamento de rotas .....	97
Terminologia de rotas .....	98
Geocercas e rastreadores .....	100
Geocercas .....	100
Rastreadores .....	102
Terminologia de geocercas .....	106
Terminologia do rastreador .....	107
Casos de uso comuns .....	109
Aplicativos de engajamento do usuário e geomarketing .....	110
Aplicativos de rastreamento de ativos .....	112
Aplicativos de entrega .....	113
Provedores de dados .....	115
Cobertura e atributos do provedor de dados .....	115
Estilos de mapa .....	117
Mais detalhes .....	117
Esri .....	117
GrabMaps .....	126
HERETecnologias .....	131
Open Data .....	138
Recursos por provedor de dados .....	148
Termos de uso e atribuição de dados .....	153
Regiões e endpoints .....	154

Regiões .....	154
Endpoints .....	156
APIEndpoints de operação .....	156
Cotas de serviço .....	158
Gerenciar suas cotas do Amazon Location Service .....	172
Desenvolvendo com o Amazon Location .....	174
Cenários e casos de uso .....	175
SDKs e ferramentas .....	176
SDKs por linguagem .....	176
MapLibre .....	180
SDKs do Amazon Location .....	185
Amazon Location .....	209
Usando o Amazon Location com um AWS SDK .....	210
Atualizações de mensagens de erro .....	210
Exemplos de código .....	244
Site de demonstração do Amazon Location .....	245
Tutorial: início rápido .....	246
Tutorial: enriquecimento do banco de dados .....	247
Exemplo: explore o aplicativo .....	247
Exemplo: estilize um mapa .....	248
Exemplo: desenhe marcadores .....	248
Exemplo: desenhe pontos agrupados .....	249
Exemplo: desenhe um polígono .....	249
Exemplo: altere o idioma do mapa .....	250
Blog: Estimated delivery time notifications (Notificações de tempo estimado de entrega) .....	251
Exemplo: atualizações da posição do fluxo .....	252
Exemplo: aplicativo móvel de geofencing and tracking .....	252
Como usar o Amazon Location .....	254
Pré-requisitos da conta .....	255
Inscreva-se para um Conta da AWS .....	255
Criar um usuário com acesso administrativo .....	256
Conceder acesso ao Amazon Location Service .....	257
Usando mapas .....	259
Pré-requisitos .....	260
Exibindo mapas .....	263
Desenho em um mapa .....	318

Extensões de configurações para um mapa .....	319
Gerenciando recursos do mapa .....	320
Pesquisa de lugares .....	324
Pré-requisitos .....	325
Geocodificação .....	328
Geocodificação reversa .....	336
Autocompletar .....	340
Usando IDs de locais .....	347
Categorias e filtragem .....	348
Tutorial: enriquecimento do banco de dados .....	353
Gerenciar recursos do índice de locais .....	368
Calculando rotas .....	371
Pré-requisitos .....	372
Calcular rota .....	376
Planejamento de rotas .....	380
Posições não localizadas em uma estrada .....	386
Horário de partida .....	388
Modo de viagem .....	389
Gerenciando os recursos de rotas .....	391
Geocercamento e rastreamento .....	394
Etapa 1: adicione geocercas .....	396
Etapa 2: inicie o rastreamento .....	403
Etapa 3: vincule um rastreador a uma coleção de geocercas .....	417
Etapa 4: avalie as posições dos dispositivos em relação às geocercas .....	419
Verifique as posições do dispositivo .....	422
Reagindo aos eventos com EventBridge .....	424
Rastreando usando AWS IoT e MQTT .....	430
Gerenciar recursos de geocerca .....	438
Gerenciando os recursos de rastreador .....	446
Exemplo de aplicativo móvel de geofencing and tracking .....	450
Marcando seus recursos .....	469
Restrições .....	470
Conceder permissão para marcar .....	471
Adicionar uma tag a um recurso .....	472
Rastreie o custo por tag .....	472
Controlar o acesso a recursos da usando tags .....	473

---

Saiba mais .....	474
Conceder acesso ao Amazon Location .....	474
Usando chaves de API .....	475
Usando Amazon Cognito .....	482
Monitorar o Amazon Location Service .....	492
Monitoramento com CloudWatch .....	493
Usando CloudTrail com a Amazon Location .....	498
Usando AWS CloudFormation para criar recursos .....	503
Modelos Amazon Location e AWS CloudFormation .....	503
Saiba mais sobre a AWS CloudFormation .....	504
Segurança .....	505
Proteção de dados .....	506
Privacidade de dados .....	507
Retenção de dados .....	507
Criptografia de dados em repouso .....	507
Criptografia de dados em trânsito .....	521
Identity and Access Management .....	521
Público .....	522
Autenticando com identidades .....	522
Gerenciando acesso usando políticas .....	526
Como o Amazon Location Service funciona com IAM .....	529
Como o Amazon Location Service funciona com usuários não autenticados .....	537
Exemplos de políticas baseadas em identidade .....	537
Solução de problemas .....	550
Resposta a incidentes .....	552
Registro e Monitoramento .....	552
Validação de conformidade .....	553
Resiliência .....	554
Segurança da infraestrutura .....	555
Análise de configuração e vulnerabilidade .....	555
Prevenção contra representante confuso .....	556
Melhores práticas de segurança .....	556
Melhores práticas de detecção .....	556
Melhores práticas preventivas .....	557
Melhores Práticas .....	558
Segurança .....	558

---

Gerenciamento de recursos .....	559
Faturamento e gerenciamento de custos .....	560
Cotas e uso .....	560
Histórico do documentos .....	561
AWS Glossário .....	572
.....	dlxxiii



# Bem-vindo ao Amazon Location Service

Bem-vindo ao Guia do desenvolvedor do Amazon Location Service.

Os tópicos a seguir podem ajudar você a começar a usar a documentação com base no que você está tentando fazer.

Tenha uma visão geral do Amazon Location

- Saiba mais sobre os [conceitos no Amazon Location](#).
- Mergulhe mais profundamente em sua funcionalidade no capítulo [Como usar o Amazon Location Service](#).
- Veja os aplicativos de demonstração no [site de demonstração do Amazon Location](#).
- Se você já tem um Conta da AWS, você pode usar o [console do Amazon Location Service](#) para explorar sua funcionalidade em primeira mão.

Use o Amazon Location como desenvolvedor

- Crie seu primeiro aplicativo com o [Início rápido](#).
- Saiba como os vários atributos do Amazon Location Service funcionam no capítulo [Como usar o Amazon Location Service](#).
- Veja os SDKs e as ferramentas disponíveis para você no capítulo [Desenvolvendo com o Amazon Location](#).
- Veja [exemplos de código e tutoriais](#) que você pode usar em seus próprios aplicativos. Você também pode visitar a [página de amostras](#) do site de demonstração do Amazon Location para encontrar amostras, filtráveis por atributo, linguagem ou plataforma.
- Obtenha informações sobre as APIs do Amazon Location no [Guia de referência de APIs](#).

## O que é o Amazon Location Service?

O Amazon Location Service permite que você adicione dados e funcionalidades de localização aos aplicativos, o que inclui recursos como mapas, pontos de interesse, geocodificação, roteamento, geocercas e rastreamento. A Amazon Location fornece serviços baseados em localização (SBL) usando dados de alta qualidade dos provedores globais e confiáveis Esri, Grab e HERE. Com dados acessíveis, recursos de rastreamento e de geocercas e métricas integradas para monitoramento de saúde, você pode criar aplicativos sofisticados habilitados para a localização.

Com o Amazon Location, você retém o controle dos dados da sua organização. O Amazon Location torna anônimas todas as consultas enviadas aos provedores de dados, removendo metadados do cliente e informações da conta. Além disso, as informações confidenciais de localização de rastreamento e de geocercas, como a localização de instalações, ativos e funcionários, não saem da sua conta AWS. Isso ajuda você a proteger informações confidenciais de terceiros, proteger a privacidade do usuário e reduzir os riscos de segurança do seu aplicativo. Com o Amazon Location, a Amazon e terceiros não têm o direito de vender seus dados ou usá-los para publicidade.

O Amazon Location é totalmente integrado a serviços como Amazon AWS CloudTrail CloudWatch EventBridge, Amazon e AWS Identity and Access Management (IAM). O Amazon Location simplifica seu fluxo de trabalho de desenvolvimento com integração de dados e acelera a produção de aplicativos com atributos integrados de monitoramento, segurança e conformidade.

Para obter destaques, detalhes e preços do produto, consulte a página de serviços do [Amazon Location Service](#).

## Principais atributos do Amazon Location

O Amazon Location fornece os seguintes atributos:

### Mapas

O Amazon Location Service Maps permite que você visualize informações de localização e é a base de muitos recursos de serviços baseados em localização. O Amazon Location Service fornece blocos de mapas de diferentes estilos provenientes dos provedores de dados de localização global Esri, Grab e HERE, além de mapas de dados abertos.

### Locais

O Amazon Location Service Places permite que você integre a funcionalidade de pesquisa em seu aplicativo, converta endereços em coordenadas geográficas em latitude e longitude (geocodificação) e converta uma coordenada em um endereço (geocodificação reversa). O Amazon Location Service obtém dados geoespaciais de alta qualidade do Esri, Grab e HERE para dar suporte para as funções do Places.

### Roteamento

O Amazon Location Service Routes permite que você encontre rotas e estime o tempo de viagem com base nas informações da up-to-date estrada e do trânsito ao vivo. Crie atributos que

permitam que seu aplicativo solicite o tempo de viagem, a distância e o caminho entre dois locais. Calcule o tempo e a distância de uma matriz de rotas a serem usadas no planejamento de rotas.

## Geocercas

O Amazon Location Service Geofences permite que você dê ao seu aplicativo a capacidade de detectar e agir quando um dispositivo entra ou sai de um limite geográfico definido, chamado de geocerca. Envie automaticamente um evento de entrada ou saída para a Amazon EventBridge quando uma violação de geofence for detectada. Isso permite que você inicie ações posteriores, como enviar uma notificação a um alvo.

## Rastreadores

O Amazon Location Service Trackers permite que você recupere a localização atual e histórica dos dispositivos que estão executando seu aplicativo habilitado para rastreamento. Você também pode vincular rastreadores às geocercas do Amazon Location Service para avaliar automaticamente as atualizações de localização dos seus dispositivos em relação às suas geocercas. Os rastreadores podem ajudar você a reduzir custos filtrando as atualizações de posição que não foram movidas antes de armazená-las ou avaliá-las em relação às geocercas.

Quando você usa rastreadores, as informações confidenciais de localização em seus dispositivos rastreados não saem da sua conta AWS. Isso ajuda você a proteger informações confidenciais de terceiros, proteger a privacidade do usuário e reduzir os riscos de segurança.

# Serviços que você pode usar com o Amazon Location

Use os seguintes serviços junto com o Amazon Location Service.

## Monitoramento e gerenciamento integrado

O Amazon Location Service é integrado à Amazon CloudWatch, AWS CloudTrail, e à Amazon EventBridge para monitoramento e gerenciamento de dados eficientes:

- Amazon CloudWatch — Visualize métricas sobre uso e integridade do serviço, incluindo solicitações, latência, falhas e registros. Para ter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#).

- AWS CloudTrail – log e monitore suas chamadas de API, que incluem ações executadas por um usuário, um cargo ou um serviço da AWS. Para ter mais informações, consulte [the section called “Usando CloudTrail com a Amazon Location”](#).
- Amazon EventBridge — Habilite uma arquitetura de aplicativo orientada por eventos para que você possa usar AWS Lambda funções para ativar outras partes do seu aplicativo e fluxos de trabalho. Para ter mais informações, consulte [the section called “Reagindo aos eventos com EventBridge”](#).

## Ferramentas de desenvolvedor

O Amazon Location Service oferece uma variedade de ferramentas para desenvolvedores criarem aplicativos habilitados para localização. Isso inclui os AWS SDKs padrão, os SDKs móveis e da web e o código de amostra para combiná-los com bibliotecas de código aberto, como MapLibre. Use o [console do Amazon Location Service](#) para aprender sobre recursos e começar a usar uma ferramenta de aprendizado visual e interativa.

# Início rápido com o Amazon Location Service

A maneira mais eficiente de começar a usar o Amazon Location Service é usar o [console Amazon Location](#). Você pode criar e gerenciar seus recursos e experimentar a funcionalidade do Amazon Location usando [a página Explore](#).

## Note

Para usar o console do Amazon Location Service ou seguir o restante deste tutorial, é necessário primeiro concluir o [Pré-requisitos para usar o Amazon Location Service](#), incluindo a criação de uma AWS conta e a permissão de acesso ao Amazon Location.

Para começar a aprender sobre as APIs do Amazon Location, use o tutorial a seguir para criar um aplicativo simples que exibe um mapa interativo e usa a funcionalidade de pesquisa. Há três versões do tutorial: uma mostra como criar uma página web simples usando JavaScript, a segunda mostra o mesmo para um aplicativo Android usando Kotlin e a terceira mostra o mesmo para um aplicativo iOS usando Swift.

## Tópicos

- [Criar um aplicativo Web](#)
- [Criar um aplicativo Android](#)
- [Criação de um aplicativo para iOS](#)

## Criar um aplicativo Web

Nesta seção, você criará uma página da web estática com um mapa e a capacidade de pesquisar em um local. Primeiro, você criará seus recursos de localização da Amazon e criará uma API chave para seu aplicativo.

## Tópicos

- [Criar recursos do Amazon Location para seu aplicativo](#)
- [Configurar a autenticação para seu aplicativo](#)
- [Criando o HTML para seu aplicativo](#)
- [Adicionar um mapa interativo ao seu aplicativo](#)

- [Adicionar pesquisas ao seu aplicativo](#)
- [Visualizar o aplicativo final](#)
- [Próximas etapas](#)

## Criar recursos do Amazon Location para seu aplicativo

Se você ainda não os tiver, você deverá criar os recursos do Amazon Location que seu aplicativo utilizará. Aqui, você cria um recurso de mapa para exibir mapas em seu aplicativo e um índice de local para buscar locais no mapa.

Para adicionar recursos de localização ao seu aplicativo


1. Escolha o estilo de mapa que deseja utilizar.
  - a. No console do Amazon Location, na página [Mapas](#), selecione Criar mapa para visualizar estilos de mapa.
  - b. Adicione um Nome e uma Descrição para o novo recurso de mapa. Anote o nome que usa para o recurso do mapa. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
  - c. Selecione um mapa.

### Note

Escolher um estilo de mapa também seleciona qual provedor de dados de mapa você usará. Se seu aplicativo estiver rastreando ou roteando ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você só poderá usá-lo HERE como seu provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).

- d. Aceite os Termos e condições do Amazon Location e selecione Criar mapa. Você pode interagir com o mapa que escolheu: ampliar, reduzir o zoom ou fazer uma panorâmica em qualquer direção.
  - e. Anote o nome do recurso da Amazon (ARN) que é exibido para seu novo recurso de mapa. Você o usará para criar a autenticação correta posteriormente neste tutorial.
2. Escolha o índice de local que deseja usar.

- a. No console do Amazon Location, na página de [Índices de local](#), selecione Criar índice de local.
- b. Adicione um Nome e uma Descrição para o novo recurso de índice de local. Anote os nomes que você utilizar para o recurso de índice de local. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
- c. Selecione um provedor de dados.

 Note

Na maioria dos casos, escolha o provedor de dados que corresponda ao provedor de mapa que você já escolheu. Isso ajuda a garantir que as pesquisas correspondam aos mapas.

Se seu aplicativo estiver rastreando ou roteando ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você só poderá usá-lo HERE como seu provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).

- d. Escolha a opção Armazenamento de dados. Neste tutorial, os resultados não são armazenados, então você pode selecionar Não, somente para uso único.
- e. Aceite os Termos e condições do Amazon Location e, em seguida, selecione Criar índice de local.
- f. Anote o ARN que é mostrado para seu novo recurso de índice de lugares. Você o usará para criar a autenticação correta na próxima seção deste tutorial.

## Configurar a autenticação para seu aplicativo


O aplicativo que você cria neste tutorial tem uso anônimo, o que significa que seus usuários não precisam fazer login AWS para usar o aplicativo. No entanto, por padrão, o Amazon Location Service APIs exige autenticação para ser usado. Você pode usar o Amazon Cognito ou API as chaves para fornecer autenticação e autorização para usuários anônimos. Neste tutorial, você criará API chaves para uso no aplicativo de amostra.

 Note

Para obter mais informações sobre o uso de API chaves ou do Amazon Cognito com o Amazon Location Service, consulte. [Conceder acesso ao Amazon Location Service](#)


Para configurar a autenticação para seu aplicativo

1. Acesse o [console de localização da Amazon](#) e escolha APIs chaves no menu à esquerda.
2. Escolha Criar chave API.

 Important

A API chave que você criar deve estar na mesma Conta da AWS AWS região dos recursos do Amazon Location Service que você criou na seção anterior.

3. Na página Criar API chave, preencha as informações a seguir.
  - Nome — Um nome para sua API chave, como MyWebAppKey.
  - Recursos: selecione o mapa e os recursos do índice de local do Amazon Location que criou na seção anterior. Você pode adicionar mais de um recurso escolhendo Adicionar recurso. Isso permitirá que a API chave seja usada com esses recursos.
  - Ações — Especifique as ações que você deseja autorizar com essa API chave. Você deve selecionar pelo menos geo: GetMap \* e geo: SearchPlaceIndexForPosition para que o tutorial funcione conforme o esperado.
  - Opcionalmente, você pode adicionar uma descrição, prazo de validade ou tags à sua API chave. Você também pode adicionar um indicador (como \*.example.com) para limitar a chave a ser usada somente em um domínio específico. Isso significa que o tutorial só funcionará nesse domínio.

 Note

É recomendável que você proteja o uso da API chave definindo um prazo de expiração ou um referenciador, se não ambos.

4. Escolha Criar API chave para criar a API chave.



- Escolha Mostrar API chave e copie o valor da chave para uso posterior no tutorial. Estará no formulário `v1.public.a1b2c3d4....`.

 Important

Você precisará dessa chave ao escrever o código para seu aplicativo posteriormente neste tutorial.

## Criando o HTML para seu aplicativo

Neste tutorial, você criará uma HTML página estática que incorpora um mapa e permite que o usuário encontre o que está em um local no mapa. O aplicativo consistirá em três arquivos: um HTML arquivo e um CSS arquivo para a página da web e um arquivo JavaScript (.js) para o código que cria o mapa e responde às interações e eventos do mapa do usuário.

Primeiro, vamos criar a CSS estrutura HTML e que será usada para o aplicativo. Essa será uma página simples com um `<div>` elemento para armazenar o contêiner do mapa e um `<pre>` elemento para mostrar as JSON respostas às suas consultas.

Para criar o HTML para seu aplicativo de início rápido

- Crie um novo arquivo chamado `quickstart.html`.
- Edite o arquivo no editor de texto ou no ambiente de sua escolha. Adicione o seguinte HTML ao arquivo.

```
<!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>
```

Isso HTML tem um ponteiro para o CSS arquivo que você criará na próxima etapa, alguns elementos de espaço reservado para o aplicativo e um texto explicativo.

Há dois elementos de espaço reservado que você usará posteriormente neste tutorial. O primeiro é o elemento `<div id="map">`, que manterá o controle do mapa. O segundo é o elemento `<pre id="response">`, que mostrará os resultados da pesquisa no mapa.

### 3. Salve o arquivo.

Agora adicione o CSS para a página da web. Isso definirá o estilo do texto e dos elementos do espaço reservado para o aplicativo.

Para criar o CSS para seu aplicativo de início rápido

1. Crie um novo arquivo chamado `main.css`, na mesma pasta do arquivo `quickstart.html` criado no procedimento anterior.
2. Edite o arquivo em qualquer editor que quiser usar. Adicione o seguinte texto ao arquivo.

```
* {
  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;
}

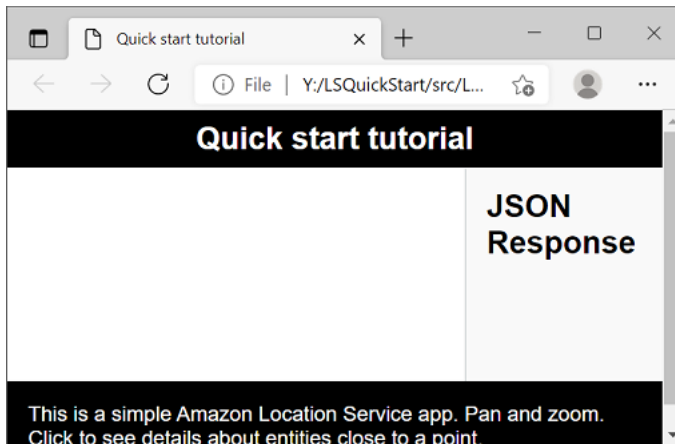
pre {
white-space: pre-wrap;
font-family: monospace;
color: #16191f;
}

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

```
}
```

Isso configura o mapa para preencher o espaço não usado por mais nada, define que a área de nossas respostas ocupe 30% da largura do aplicativo e define cores e estilos para o título e o texto explicativo.

3. Salve o arquivo.
4. Agora você pode visualizar o arquivo `quickstart.html` em um navegador para ver o layout do aplicativo.



Em seguida, você adicionará o controle de mapas ao aplicativo.

## Adicionar um mapa interativo ao seu aplicativo

Agora que você tem uma estrutura e um espaço reservado para `div`, pode adicionar o controle de mapa ao seu aplicativo. Este tutorial usa o [MapLibre GL JS](#) como controle de mapa, obtendo dados do Amazon Location Service. Você também usará o [JavaScript Auxiliar de autenticação](#) para facilitar a assinatura de chamadas para o local da Amazon APIs com sua API chave.

Para adicionar um mapa interativo ao seu aplicativo

1. Abra o arquivo `quickstart.html` que você criou na seção anterior.
2. Adicione referências às bibliotecas necessárias e ao arquivo de script que você criará. As alterações que você precisa fazer são mostradas em **green**.

```
<!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>
```

Isso adicionará as seguintes dependências ao seu aplicativo:

- MapLibre VÁ JS. Essa biblioteca e folha de estilo incluem um controle de mapa que exibe blocos de mapa e inclui interatividade, como panorâmica e zoom. O controle também permite extensões, como desenhar seus próprios atributos no mapa.
- Cliente do Amazon Location. Isso fornece interfaces para a funcionalidade do Amazon Location necessária para obter dados do mapa e pesquisar lugares no mapa. O cliente Amazon Location é baseado no AWS SDK for JavaScript v3.
- Assistente de autenticação do Amazon Location. Isso fornece funções úteis para autenticar o Amazon Location Service com API chaves ou o Amazon Cognito.

Essa etapa também adiciona uma referência a `main.js`, que você criará a seguir.

3. Salve o arquivo `quickstart.html`.
4. Crie um novo arquivo chamado `main.js` na mesma pasta que o seu HTML e CSS os arquivos e abra-o para edição.

5. Adicione o seguinte script ao seu arquivo. O texto em *red* deve ser substituído pelo valor da API chave, pelo nome do recurso do mapa e pelo nome do recurso do local que você criou anteriormente, bem como pelo identificador da região (comous-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 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() {
  // Initialize map and Amazon Location SDK client:
  const map = await initializeMap();
}

main();
```

Esse código configura os recursos de localização da Amazon, depois configura e inicializa um controle de mapa MapLibre GL JS e o coloca em seu `<div>` elemento com o id. map

É importante entender a função `initializeMap()`. Ele cria um novo controle de MapLibre mapa (chamado `mlglMap` localmente, mas chamado `map` no resto do código) que é usado para renderizar o mapa em seu aplicativo.

```
// Initialize the map
```

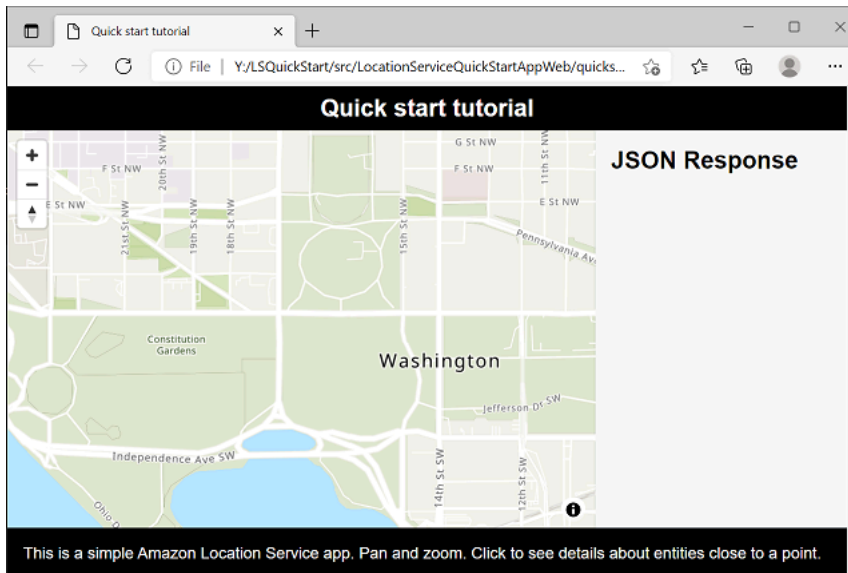
```
const mlg1Map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}`, // Defines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

Quando você cria um novo controle de MapLibre mapa, os parâmetros que você passa indicam o estado inicial do controle de mapa. Aqui, definimos os seguintes parâmetros:

- HTMLcontainer, que usa o elemento map div em nossoHTML.
  - O centro inicial do mapa até um ponto em Washington, DC.
  - O nível de zoom até 16 (ampliado para o nível de um bairro ou quarteirão).
  - O estilo a ser usado para o mapa, que permite MapLibre usar URL para obter os blocos do mapa e outras informações para renderizar o mapa. Observe que isso URL inclui sua API chave para autenticação.
6. Salve seu JavaScript arquivo e abra-o com um navegador. Agora você tem um mapa na sua página, onde você pode usar ações de panorâmica e zoom.

#### Note

Você pode usar este aplicativo para ver como o controle do MapLibre mapa se comporta. Você pode tentar usar Ctrl ou Shift ao usar uma operação de arrastar, para ver outras formas de interagir com o mapa. Toda essa funcionalidade é personalizável.



Seu aplicativo está quase completo. Na próxima seção, você escolherá um local no mapa e mostrará o endereço do local escolhido. Você também mostrará o resultado JSON na página, para ver os resultados completos.

## Adicionar pesquisas ao seu aplicativo

A última etapa do seu aplicativo é adicionar uma pesquisa no mapa. Nesse caso, você adicionará uma pesquisa de geocodificação reversa, na qual você encontrará os itens em um local.

### Note

O Amazon Location Service também oferece a capacidade de pesquisar por nome ou endereço para encontrar a localização dos lugares no mapa.

Para adicionar a funcionalidade de pesquisa ao seu aplicativo

1. Abra o arquivo `main.js` que você criou na seção anterior.
2. Modifique a função `main`, conforme mostrado. As alterações que você precisa fazer são mostradas em **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
});

// 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.");
  }
});
```

```
    }  
  });  
}
```

Esse código começa configurando o assistente de autenticação de localização da Amazon para usar sua API chave.

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

Em seguida, ele usa esse assistente de autenticação e a região que você está usando para criar um novo cliente do Amazon Location.

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

Em seguida, o código responde ao usuário escolher um ponto no controle do mapa. Ele faz isso capturando um evento MapLibre fornecido para `click`.

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

O MapLibre `click` evento fornece parâmetros que incluem a latitude e a longitude que o usuário escolheu (`e.lngLat`). Dentro do evento `click`, o código cria o `searchPlaceIndexForPositionCommand` para encontrar as entidades na latitude e longitude fornecidas.

```
// 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);
  ...
});
```

Aqui, `IndexName` é o nome do recurso Índice de local que você criou anteriormente, `Position` é latitude e a longitude a serem pesquisadas, `Language` é o idioma preferido para os resultados e `MaxResults` diz ao Amazon Location que retorne apenas no máximo cinco resultados.

O código restante verifica se há um erro e, em seguida, exibe os resultados da pesquisa no elemento `<pre>` chamado `response` e mostra o resultado principal em uma caixa de alerta.

3. (Opcional) Se você salvar e abrir o arquivo `quickstart.html` em um navegador agora, a escolha de um local no mapa mostrará o nome ou endereço do local escolhido.
4. A etapa final do aplicativo é usar a MapLibre funcionalidade para adicionar um marcador no local selecionado pelo usuário. Modifique a função `main` da seguinte forma. As alterações que você precisa fazer são mostradas em **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",
};

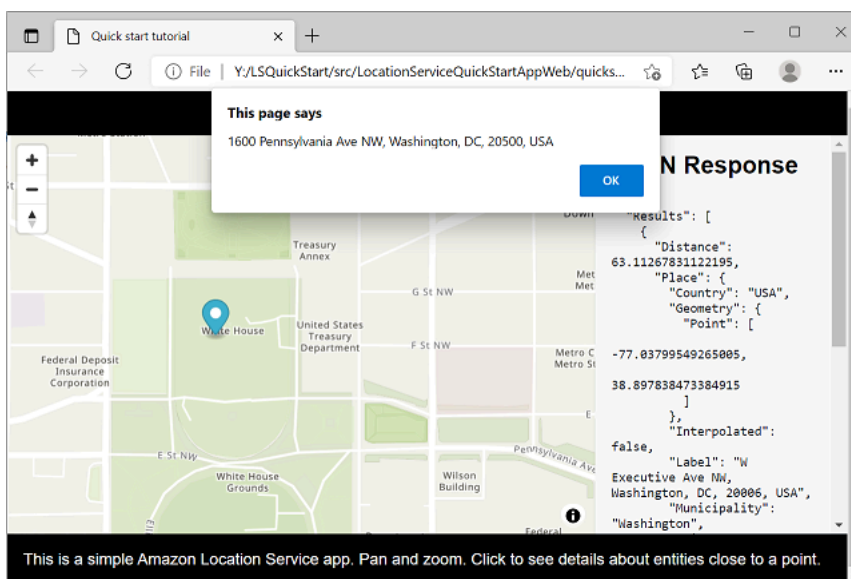
// Set up command to search for results around clicked point
const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

...

```

Esse código declara uma variável `marker`, que é preenchida toda vez que o usuário seleciona um local, mostrando onde ele selecionou. O marcador é renderizado automaticamente pelo controle do mapa, uma vez adicionado ao mapa com `.addTo(map)`; . O código também verifica se há um marcador anterior e o remove, de forma que haja apenas 1 marcador na tela por vez.

5. Salve o arquivo `main.js` e abra o arquivo `quickstart.html` em um navegador. Você pode mover e ampliar o mapa, como antes, mas agora, se escolher um local, verá detalhes sobre o local escolhido.



Seu aplicativo de início rápido está completo. Este tutorial mostrou como criar um HTML aplicativo estático que:

- Cria um mapa com o qual os usuários podem interagir.
- Manipula um evento de mapa (`click`).
- Chama um Amazon Location Service API, especificamente para pesquisar o mapa em um local, usando `searchPlaceIndexForPosition`.
- Usa o controle de MapLibre mapa para adicionar um marcador.

## Visualizar o aplicativo final

O código-fonte final desse aplicativo está incluído nesta seção. Você também pode encontrar o projeto final [em GitHub](#).

[Você também pode encontrar uma versão do aplicativo que usa o Amazon Cognito em vez de API teclas ativadas. GitHub](#)

### Overview

Selecione cada guia para ver o código-fonte final dos arquivos neste tutorial de início rápido.

Os arquivos são:

- `quickstart.html` — a estrutura do seu aplicativo, incluindo os suportes de HTML elementos para o mapa e os resultados da pesquisa.
- `main.css`: a folha de estilo do aplicativo.
- `main.js`: o script do seu aplicativo que autentica o usuário, cria o mapa e pesquisa um evento `click`.

### quickstart.html

A HTML estrutura para o aplicativo de início rápido.

```
<!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

A folha de estilo do aplicativo de início rápido.

```
* {
  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

O código do aplicativo de início rápido. O texto em *red* devem ser substituídos pelos nomes de objetos de localização da Amazon apropriados.

```
// 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 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
  });

  // Add navigation control to the top left of the map
  mlg1Map.addControl(new maplibregl.NavigationControl(), "top-left");

  return mlg1Map;
}

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();
```

## Próximas etapas

Você concluiu o tutorial de início rápido e deve ter uma ideia de como o Amazon Location Service é usado para criar aplicativos. Para aproveitar melhor o Amazon Location, você pode conferir os seguintes recursos:

- Saiba mais sobre os [conceitos do Amazon Location Service](#)
- Obtenha mais informações sobre [como usar os atributos e funcionalidades do Amazon Location](#)
- Veja como expandir essa amostra e criar aplicativos mais complexos analisando [exemplos de código que usam o Amazon Location](#)

## Criar um aplicativo Android

Nesta seção, você criará um aplicativo Android com um mapa, a capacidade de pesquisar em um local e rastrear em primeiro plano. Primeiro, você criará seus recursos de localização da Amazon, uma identidade do Amazon Cognito e uma chave de API para seu aplicativo.

### Tópicos

- [Criar recursos do Amazon Location para seu aplicativo](#)
- [Configurar a autenticação para seu aplicativo](#)
- [Criar o aplicativo base para Android](#)
- [Adicionar um mapa interativo ao seu aplicativo](#)
- [Adicionando a pesquisa de geocodificação reversa ao seu aplicativo](#)
- [Adicionar rastreamento ao seu aplicativo](#)
- [Próximas etapas](#)


## Criar recursos do Amazon Location para seu aplicativo

Se você ainda não os tiver, você deverá criar os recursos do Amazon Location que seu aplicativo utilizará. Aqui, você cria um recurso de mapa para exibir mapas em seu aplicativo, um índice de locais para pesquisar localizações no mapa e um rastreador para rastrear um objeto no mapa.

Para adicionar recursos de localização ao seu aplicativo

1. Escolha o estilo de mapa que deseja utilizar.

- a. No console do Amazon Location, na página [Mapas](#), selecione Criar mapa para visualizar estilos de mapa.
- b. Adicione um Nome e uma Descrição para o novo recurso de mapa. Anote o nome que usa para o recurso do mapa. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
- c. Recomendamos que você escolha o estilo de mapa HERE para seu mapa.

 Note

Escolher um estilo de mapa também seleciona qual provedor de dados de mapa você usará. Se o aplicativo estiver rastreando ou roteando os ativos que você utiliza em sua empresa, como veículos de entrega ou funcionários, você só poderá utilizar o HERE como provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).

- d. Aceite os Termos e condições do Amazon Location e selecione Criar mapa. Você pode interagir com o mapa que escolheu: ampliar, reduzir o zoom ou fazer uma panorâmica em qualquer direção.
  - e. Anote o Nome de recurso da Amazon (ARN) que for exibido do seu novo recurso do mapa. Você o usará para criar a autenticação correta posteriormente neste tutorial.
2. Escolha o índice de local que deseja usar.
- a. No console do Amazon Location, na página de [Índices de local](#), selecione Criar índice de local.
  - b. Adicione um Nome e uma Descrição para o novo recurso de índice de local. Anote os nomes que você utilizar para o recurso de índice de local. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
  - c. Selecione um provedor de dados.

 Note

Na maioria dos casos, escolha o provedor de dados que corresponda ao provedor de mapa que você já escolheu. Isso ajuda a garantir que as pesquisas correspondam aos mapas.

Se o aplicativo estiver rastreando ou roteando os ativos que você utiliza em sua empresa, como veículos de entrega ou funcionários, você só poderá utilizar o HERE

como provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).

- d. Escolha a opção Armazenamento de dados. Neste tutorial, os resultados não são armazenados, então você pode selecionar Não, somente para uso único.
  - e. Aceite os Termos e condições do Amazon Location e, em seguida, selecione Criar índice de local.
  - f. Anote o ARN exibido para seu novo recurso de índice de local. Você o usará para criar a autenticação correta na próxima seção deste tutorial.
3. Para criar um rastreador usando o console de localização da Amazon.
- a. Abra o [console do Amazon Location Service](#).
  - b. No painel de navegação à esquerda, selecione Rastreadores.
  - c. Selecione Criar rastreador.
  - d. Preencha todos os campos obrigatórios.
  - e. Em Filtragem de posição, recomendamos que você use a configuração padrão: TimeBased
  - f. Escolha Criar rastreador para finalizar.

## Configurar a autenticação para seu aplicativo

O aplicativo que você cria neste tutorial tem uso anônimo, o que significa que seus usuários não precisam fazer login AWS para usar o aplicativo. No entanto, as APIs do Amazon Location Service exigem autenticação para utilização. Você pode utilizar chaves de API ou o Amazon Cognito para fornecer autenticação e autorização para usuários anônimos. Este tutorial usará o Amazon Cognito e as chaves de API para autenticar seu aplicativo.

### Note

Para obter mais informações sobre o uso do Amazon Cognito ou das chaves de API com o Amazon Location Service, consulte [Conceder acesso ao Amazon Location Service](#).

Os tutoriais a seguir mostram como configurar a autenticação para o mapa, o índice de lugares e o rastreador que você criou, além de configurar permissões para a localização da Amazon.

## Configurar a autenticação

1. Navegue até o [console de localização da Amazon](#) e selecione as chaves de API no menu à esquerda.
2. Clique em “Criar chave de API”. Lembre-se de que a chave de API deve estar na mesma AWS conta e região dos recursos do Amazon Location Service criados anteriormente.
3. Preencha os detalhes necessários na página “Criar chave de API”:
  - Nome: forneça um nome para sua chave de API, como `MyAppKey`.
  - Recursos: Escolha os recursos de índice Amazon Location Service Map and Place criados anteriormente. Você pode adicionar vários recursos selecionando “Adicionar recurso”. Isso permite que a chave da API seja usada com recursos específicos.
  - Ações: especifique ações autorizadas para essa chave de API. No mínimo, selecione `geo:GetMap` e garanta `geo:SearchPlaceIndexForPosition` que o tutorial funcione conforme pretendido.
  - Opcionalmente, você pode adicionar uma descrição, tempo de expiração, tags ou um referenciador, por exemplo, `https://www.example.com` para limitar o uso da chave a um domínio específico, permitindo que o tutorial funcione somente dentro desse domínio.
4. Clique em Criar chave de API para gerar a chave de API.
5. Selecione Mostrar chave de API e copie o valor da chave, `v1.public.a1b2c3d4` por exemplo, para uso posterior no tutorial.

## Crie uma política do IAM para rastreamento

1. Faça login no console do IAM em <https://console.aws.amazon.com/iam/> com seu usuário que tenha permissões de administrador.
2. No painel de navegação, escolha Políticas.
3. No painel de conteúdo, escolha Criar política.
4. Escolha a opção JSON e, em seguida, copie e cole essa política JSON na caixa de texto JSON.

```
{
  "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}"
    ]
  }
]
```

Este é um exemplo de política para Rastreamento. Para usar o exemplo para sua própria política, substitua os `Region` `TrackerName` espaços reservados `Account`, e.

#### Note

Embora os grupos de identidade não autenticados sejam destinados à exposição em sites da Internet não seguros, observe que eles serão trocados por credenciais padrão e com limite de tempo da AWS.

É importante definir corretamente o escopo dos perfis do IAM associados aos bancos de identidades não autenticados. Para obter mais informações sobre como usar e definir o escopo adequado das políticas no Amazon Cognito com o Amazon Location Service, [consulte Conceder acesso ao Amazon Location Service](#).

5. Na página Revisar e criar, forneça um nome para o campo nome da política. Revise as permissões concedidas pela sua política e escolha Criar política para salvar seu trabalho.

A nova política aparece na lista de políticas gerenciadas e está pronta para ser anexada.

Configure a autenticação para seu rastreamento

1. Configure a autenticação para seu aplicativo de mapas no console do [Amazon Cognito](#).
2. Abra a página de grupos de identidades.

**Note**

O pool que você cria deve estar na mesma AWS conta e AWS região dos recursos do Amazon Location Service que você criou na seção anterior.

3. Escolha Criar pool de identidades.
4. Começando com a etapa Configurar confiança do pool de identidades. Para autenticação de acesso do usuário, selecione Acesso de convidado e pressione Avançar.
5. Na página Configurar permissões, selecione Usar uma função do IAM existente e insira o nome da função do IAM que você criou na etapa anterior. Quando estiver pronto, pressione Avançar para passar para a próxima etapa.
6. Na página Configurar propriedades, forneça um nome para seu grupo de identidades. Em seguida, pressione Avançar.
7. Na página Revisar e criar, revise todas as informações presentes e pressione Criar pool de identidades.
8. Abra a página Conjuntos de identidades e selecione o grupo de identidades que você acabou de criar. Em seguida, copie ou anote o IdentityPoolId que você usará posteriormente no script do seu navegador.

## Criar o aplicativo base para Android

Neste tutorial, você criará um aplicativo Android que incorpora um mapa e permite que o usuário encontre o que está em um local no mapa.

Primeiro, crie um aplicativo Kotlin vazio usando o novo assistente de projeto do Android Studio.

Para criar um aplicativo vazio (AndroidStudio)

1. Começar AndroidStudio. Abra o menu e escolha Arquivo, Novo, Novo projeto.
2. Na guia Telefone e tablet, selecione Atividade vazia e, depois, selecione Avançar.
3. Escolha um Nome, Nome do pacote e Salvar local para o seu aplicativo.
4. Na lista suspensa Idioma, selecione Kotlin.
5. Selecione Concluir para criar seu aplicativo vazio.

## Adicionar um mapa interativo ao seu aplicativo

Agora que você criou um aplicativo básico, você pode adicionar controle de mapa ao seu aplicativo. Este tutorial usa chaves de API para gerenciar a visualização do mapa. O controle do mapa em si faz parte da [biblioteca MapLibre nativa](#), com a chave da API e MapLibre, e os dados do mapa vêm da Amazon Location.

Para adicionar um mapa ao seu aplicativo, você precisará executar as seguintes ações:

- Adicione a MapLibre dependência ao seu projeto.
- Configure o código de visualização do mapa com compose.
- Escreva um código para mostrar o mapa.

Use o procedimento a seguir para adicionar o mapa ao seu aplicativo:

1. Adicione a MapLibre dependência ao seu projeto
  - a. Em AndroidStudio, selecione o menu Exibir e escolha Janelas de ferramentas, Projeto. Isso abrirá a janela Projeto, que dará acesso a todos os arquivos do seu projeto.
  - b. Na janela Projeto, abra o gradle e, em seguida, abra o `libs.versions.toml` arquivo na visualização em árvore. Isso abrirá o arquivo `libs.versions.toml` para edição. Agora, adicione a versão abaixo e os dados das bibliotecas no `libs.versions.toml` arquivo.

```
[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. Depois de terminar de editar o `libs.versions.toml` arquivo, é AndroidStudio necessário sincronizar novamente o projeto. Na parte superior da janela de `libs.versions.toml`



edição, AndroidStudio solicita que você sincronize. Selecione “Sincronizar agora” para sincronizar seu projeto antes de continuar.

- d. Na janela Projeto, abra os scripts do Gradle na visualização em árvore e selecione o `build.gradle` arquivo para o módulo do seu aplicativo. Isso abrirá o arquivo `build.gradle` para edição.
- e. Na parte inferior do arquivo, na seção de dependências, adicione a seguinte dependência.

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

- f. Depois de terminar de adicionar as dependências do Gradle, o Android Studio precisa sincronizar novamente o projeto. Na parte superior da janela de edição `build.gradle`, Android Studio, selecione Sincronizar agora para sincronizar seu projeto antes de continuar.

2. Agora você configurará o código de visualização do mapa com compose. Use as seguintes etapas:

- a. Na janela Projeto, abra App, Java, o *nome do seu pacote* na visualização em árvore e vá para a pasta `ui`. Dentro da pasta `ui`, crie um diretório de visualização.
- b. O diretório de visualização interna cria um `MapLoadScreen.kt` arquivo.
- c. Adicione o seguinte código ao arquivo `MapLoadScreen.kt`.

```
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. Escreva um código para mostrar o mapa.

#### a. Adicione o seguinte código ao arquivo MainActivity.kt.

```
// ...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. Salve o arquivo `MainActivity.kt`. Agora você pode criar o aplicativo. Para executá-lo, talvez seja necessário configurar um dispositivo para emulá-lo ou usar o aplicativo em seu dispositivo. Android Studio Use este aplicativo para ver como o controle do mapa se comporta. Você pode fazer uma panorâmica arrastando-a no mapa e apertando-a para ampliar.

Na próxima seção, você adicionará um marcador no mapa e mostrará o endereço do local onde o marcador está à medida que você move o mapa.

## Adicionando a pesquisa de geocodificação reversa ao seu aplicativo

Agora você adicionará a pesquisa de geocodificação reversa ao seu aplicativo, onde encontrará os itens em um local. Para simplificar o uso de um aplicativo Android, pesquisaremos no centro da tela. Para encontrar um novo local, mova o mapa para onde você deseja pesquisar. Colocaremos um marcador no centro do mapa para mostrar onde estamos pesquisando.

Adicionar uma pesquisa de geocodificação reversa consistirá em duas partes.

- Adicione um marcador no centro da tela para mostrar ao usuário onde estamos pesquisando.
- Adicione uma caixa de texto para ver os resultados, depois pesquise o que está no local do marcador e mostre-o na caixa de texto.

Para adicionar um marcador ao seu aplicativo

1. Salve esta imagem em seu projeto na `app/res/drawable` pasta como `red_marker.png` (você também pode acessar a imagem em [GitHub](#). Como alternativa, você pode criar sua imagem. Você também pode usar um arquivo.png com transparência para as partes que você não quer que sejam mostradas.
2. Adicione o código a seguir ao seu `MapLoadScreen` arquivo.kt.

```
// ...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. Crie e execute seu aplicativo para visualizar a funcionalidade.

Agora, seu aplicativo tem um marcador na tela. Nesse caso, ele é uma imagem estática, que não se move. Ele é usado para mostrar o centro da visualização do mapa, que é onde vamos pesquisar. No procedimento a seguir, adicionaremos a pesquisa nesse local.

Para adicionar a pesquisa de geocodificação reversa em um local ao seu aplicativo

1. Na janela Projeto, abra `gradle to libs.versions.toml` file na visualização em árvore. Isso abrirá o arquivo `libs.versions.toml` para edição. Agora, adicione a versão abaixo e os dados das bibliotecas no `libs.versions.toml` arquivo.

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

[libraries]
...
com-squareup-okhttp3 = { group = "com.squareup.okhttp3", name = "okhttp",
version.ref = "okhttp" }

[plugins]
...
```

2. Depois de terminar de editar o `libs.versions.toml` arquivo, é AndroidStudio necessário sincronizar novamente o projeto. Na parte superior da janela de `libs.versions.toml` edição, AndroidStudio solicita que você sincronize. Selecione “Sincronizar agora” para sincronizar seu projeto antes de continuar.
3. Na janela Projeto, abra os scripts do Gradle na visualização em árvore e selecione o `build.gradle` arquivo para o módulo do seu aplicativo. Isso abrirá o arquivo `build.gradle` para edição.
4. Na parte inferior do arquivo, na seção de dependências, adicione a seguinte dependência.

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

5. Depois de terminar de editar as dependências do Gradle, é AndroidStudio necessário sincronizar novamente o projeto. Na parte superior da janela de `build.gradle` edição, AndroidStudio solicita que você sincronize. Selecione SyncNow para sincronizar seu projeto antes de continuar.
6. Agora, na visualização em árvore, adicione os dados ao diretório de solicitações e crie a classe `ReverseGeocodeRequest.kt` de dados. Adicione o código a seguir à classe.

```
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. Agora, na visualização em árvore, adicione os dados ao diretório de resposta e crie a classe `ReverseGeocodeResponse.kt` de dados. Adicione o código a seguir dentro dele.

```
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. Agora, na janela Projeto, abra App, Java, o *nome do seu pacote* na visualização em árvore e vá para a pasta ui, dentro da pasta ui create viewModel.
9. Dentro do diretório ViewModel, crie um MainViewModel.kt arquivo.
10. Adicione o seguinte código ao arquivo MainViewModel.kt.

```
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) {
                    label = responseLabel
                    isLabelAdded = true
                }
            }
        }
    })
}
```



11. Se ainda não estiver aberto, abra o arquivo `MapLoadScreen.kt`, como no procedimento anterior. Adicione o seguinte código. Isso criará uma visualização de texto composto, na qual você verá os resultados da pesquisa com geocodificação reversa no local.

```
// ...other imports
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Text
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.testTag
import androidx.compose.ui.semantics.contentDescription
import androidx.compose.ui.semantics.semantics
import androidx.compose.ui.unit.sp
import com.amazon.testmapapp.ui.viewModel.MainViewModel

@Composable
fun MapLoadScreen(
    mapReadyCallback: OnMapReadyCallback,
    mainViewModel: MainViewModel,
) {
    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))
        }
    }
}

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

12. No aplicativo, em java, na pasta do nome do pacote em AndroidStudio, abra o `MainActivity.kt` arquivo. Modifique o código conforme mostrado.

```
// ...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)
        setContent {
            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"
                ),
            ) {
            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
            )
        }
    }
}
}
```

Esse código funciona com a visualização do mapa. A posição da câmera virtual define a visualização do mapa em MapLibre. Mover o mapa pode ser considerado como mover aquela câmera virtual.

- ViewModel tem uma variável de rótulo: essa variável define dados na visualização de texto composto.
- onMapReady: Esta função é atualizada para registrar dois novos eventos.
- O onCameraMove evento acontece sempre que o usuário está movendo o mapa. Em geral, ao mover o mapa, queremos ocultar a pesquisa até que o usuário termine de mover o mapa.

- O `onCameraIdle` evento ocorre quando o usuário pausa a movimentação do mapa. Esse evento chama nossa função de geocodificação reversa para pesquisar no centro do mapa.
- `reverseGeocode(latLng: LatLng, apiKey: String)`: Essa função, chamada no evento `onCameraIdle`, pesquisa no centro do mapa por uma localização e atualiza o rótulo para mostrar os resultados. Ele usa o alvo da câmera, que define o centro do mapa (para onde a câmera está olhando).

13. Salve seus arquivos, crie e execute seu aplicativo para ver se ele funciona.

Seu aplicativo de início rápido com funcionalidade de pesquisa está completo.

## Adicionar rastreamento ao seu aplicativo

Para adicionar rastreamento ao seu aplicativo de amostra, siga estas etapas:

1. Adicione dependências do SDK de rastreamento e autenticação ao seu projeto.
  2. Inclua entradas de permissão e serviço em seu `AndroidManifest` arquivo.xml.
  3. Configure o código do botão de rastreamento iniciar/parar com `compose`.
  4. Adicione código para criar um `LocationTracker` objeto e iniciar e interromper o rastreamento.
  5. Crie uma rota de teste com o Android Emulator.
1. Adicione dependências do SDK de rastreamento e autenticação ao seu projeto.
    - a. Na janela Projeto, abra o `gradle` e, em seguida, abra o `libs.versions.toml` arquivo na visualização em árvore. Isso abrirá o arquivo `libs.versions.toml` para edição. Agora, adicione a versão abaixo e os dados das bibliotecas no `libs.versions.toml` arquivo.

```
[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. Depois de terminar de editar o `libs.versions.toml` arquivo, é AndroidStudio necessário sincronizar novamente o projeto. Na parte superior da janela de `libs.versions.toml` edição, AndroidStudio solicita que você sincronize. Selecione “Sincronizar agora” para sincronizar seu projeto antes de continuar.
- c. Na janela Projeto, abra os scripts do Gradle na visualização em árvore e selecione o `build.gradle` arquivo para o módulo do seu aplicativo. Isso abrirá o arquivo `build.gradle` para edição.
- d. Na parte inferior do arquivo, na seção de dependências, adicione a seguinte dependência.

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

- e. Depois de terminar de editar as dependências do Gradle, é AndroidStudio necessário sincronizar novamente o projeto. Na parte superior da janela de edição `build.gradle`, AndroidStudio solicita que você sincronize. Selecione SyncNow para sincronizar seu projeto antes de continuar.
2. Inclua entradas de permissão e serviço no seu `AndroidManifest` arquivo.xml.
    - Para incluir as entradas corretas de permissão e serviço no seu `AndroidManifest.xml` file, atualize o arquivo com o seguinte código:

```
<?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. Configure o código do botão de rastreamento iniciar/parar com compose.
  - a. Adicione duas imagens de Play e Pause in res abaixo do drawable chamadas ic\_pause e ic\_play. Você também pode acessar a imagem em [GitHub](#).
  - b. Se ainda não estiver aberto, abra o arquivo MapLoadScreen.kt, como no procedimento anterior. Adicione o seguinte código. Isso criará uma visualização do botão de composição, na qual podemos clicar nela para iniciar e interromper o rastreamento.

```
// ...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. Adicione código para criar um `LocationTracker` objeto e iniciar e interromper o rastreamento.
  - a. Adicione o código a seguir dentro do `MainViewModel.kt` arquivo.

```

...
var isLocationTrackingForegroundActive: Boolean by mutableStateOf(false)

```

```
var locationTracker: LocationTracker? = null
```

b. Adicione o seguinte código ao arquivo MainActivity.kt.

```
// ...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 {
                        override fun
                        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
            )
        }
    }
}
}
```

O código acima mostra como criar um `LocationTracker` objeto com `AuthHelper` e como iniciar e interromper o rastreamento com `LocationTracker`.

- `authenticateUser()`: Esse método cria `LocationTracker` objetos `AuthHelper` e objetos.

- `onStartStopTrackingClick`: esse retorno de chamada é acionado quando o usuário clica no botão iniciar/parar o rastreamento, que iniciará ou interromperá o rastreamento com o SDK de rastreamento.
5. Crie uma rota de teste com o Android Emulator.
    - a. Abra o Emulador iniciando o AVD usando o Android Studio.
    - b. Abra os controles estendidos clicando no ícone Mais (três pontos) na barra de ferramentas do emulador.
    - c. Abra o Local selecionando Local na barra lateral.
    - d. Crie uma rota com dados GPX ou clicando no mapa e escolhendo os dados de origem e destino.
    - e. Inicie a simulação clicando em PLAY ROUTE para começar a simular a rota GPS.
    - f. Teste o aplicativo executando seu aplicativo e observando como ele lida com a rota simulada.

Esta é a demonstração completa do aplicativo Android Quick Start.

## Próximas etapas

O código-fonte desse aplicativo está disponível em [GitHub](#).

Para aproveitar melhor o Amazon Location, você pode conferir os seguintes recursos:

- Saiba mais sobre os [conceitos do Amazon Location Service](#)
- Obtenha mais informações sobre [como usar os atributos e funcionalidades do Amazon Location](#)
- Veja como expandir essa amostra e criar aplicativos mais complexos analisando [exemplos de código que usam o Amazon Location](#)

## Criação de um aplicativo para iOS

Nesta seção, você criará um aplicativo iOS com a capacidade de pesquisar em um local e rastrear em primeiro plano. Primeiro, você criará seus recursos de localização da Amazon e uma identidade do Amazon Cognito para seu aplicativo.

### Tópicos

- [Criar recursos do Amazon Location para seu aplicativo](#)

- [Configurar a autenticação para seu aplicativo](#)
- [Criando o aplicativo iOS básico](#)
- [Configurando o código inicial](#)
- [Adicionar um mapa interativo ao seu aplicativo](#)
- [Adicionar pesquisas ao seu aplicativo](#)
- [Adicionar rastreamento ao seu aplicativo](#)
- [Próximas etapas](#)

## Criar recursos do Amazon Location para seu aplicativo

Se você ainda não os tiver, você deverá criar os recursos do Amazon Location que seu aplicativo utilizará. Você criará um recurso de mapa para exibir mapas em seu aplicativo, um índice de locais para pesquisar localizações no mapa e um rastreador para rastrear um objeto no mapa.

Para adicionar recursos de localização ao seu aplicativo


1. Escolha o estilo de mapa que deseja utilizar.
  - a. No console do Amazon Location, na página [Mapas](#), selecione Criar mapa para visualizar estilos de mapa.
  - b. Adicione um Nome e uma Descrição para o novo recurso de mapa. Anote o nome que usa para o recurso do mapa. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
  - c. Selecione um mapa.

### Note

Escolher um estilo de mapa também seleciona qual provedor de dados de mapa você usará. Se o aplicativo estiver rastreando ou roteando os ativos que você utiliza em sua empresa, como veículos de entrega ou funcionários, você só poderá utilizar o HERE como provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).

- d. Aceite os Termos e condições do Amazon Location e selecione Criar mapa. Você pode interagir com o mapa que escolheu: ampliar, reduzir o zoom ou fazer uma panorâmica em qualquer direção.

- e. Anote o Nome de recurso da Amazon (ARN) que for exibido do seu novo recurso do mapa. Você o usará para criar a autenticação correta posteriormente neste tutorial.
2. Escolha o índice de local que deseja usar.
    - a. No console do Amazon Location, na página de [Índices de local](#), selecione Criar índice de local.
    - b. Adicione um Nome e uma Descrição para o novo recurso de índice de local. Anote os nomes que você utilizar para o recurso de índice de local. Você precisará deles ao criar seu arquivo de script posteriormente no tutorial.
    - c. Selecione um provedor de dados.

 **Note**

Na maioria dos casos, escolha o provedor de dados que corresponda ao provedor de mapa que você já escolheu. Isso ajuda a garantir que as pesquisas correspondam aos mapas.

Se o aplicativo estiver rastreando ou roteando os ativos que você utiliza em sua empresa, como veículos de entrega ou funcionários, você só poderá utilizar o HERE como provedor de geolocalização. Para obter mais informações, consulte a seção 82 dos [Termos de serviço da AWS](#).
    - d. Escolha a opção Armazenamento de dados. Neste tutorial, os resultados não são armazenados, então você pode selecionar Não, somente para uso único.
    - e. Aceite os Termos e condições do Amazon Location e, em seguida, selecione Criar índice de local.
    - f. Anote o ARN exibido para seu novo recurso de índice de local. Você o usará para criar a autenticação correta na próxima seção deste tutorial.
  3. Para criar um rastreador usando o console de localização da Amazon.
    - a. Abra o [console do Amazon Location Service](#).
    - b. No painel de navegação à esquerda, selecione Rastreadores.
    - c. Selecione Criar rastreador.
    - d. Preencha todos os campos obrigatórios.
    - e. Em Filtragem de posição, recomendamos que você use a configuração padrão: TimeBased
    - f. Escolha Criar rastreador para finalizar.



## Configurar a autenticação para seu aplicativo

O aplicativo que você cria neste tutorial tem uso anônimo, o que significa que seus usuários não precisam fazer login AWS para usar o aplicativo. No entanto, as APIs do Amazon Location Service exigem autenticação para utilização. Você usará o Amazon Cognito para fornecer autenticação e autorização para usuários anônimos. Este tutorial usará o Amazon Cognito para autenticar seu aplicativo.

### Note

Para obter mais informações sobre o uso do Amazon Cognito com o Amazon Location Service, consulte [Conceder acesso ao Amazon Location Service](#)

Os tutoriais a seguir mostram como configurar a autenticação para o mapa, o índice de lugares e o rastreador que você criou, além de configurar permissões para a localização da Amazon.

Crie uma política do IAM para rastreamento

1. Faça login no console do IAM em <https://console.aws.amazon.com/iam/> com seu usuário que tenha permissões de administrador.
2. No painel de navegação, escolha Políticas.
3. No painel de conteúdo, escolha Criar política.
4. Escolha a opção JSON e, em seguida, copie e cole essa política JSON na caixa de texto JSON.

```
{
  "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}"
        ]
    }
}
```

Este é um exemplo de política para Rastreamento. Para usar o exemplo para sua própria política, substitua os TrackerName espaços reservados Region Account IndexName,, MapName e.

#### Note

Embora os grupos de identidade não autenticados sejam destinados à exposição em sites da Internet não seguros, observe que eles serão trocados por credenciais padrão com limite de tempo. AWS

É importante definir corretamente o escopo dos perfis do IAM associados aos bancos de identidades não autenticados. Para obter mais informações sobre como usar e definir o escopo adequado das políticas no Amazon Cognito com o Amazon Location Service, [consulte Conceder acesso ao Amazon Location Service](#).

5. Na página Revisar e criar, forneça um nome para o campo nome da política. Revise as permissões concedidas pela sua política e escolha Criar política para salvar seu trabalho.

A nova política aparece na lista de políticas gerenciadas e está pronta para ser anexada.

Configure a autenticação para seu rastreamento

1. Configure a autenticação para seu aplicativo de mapas no console do [Amazon Cognito](#).
2. Abra a página de grupos de identidades.

#### Note

O pool que você cria deve estar na mesma AWS conta e AWS região dos recursos do Amazon Location Service que você criou na seção anterior.

3. Escolha Criar pool de identidades.

4. Começando com a etapa Configurar confiança do pool de identidades. Para autenticação de acesso do usuário, selecione Acesso de convidado e pressione Avançar.
5. Na página Configurar permissões, selecione Usar uma função do IAM existente e insira o nome da função do IAM que você criou na etapa anterior. Quando estiver pronto, pressione Avançar para passar para a próxima etapa.
6. Na página Configurar propriedades, forneça um nome para seu grupo de identidades. Em seguida, pressione Avançar.
7. Na página Revisar e criar, revise todas as informações presentes e pressione Criar pool de identidades.
8. Abra a página Conjuntos de identidades e selecione o grupo de identidades que você acabou de criar. Em seguida, copie ou anote o IdentityPoolId que você usará posteriormente no script do seu navegador.

## Criando o aplicativo iOS básico

Neste tutorial, você criará um aplicativo iOS que incorpora um mapa e permite que o usuário encontre o que está em um local no mapa.

Primeiro, vamos criar um aplicativo Swift usando o assistente de projeto do Xcode.

Para criar um aplicativo vazio (Xcode)

1. Abra o Xcode e, no menu, escolha Arquivo, Novo, Novo projeto.
2. Na guia iOS, selecione Aplicativo e, em seguida, escolha Avançar.
3. Forneça um nome de produto, um identificador de organização e, no campo Interface, insira SwiftUI. Escolha Avançar para finalizar a seleção.
4. Selecione um local onde você salvará seu projeto e pressione o botão criar para criar o aplicativo vazio.

Depois de criar o aplicativo básico, você precisará instalar os pacotes necessários para o aplicativo de amostra.

Instalando as dependências necessárias

1. No Xcode, clique com o botão direito do mouse no projeto e escolha Adicionar pacotes... . Isso abrirá a janela Pacotes, onde você poderá adicionar pacotes ao seu projeto.

2. Na janela Pacotes, adicione os seguintes pacotes:

- [Para o pacote nativo do Maplibre, use este URL: https://github.com/maplibre/.maplibre-gl-native-distribution](https://github.com/maplibre/.maplibre-gl-native-distribution) Na URL, adicione esses pacotes: `maplibre-gl-native-distribution` `Mapbox` e.
- Para o SDK para iOS de autenticação de localização da Amazon, use este URL: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios>. Na URL, adicione esses pacotes: `amazon-location-mobile-auth-sdk-ios` `AmazonLocationiOSAuthSDK` e.
- Para o SDK para iOS de rastreamento de localização da Amazon, use este URL: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios>. Na URL, adicione esses pacotes: `amazon-location-mobile-tracking-sdk-ios` `AmazonLocationiOSTrackingSDK` e.

## Configurando o código inicial

Ative as permissões de localização em seu aplicativo

1. Abra seu projeto do Xcode.
2. Localize o `Info.plist` arquivo do projeto.
3. Adicione as chaves necessárias para permissões de localização com base nos requisitos do seu aplicativo. Aqui estão as chaves:
  - `NSLocationWhenInUseUsageDescription`: descrição do motivo pelo qual seu aplicativo precisa de acesso à localização quando está em uso.
  - `NSLocationAlwaysAndWhenInUseUsageDescription`: descrição do motivo pelo qual seu aplicativo precisa de acesso contínuo à localização.

Agora você precisará configurar os valores dos recursos em seu aplicativo. Adicione um novo arquivo chamado `Config.xcconfig` e preencha os valores que você criou anteriormente no console da Amazon.

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

1. Na seção do navegador do lado esquerdo, selecione o projeto.
2. Na seção de alvos, selecione seu aplicativo e clique na guia de informações.
3. Adicione propriedades de informações com valores como os abaixo:
4. Adicione o `Config.swift` arquivo com o conteúdo abaixo, que lerá os valores de configuração do arquivo de informações do pacote.

```
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. Crie uma nova pasta com o nome `ViewModel` e adicione um `TrackingViewModel.swift` arquivo dentro dela.

```
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
}
```

```
}
```

## Adicionar um mapa interativo ao seu aplicativo

Agora você adicionará o controle do mapa ao seu aplicativo. Este tutorial usa MapLibre a AWS API para gerenciar a visualização do mapa no aplicativo. O controle do mapa em si faz parte da biblioteca [MapLibre GL Native iOS](#).

1. Adicione o `MapView.swift` arquivo na pasta Views com o seguinte código:

```
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)
    }
}
}
```

2. Adicione `AWSSignatureV4Delegate` o arquivo na `ViewModel` pasta. Esse arquivo é usado para assinar com todas as solicitações `MapView` http para renderizar o mapa:

```
import MapLibre
import AmazonLocationiOSAuthSDK

class AWSSignatureV4Delegate : NSObject, MLNOfflineStorageDelegate {
    private let awsSigner: AWSSigner

    init(credentialsProvider: LocationCredentialsProvider) {
        self.awsSigner = DENY_LIST_ERROR , serviceName: "geo")
        super.init()
    }

    func offlineStorage(_ storage: MLNOfflineStorage, urlForResourceOf kind:
MLNResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            return url
        }
        let signedURL = awsSigner.signURL(url: url, expires: .hours(1))

        return signedURL
    }
}
```

3. Adicione `UserLocationView.swift` o arquivo na pasta `Views`. Isso adiciona um botão que centraliza o mapa na localização do usuário

```
import SwiftUI
```

```

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. Adicione o `TrackingView.swift` arquivo com o seguinte código:

```

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)
            }
        }
    }
}

```

Agora você pode criar o aplicativo. Para executá-lo, talvez seja necessário configurar um dispositivo para emulá-lo no Xcode ou usar o aplicativo em seu dispositivo. Use este aplicativo para ver como o controle do mapa se comporta. Você pode fazer uma panorâmica arrastando no mapa e apertando para ampliar. Por conta própria, você pode alterar a forma como o controle do mapa funciona para personalizá-lo de acordo com as necessidades do seu aplicativo.

## Adicionar pesquisas ao seu aplicativo

Agora você adicionará a pesquisa de geocodificação reversa ao aplicativo, onde encontrará os itens em um local. Para simplificar o uso de um aplicativo iOS, pesquisaremos no centro da tela. Para encontrar um novo local, mova o mapa para onde você deseja pesquisar. Colocaremos um marcador no centro do mapa para mostrar onde estamos pesquisando.

1. Adicione o seguinte código no arquivo `TrackingViewModel.swift` que está relacionado à pesquisa reversa de geocodificação

```
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. Atualize o `TrackingView.swift` arquivo com o seguinte código, que mostrará o endereço da localização centralizada do mapview

```
import SwiftUI

struct TrackingView: View {
    @ObservedObject var trackingViewModel: TrackingViewModel
    var body: some View {
        ZStack(alignment: .bottom) {
            if trackingViewModel.mapSigningIntialised {
                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)
                    }
                }
            }
        }
    }
}
```

## Adicionar rastreamento ao seu aplicativo

A última etapa do seu aplicativo é adicionar a funcionalidade de rastreamento ao seu aplicativo. Nesse caso, você adicionará o rastreamento inicial, interromperá o rastreamento, buscará e exibirá pontos de rastreamento no seu aplicativo.

1. Adicione o `TrackingBottomView.swift` arquivo ao seu projeto. Que tem um botão que inicia e para de rastrear a localização dos usuários e mostra os pontos de rastreamento no mapa.

```
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. Atualize TrackingView.swift o arquivo com o seguinte código

```
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. Adicione o código a seguir no TrackingViewModel.swift arquivo. Essas funções são responsáveis por iniciar e interromper o rastreamento. Também mostrará um alerta de erro se a permissão de localização do usuário for negada.
4. Para implementar o rastreamento em primeiro plano, copie e cole o seguinte exemplo de código:

```
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

Eles `startTracking` solicitarão a permissão de localização do usuário. O aplicativo deve usar as permissões `When In Use` ou `Only Once`. Caso contrário, o aplicativo emitirá um erro de permissão negada.

Para obter e exibir os locais de rastreamento, siga este procedimento:

1. Para obter os locais do dispositivo do usuário, você precisa fornecer a data e a hora de início e término. Uma única chamada retorna no máximo 100 locais de rastreamento, mas se houver mais de 100 locais de rastreamento, ela retornará um valor `nextToken`. Você precisará chamar chamadas subsequentes de `getTrackerDeviceLocation` com `nextToken` para carregar mais pontos de rastreamento para o horário de início e término determinado.

```
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. Agora, substitua o código no `MapView.swift` arquivo pelo seguinte código:

```

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]?)
}

```

Para localizar valores de cadeia de caracteres, use o procedimento a seguir.

1. Crie e adicione um novo arquivo chamado `Localizable.xcstrings`.
2. Clique com o botão direito do mouse no `Localizable.xcstrings` arquivo e abra-o como código-fonte.
3. Substitua seu conteúdo pelo seguinte:

```
{
  "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. Salve seus arquivos, crie e execute seu aplicativo para visualizar a funcionalidade.
5. Permita a permissão de localização e toque no botão de rastreamento. O aplicativo começará a carregar as localizações dos usuários e enviá-las para o rastreador de localização da Amazon. Também mostrará alterações na localização do usuário, pontos de rastreamento e endereço atual no mapa.

Sua inscrição de início rápido está completa. Este tutorial mostrou como criar um aplicativo iOS que:

- Cria um mapa com o qual os usuários podem interagir.
- Manipula vários eventos do mapa associados à alteração da visualização do mapa pelo usuário.
- Chama uma API do Amazon Location Service, especificamente para pesquisar o mapa em um local, usando a `searchByPosition` API do Amazon Location.

## Próximas etapas

O código-fonte desse aplicativo está disponível em [GitHub](#).

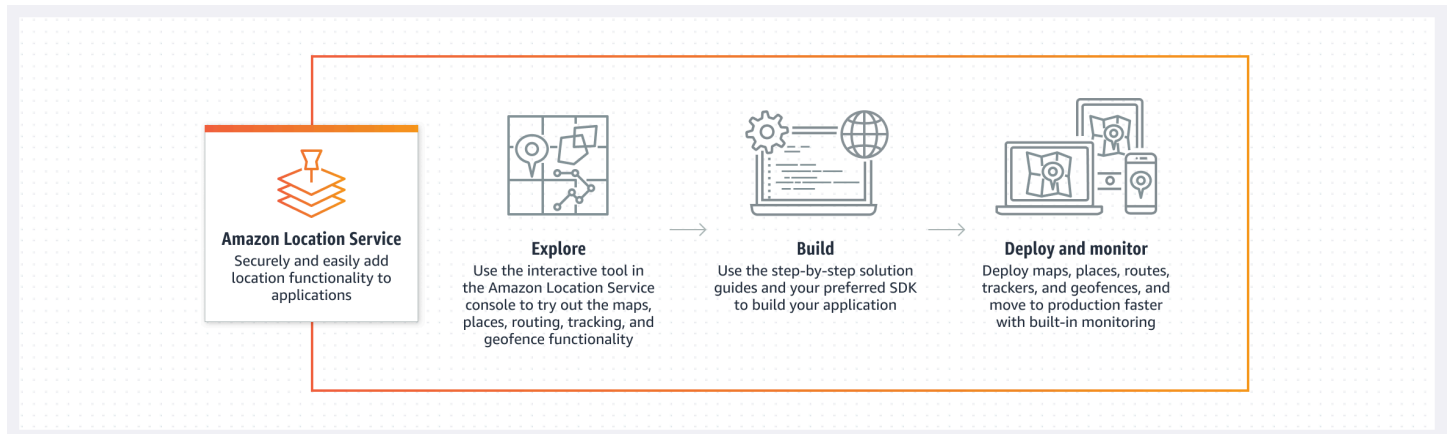
Para aproveitar melhor o Amazon Location, você pode conferir os seguintes recursos:

- Saiba mais sobre os [conceitos do Amazon Location Service](#)
- Obtenha mais informações sobre [como usar os atributos e funcionalidades do Amazon Location](#)
- Veja como expandir essa amostra e criar aplicativos mais complexos analisando [exemplos de código que usam o Amazon Location](#)



# Conceitos do Amazon Location Service

Com o Amazon Location Service, você pode adicionar dados de localização ao seu aplicativo com segurança. Explore alguns dos recursos usando a [ferramenta visual e interativa](#), disponível no console do Amazon Location. Usando a ferramenta de exploração, você pode manipular um mapa padrão, pesquisar pontos de interesse, desenhar geocercas em torno de áreas de interesse e simular o envio de localizações de dispositivos para um rastreador.



Quando estiver pronto para construir, crie seus recursos e escolha entre uma variedade de estilos de mapas e provedores de dados. Em seguida, você pode instalar o SDK que corresponde ao seu ambiente de desenvolvimento e usar o Amazon Location APIs usando as instruções deste guia. Além disso, você pode integrar o monitoramento usando Amazon CloudWatch AWS CloudTrail e.

Os tópicos desta seção fornecem uma visão geral dos principais conceitos do Amazon Location e preparam você para começar a trabalhar com localização em seus próprios aplicativos.

## Tópicos

- [Visão geral do Amazon Location](#)
- [Mapas](#)
- [Pesquisa de lugares](#)
- [Rotas](#)
- [Geocercas e rastreadores](#)
- [Casos de uso comuns para o uso do Amazon Location Service](#)
- [O que é um provedor de dados?](#)
- [Regiões e endpoints do Amazon Location](#)

- [Amazon Location Service quotas](#)

## Visão geral do Amazon Location

O Amazon Location Service fornece acesso a funcionalidades baseadas em localização e provedores de dados por meio de recursos da AWS. A Amazon Location oferece cinco tipos de AWS recursos, dependendo do tipo de funcionalidade que você precisa. Use os diferentes recursos juntos para criar um aplicativo completo e baseado em localização. Você pode criar um ou mais desses recursos usando o console de localização da Amazon, a localização APIs da Amazon ou SDKs o.

Cada recurso define o [provedor de dados](#) subjacente a ser usado (quando aplicável) e dá acesso à funcionalidade relacionada a seu tipo.

Por exemplo:

- O [Amazon Location Service Maps](#) permite que você escolha um mapa de um provedor de mapas para usar em seu aplicativo móvel ou web.
- O [Amazon Location Service Places](#) permite que você escolha um provedor de dados para pesquisar pontos de interesse, preencher texto parcial, geocodificar e fazer geocodificação reversa.
- O [Amazon Location Service Routes](#) permite que você escolha um provedor de dados, encontre rotas e estime o tempo de viagem com base nas informações up-to-date rodoviárias e de trânsito ao vivo.
- O [Amazon Location Service Geofences](#) permite que você defina áreas de interesse como uma fronteira virtual. Assim, você pode avaliar os locais comparando uns aos outros e receber notificações de eventos de entrada e saída.
- O [Amazon Location Service Trackers](#) recebe atualizações de localização dos seus dispositivos. Você pode vincular rastreadores a coleções de geocercas para que todas as atualizações de posição sejam avaliadas automaticamente em relação às suas geocercas.

Você pode usar IAM políticas para gerenciar e autorizar o acesso aos seus recursos de localização da Amazon. Você também pode organizar seus recursos em grupos de recursos para gerenciar e automatizar tarefas à medida que o número de recursos aumenta. Para obter mais informações

sobre o gerenciamento de AWS recursos, consulte [O que são AWS Resource Groups?](#) No Guia do Usuário do AWS Resource Groups.

A localização é definida usando coordenadas de latitude e longitude que seguem o Sistema [Geodésico Mundial \(WGS84\)](#), comumente usado como sistema de referência de coordenadas padrão para serviços do Sistema de Posicionamento Global (GPS).

As seções a seguir descrevem como os componentes do Amazon Location funcionam.

## Mapas

O recurso Amazon Location Service Map fornece acesso aos dados subjacentes do mapa de base para um mapa. Você usa o recurso Map com uma biblioteca de renderização de mapas para adicionar um mapa interativo ao seu aplicativo. Você pode adicionar outras funcionalidades ao seu mapa, como marcadores (ou pinos), rotas e áreas poligonais, conforme necessários para seu aplicativo.

### Note

Para obter informações sobre como usar recursos de mapas na prática, consulte [Usando o Amazon Location Maps em seu aplicativo](#).

Veja a seguir uma visão geral de como criar e usar os recursos de mapas:

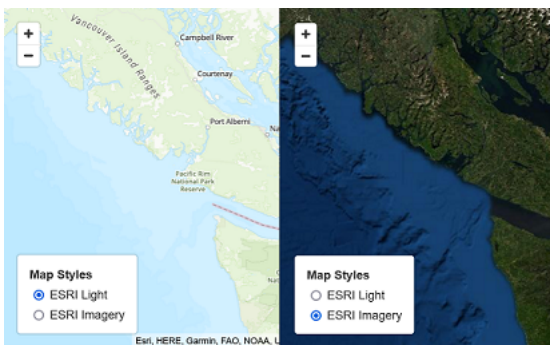


1. Você cria um recurso de mapa em sua AWS conta selecionando um estilo de mapa de um provedor de dados.
2. Em seguida, você pode selecionar e instalar o SDK que corresponda ao seu ambiente de desenvolvimento e aplicativos. Para obter mais informações sobre as opções disponíveis, consulte o tópico sobre como [Acessar o Amazon Location](#).
3. Para exibir um mapa em seu aplicativo, combine um recurso de mapa com uma biblioteca de renderização, como Amplify ou MapLibre Tangram. Para obter mais informações, consulte [Utilizar mapas](#) neste guia.

- Em seguida, você pode integrar o monitoramento usando serviços, como a Amazon CloudWatch e AWS CloudTrail com o Amazon Location. Para obter mais informações, consulte [Monitorando o Amazon Location Service com a Amazon CloudWatch](#) e [Registro e monitoramento com o AWS CloudTrail](#).

## Estilos de mapa

Quando criar um recurso de mapa, você deve escolher um estilo de mapa para esse recurso. Os estilos de mapa definem a aparência do mapa renderizado. Por exemplo, a imagem a seguir mostra o mesmo provedor de dados com dois estilos diferentes de diferentes recursos de mapas no Amazon Location. Um estilo é um típico estilo de estrada, baseado nos dados vetoriais no mapa. O outro estilo inclui dados raster mostrando imagens de satélite. O estilo pode mudar quando você aumenta ou diminui o zoom no mapa, mas normalmente os estilos têm um tema constante. É possível substituir partes ou todas as informações de estilo antes de passá-las para a biblioteca de renderização de mapas.



## Visões políticas

Certos estilos de mapas no Amazon Location Service são compatíveis com visões políticas adicionais.

### Note

A visão política deve ser usada conforme as leis aplicáveis, incluindo as leis sobre o mapeamento do país ou região onde os mapas, imagens e outros dados e conteúdos de terceiros que você acessa por meio do Amazon Location Service são disponibilizados.

Os estilos de mapa a seguir oferecem suporte a uma visão política da Índia (IND).

- [Estilos de mapa da Esri:](#)
  - Navegação da Esri
  - Esri Light
  - Mapa de ruas da Esri
  - Esri Dark Gray Canvas
  - Esri Light Gray Canvas
- [Estilos de mapas de dados abertos:](#)
  - Padrão claro de dados abertos
  - Padrão escuro de dados abertos
  - Visualização clara de dados abertos
  - Visualização escura de dados abertos

No console do Amazon Location Service, você pode filtrar os estilos apresentados para mostrar apenas os estilos que apoiam a visão política da Índia.

## Camadas personalizadas

Uma camada personalizada é uma camada adicional que você pode habilitar para um estilo de mapa. Atualmente, somente o estilo do VectorEsriNavigation mapa suporta a camada POI personalizada.

Quando você habilita a camada personalizada POI, ela adiciona um conjunto mais rico de lugares, como lojas, serviços, restaurantes, atrações e outros pontos de interesse no seu mapa. Por padrão, a camada personalizada é unset. Para obter mais informações, consulte, [MapConfiguration](#) na API Referência de localização.

## Renderização de mapas

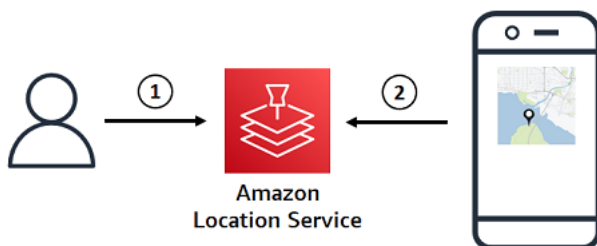
Para renderizar um mapa no seu aplicativo, você normalmente usará uma biblioteca de renderização de mapas. Existem várias opções comuns de uso das bibliotecas:

- MapLibre— MapLibre é uma biblioteca de código aberto específica para renderizar mapas interativos e é o método preferido de renderização de mapas do Amazon Location Service. MapLibre inclui a capacidade de renderizar dados raster e vetoriais de uma fonte de dados (como um recurso de mapa de localização da Amazon). Você pode estender MapLibre para desenhar seus próprios dados no mapa.

- **Amplify:** o Amplify é uma estrutura de código aberto para criar aplicativos para a web, iOS, Android e muito mais. Se seu aplicativo usa o Amplify, você pode estendê-lo para incluir a funcionalidade do Amazon Location. O Amplify inclui bibliotecas específicas para criar aplicativos baseados no Amazon Location, incluindo a renderização de mapas. O Amplify usa MapLibre para renderizar o mapa, mas fornece funcionalidades adicionais específicas do Amazon Location Service para torná-lo mais eficiente de usar, além de adicionar pesquisa e outras funcionalidades.
- **Tangram** — O Tangram é uma biblioteca alternativa de código aberto que renderiza mapas interativos, semelhantes a. MapLibre

A biblioteca de renderização de mapas extrai dados do Amazon Location Service no runtime, renderizando os dados de mapas com base no recurso do mapa selecionado. O recurso de mapa define o provedor de dados e o estilo de mapa que serão usados.

A imagem a seguir mostra como o recurso de mapas é usado no Amazon Location Service juntamente com uma biblioteca de renderização de mapas para criar o mapa final.



1. Você cria um recurso de mapa no Amazon Location Service, usando o AWS Management Console ou AWS CLI. Ele define o provedor de dados e o estilo de mapa que você deseja usar.
2. Seu aplicativo inclui uma biblioteca de renderização de mapas. Você dá à biblioteca de renderização de mapas o nome do recurso de mapas a ser usado. A biblioteca de renderização de mapas extrai dados e informações de estilo desse recurso de mapas do Amazon Location e renderiza o mapa na tela.

## Terminologia dos mapas

### Recurso de mapas

Permite que você acesse dados de mapas de um provedor selecionado. Use o recurso de mapas para buscar blocos de mapas que contêm dados dos mapas e um descritor de estilo para especificar como as funções são renderizadas em um mapa.

## Mapa-base

Fornecer contexto geográfico para o seu mapa, que fica armazenado como camadas de mosaicos vetoriais. As camadas de mosaico incluem contexto geográfico como nomes de ruas, edifícios e uso do solo para referência visual.

## Vetor

Os dados vetoriais são dados sobre formas compostas por pontos, linhas e polígonos. Geralmente são usados para armazenar e exibir estradas, locais e áreas em um mapa. Uma forma vetorial também pode ser usada como ícone para marcadores em um mapa.

## Raster

Dados rasterizados são dados de imagem compostos por uma grade e, geralmente, por cores. Geralmente eles são usados para armazenar e exibir uma representação de dados contínuos em mapas, como terreno, imagens de satélite ou mapas de calor. Imagens rasterizadas também podem ser usadas como imagens ou ícones.

## Estilos de mapa

Os dados vetoriais não incluem inerentemente informações sobre como desenhar as camadas de dados para criar o mapa final. Um estilo de mapa define a cor e outras informações de estilo dos dados para definir sua aparência quando renderizados. Os recursos do mapa incluem informações de estilo do mapa.

O Amazon Location Service fornece estilos segundo a [especificação de estilo Mapbox GL](#).

## Camada vetorial

Um formato em camadas que armazena dados do mapa usando formas vetoriais. Esses dados resultam em um mapa que pode se ajustar à resolução da tela e renderizar seletivamente os atributos de várias formas, mantendo um tamanho de arquivo pequeno para um desempenho ideal.

Formato de arquivo vetorial suportado: Mapbox Vector Tiles (MVT).

## Arquivo com glifos

Um arquivo binário contendo caracteres Unicode codificados. Usado por um renderizador de mapas para exibir rótulos.

## Arquivo Sprite

Um arquivo de imagem Portable Network Graphic (PNG) que contém pequenas imagens rasterizadas, com descrições de localização em um JSON arquivo. Usado por um renderizador de mapas para renderizar ícones ou texturas em um mapa.

# Pesquisa de lugares

Uma função essencial do Amazon Location Service é a capacidade de pesquisar as informações de geolocalização. O Amazon Location fornece essa funcionalidade por meio do recurso Índice de local.

### Note

Para informações sobre como usar recursos de Índice de local para pesquisar na prática, consulte [Pesquisando dados de localização e geolocalização usando Amazon Location](#).

Você pode usar o índice de locais APIs para pesquisar:

- Pontos de interesse, como restaurantes e pontos turísticos. Por nome e localização opcional da pesquisa e receber uma lista de opções ordenadas por relevância.
- Um endereço, recebendo uma latitude e longitude para esse endereço. Isso é conhecido como geocodificação.
- Uma posição de latitude e longitude, recebendo o endereço associado ou outras informações sobre o local. Isso é conhecido como geocodificação reversa.
- Uma consulta de texto de formato livre parcial ou com ortografia incorreta, normalmente quando o usuário digita. Isso é conhecido como preenchimento automático, sugestão automática ou correspondência difusa

O índice de local inclui qual provedor de dados usar para a pesquisa.



**Note**

Os dados do mapa e outras informações de geolocalização, inclusive localizações exatas, podem variar entre os provedores de dados. Como boa prática, use o mesmo provedor de dados para o seu índice de local, mapa e outros recursos do Amazon Location. Por exemplo, se os locais informados pelo seu índice de local não corresponderem à localização dos mesmos locais fornecidos pelo seu recurso de mapa, você poderá colocar um marcador no que parece ser a localização errada no mapa.

Veja a seguir como criar e usar recursos de índice de local:



1. Primeiro, você cria um recurso de índice de localização em sua AWS conta selecionando um provedor de dados.
2. Em seguida, você pode selecionar e instalar o SDK que corresponda ao seu ambiente de desenvolvimento e aplicativos. Para obter mais informações sobre as opções disponíveis, consulte o tópico sobre como [Acessar o Amazon Location](#).
3. Comece a usar os Location Places da Amazon APIs. Para obter mais informações, consulte o tópico sobre o uso de [Pesquisa de locais](#).
4. Em seguida, você pode integrar o monitoramento usando serviços como Amazon CloudWatch e AWS CloudTrail e. Para obter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

## Conceitos de geocodificação

Um índice de local do Amazon Location fornece uma ação chamada [SearchPlaceIndexForText](#), que permite especificar o texto a ser pesquisado. Por exemplo, você pode pesquisar por:

- Locais: uma busca por **Paris** pode retornar a localização da cidade na França.
- Empresas: uma busca por **coffee shop** pode retornar uma lista de cafeterias, incluindo seus nomes e localizações. Você também pode especificar uma localização para pesquisar ou uma

caixa delimitadora para pesquisar e tornar os resultados mais relevantes. Nesse caso, fornecer uma localização no centro de Seattle, Washington, mostraria cafeterias nessa área.

- Endereços: uma busca por **1600 Pennsylvania Ave, Washington D.C.** pode mostrar a localização da Casa Branca nos Estados Unidos (que está nesse endereço).

Pesquisar textos dessa forma geralmente é chamado de geocodificação, que envolve encontrar uma localização geográfica para o endereço ou local.

O Amazon Location Service também fornece uma ação de geocodificação reversa, chamada de [SearchPlaceIndexForPosition](#). Ela pega uma localização geográfica e mostra o endereço, a empresa ou outras informações sobre o que está naquela localização.

## Resultados da pesquisa

Quando você faz uma solicitação de pesquisa bem-sucedida no Amazon Location Service, um ou mais resultados são exibidos. Cada resultado inclui um rótulo, que é o nome ou a descrição do resultado. Por exemplo, uma pesquisa por **coffee shop** pode exibir um resultado com o rótulo `Hometown Cafe`, informando que uma cafeteria chamada “Hometown Cafe” foi encontrada. Normalmente, o resultado da pesquisa também inclui um endereço estruturado (com propriedades como número do endereço, unidade, rua e código postal). Dependendo do provedor de dados, também incluirá outros metadados, como o país e o fuso horário.

Para uma pesquisa sobre o nome ou categoria de uma empresa (como **coffee shop**), talvez você queira mostrar todos os resultados exibidos em um mapa. Para uma pesquisa de endereço, talvez você queira usar o primeiro resultado automaticamente. Consulte o próximo tópico para obter informações sobre relevância.

## Múltiplos resultados e relevância

Ao pesquisar por texto, o Amazon Location Service geralmente encontrará mais de um resultado único. Por exemplo, uma busca por **Paris** pode retornar a cidade na França, mas também a cidade no Texas. Os resultados são classificados por relevância, conforme determinado pelo provedor de dados.

### Note

Os resultados são exibidos em ordem de relevância de todos os fornecedores. Se você escolher a Esri ou a Grab como o seu provedor de dados, os resultados incluem um valor

de relevância que você pode usar para entender a relevância relativa entre os resultados de uma única solicitação.

Especificar informações adicionais, como o nome do país ou um local para pesquisar, pode alterar a ordem dos resultados, reduzir o número de resultados ou até mesmo alterar o conjunto de resultados retornados. Por exemplo, uma pesquisa por **Paris** em um local no Texas tem mais probabilidade de retornar **Paris, Texas** como o primeiro resultado do que **Paris, France**.

Em um aplicativo interativo, você pode usar a relevância para ajudar na decisão de aceitar o primeiro resultado ou pedir ao usuário que elimine a ambiguidade entre vários resultados retornados. Se o primeiro resultado tiver alta relevância, você pode simplesmente aceitá-lo como a resposta correta. Se houver múltiplos resultados de alta relevância ou nenhum resultado de alta relevância, você pode listar os resultados e deixar que o usuário selecione o melhor deles.

## Resultados de endereço

Você pode pesquisar endereços com o Amazon Location Service usando a mesma ação [SearchPlaceIndexForText](#). Quanto mais informações você fornecer, maior será a probabilidade de o endereço retornado corresponder ao fornecido. Por exemplo, **123 Main St** é menos provável de encontrar um resultado correto do que **123 Main St, Anytown, California, 90210**.

Os endereços têm vários atributos, como rua, número do local, cidade, região e código postal, etc. Esses atributos são usados para encontrar um endereço no índice de local que corresponda ao maior número possível de aspectos. Quanto mais atributos forem encontrados, mais relevante será a correspondência considerada e maior a probabilidade de ela ser retornada.

### Note

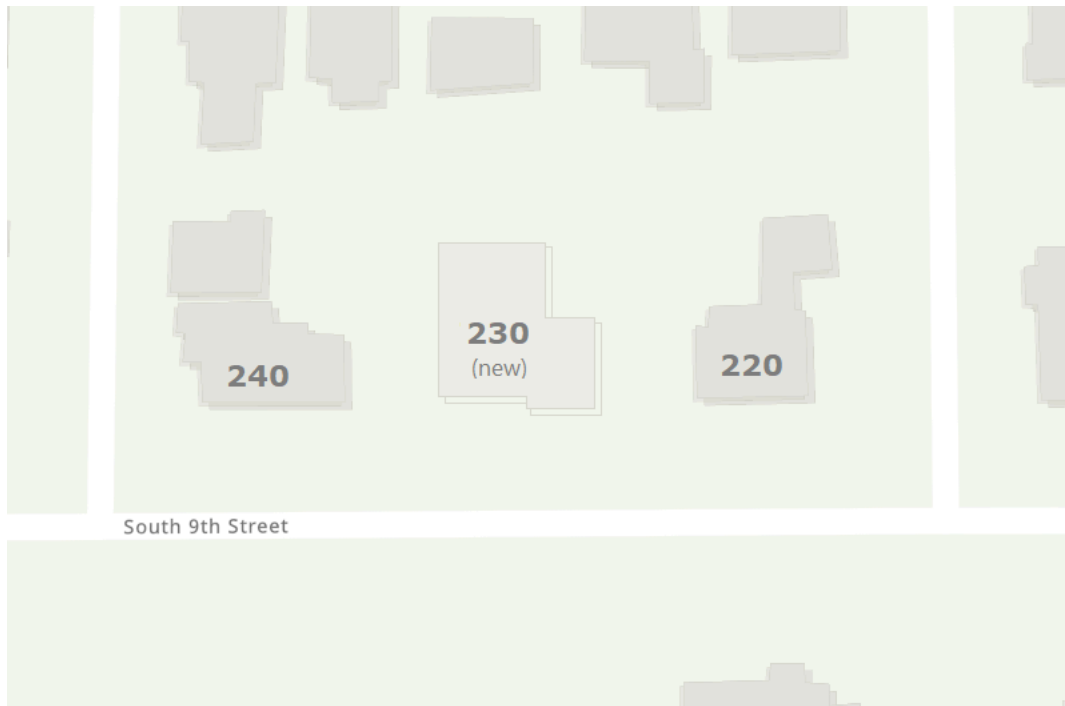
A relevância dos resultados do endereço se baseia em até que ponto o resultado corresponde à entrada. Pode ser o número de atributos correspondentes, mas também pode ser até que ponto os resultados correspondem à entrada. Por exemplo, uma entrada de **123 Main St** teria uma relevância maior quando **Main St** é encontrado nos dados do que se **Main St** fosse o único resultado. **Main St** ainda será exibido, mas provavelmente com um valor de relevância menor.

Os resultados da pesquisa incluem um rótulo para o endereço completo (**123 Main St, Anytown, California, 90210**), mas também os atributos individuais e estruturados do

endereço exibido. Isso é útil porque você pode usar, por exemplo, para preencher campos de endereço em um banco de dados ou para examinar os resultados e encontrar a cidade, região ou código postal da localização encontrada.

## Interpolação

Os endereços nos dados do índice de local incluem correspondências exatas de endereços. Por exemplo, suponha que haja uma rua 9th street e um quarteirão tenha 2 casas, 220 e 240, como na imagem a seguir.



O provedor de dados cria os dados de geolocalização com esses dois endereços conhecidos. Você pode pesquisar esses dois endereços e eles serão encontrados. Depois que o provedor de dados criar os dados do mapa, vamos supor que uma nova casa seja adicionada entre os dois primeiros endereços. Esta nova casa recebe o endereço 230. Se você pesquisar **230 S 9th St**, o provedor de dados ainda encontrará um resultado. Em vez de usar um endereço conhecido, ele interpolará entre os endereços já conhecidos e calculará a posição do novo endereço a partir deles. Nesse caso, ele pode deduzir que 230 está na metade do caminho entre 220 e 240 (e no mesmo lado da rua) e retornar uma localização aproximada com base nisso.

### Note

Os provedores de dados atualizam periodicamente seus dados de geolocalização com novos endereços. Nesse caso, 230 S 9th St seria adicionado aos dados do provedor, mas

normalmente haverá um período em que um novo endereço foi criado, mas ainda não foi adicionado aos dados.

Nesse caso, o provedor de dados não sabe se o novo endereço existe no mundo, pois ele ainda não está nos dados, mas fornece a melhor resposta possível com base nas informações que possui. Esse resultado é chamado de interpolado e pode ser retornado pelo provedor de dados nos resultados. Se `interpolated` retornar `false`, é um endereço conhecido. Se retornar `true`, é um endereço aproximado. Se não for retornado, o provedor de dados não informou se o resultado veio de interpolação.

#### Important

O provedor de dados também pode retornar resultados interpolados para endereços que não existem. Por exemplo, neste caso, se você buscasse **232 S 9th St**, o provedor encontraria esse endereço inexistente e retornaria um local próximo a 230, mas no lado 240. Endereços interpolados são úteis para levar você à localização certa, mas é bom ter em mente que eles não são endereços conhecidos.

## Armazenamento de resultados de geocodificação

Ao criar um recurso de índice de local, você deve especificar uma opção de armazenamento de dados (chamada `IntendedUse` no API). Isso pode ser definido como resultados de uso único ou armazenados. Ele está perguntando sobre o uso pretendido dos resultados. Se quiser armazenar os resultados (mesmo para fins de armazenamento em cache), você deve escolher a opção armazenamento e não a opção uso único.

#### Note

Quando você escolhe a opção armazenada (rotulada como Sim, os resultados serão armazenados no console ou escolhendo `storage` no `CreatePlaceIndexAPI`), o Amazon Location Service não armazena os resultados para você. É uma indicação de que você está planejando armazenar os resultados.

Ao analisar como você usará os resultados de suas consultas no Amazon Location Service, você deve sempre estar ciente dos [Termos do serviço AWS](#) aplicáveis.

# Terminologia de locais

## Recurso de índice de local

Permite que você escolha uma fonte de dados para dar suporte às consultas de pesquisas. Por exemplo, você pode pesquisar pontos de interesse, endereços ou coordenadas. Quando uma consulta de pesquisa é enviada para um recurso de índice de local, ela é atendida usando a fonte de dados configurada do recurso.

## Geocodificação

A geocodificação é o processo de pegar uma entrada de texto, buscá-la no índice de local e retornar resultados com a posição.

## Geocodificação reversa

A geocodificação reversa é o processo de receber uma posição e retornar informações sobre essa posição de dentro do índice de local, como endereço, cidade ou empresa naquela localização.

## Relevância

Relevância é até que ponto um resultado corresponde à entrada. Não é uma medida de exatidão.

## Interpolação

A interpolação é o processo de encontrar endereços desconhecidos usando localizações de endereços conhecidos como pontos de orientação.

## ISO3166 códigos de país

O Amazon Location Service Places usa [os códigos de país 3166 da International Organization for Standardization \(ISO\)](#) para se referir a países ou regiões.

Para encontrar o código de um país ou região específica, use a [Plataforma de Navegação ISO Online](#).

# Rotas

Esta seção fornece uma visão geral dos conceitos de roteamento usando o Amazon Location Service.

## Note

Para obter informações sobre como usar os recursos de rotas na prática, consulte [Calcular rotas usando o Amazon Location Service](#).

## Recursos da calculadora de rotas

Os recursos da calculadora de rotas permitem que você encontre rotas e estime o tempo de viagem com base na rede up-to-date rodoviária e nas informações de tráfego ao vivo do provedor de dados escolhido.

Você pode usar as rotas APIs para criar recursos que permitem que seu aplicativo solicite o tempo de viagem, a distância e a geometria da rota entre quaisquer dois locais. Você também pode usar as Rotas API para solicitar o tempo de viagem e a distância entre um conjunto de partidas e destinos em uma única solicitação para calcular uma matriz.

Veja a seguir como criar e usar um recurso de calculadora de rotas:



1. Primeiro, você cria um recurso de calculadora de rotas em sua AWS conta selecionando um provedor de dados.
2. Em seguida, você pode selecionar e instalar o SDK que corresponda ao seu ambiente de desenvolvimento e aplicativos.
3. Comece a usar as rotas de localização da Amazon APIs. Para obter mais informações sobre como usar o roteamento APIs, consulte o tópico em [Calcular rotas usando o Amazon Location Service](#).

4. Em seguida, você pode integrar o monitoramento usando serviços como Amazon CloudWatch AWS CloudTrail e. Para obter mais informações, consulte [Monitorando o Amazon Location Service com a Amazon CloudWatch](#) e [Registro e monitoramento com o AWS CloudTrail](#).

## Cálculo de uma rota

Um recurso da calculadora de rotas do Amazon Location fornece uma ação chamada `CalculateRoute` que você pode usar para criar uma rota entre duas localizações geográficas (a partida e o destino). A rota calculada inclui a geometria para desenhar a rota em um mapa, além do tempo geral e da distância da rota.

### Uso de pontos de parada

Ao criar sua solicitação de rota, você pode adicionar pontos de parada à rota. Esses são pontos entre a partida e o destino que funcionam como paradas ao longo da rota. A rota será calculada por meio de cada um dos pontos de parada especificados. A rota de um ponto na solicitação para o próximo é chamada de Leg. Cada perna inclui uma distância, tempo e geometria dessa parte da rota.

#### Note

Os pontos de parada são roteados na ordem indicada na solicitação. Eles não são reordenados para o caminho mais curto. Consulte a seção [Planejamento de rotas](#) para obter informações sobre como encontrar o caminho mais curto.

Você pode incluir até 25 pontos de parada em uma única solicitação para calcular uma rota.

### Trânsito e horário de partida

Ao calcular uma rota, o Amazon Location Service leva em consideração o tráfego. O tráfego que ele considera é baseado no horário que você especifica. Você pode especificar a partida agora ou fornecer um horário específico em que deseja sair, o que afetará o resultado da rota ao se ajustar ao tráfego no horário especificado.

#### Note

Você pode calcular o horário de chegada usando o horário de partida e o tempo de resposta da rota para estimar a chegada de um motorista, por exemplo.



Se quiser que o Amazon Location não leve em consideração o tráfego, não especifique um horário de partida e não especifique a partida agora. Assim, será calculada uma rota que pressupõe as melhores condições de tráfego para a rota.

## Opções do modo de viagem

Você pode definir o modo de viagem ao calcular uma rota usando o Amazon Location Service. O modo de viagem padrão é de carro, mas você pode alternadamente selecionar caminhão ou andando.

Se especificar o modo carro ou caminhão, você também poderá especificar opções adicionais.

Para o modo carro, você pode especificar que deseja evitar estradas com pedágio ou balsas. Ele tentará evitar balsas e estradas com pedágio, mas ainda assim irá gerar a rota por elas, se forem a única maneira de chegar ao destino.

No modo caminhão, você também pode evitar balsas e estradas com pedágio, mas, além disso, é possível especificar o tamanho e o peso do caminhão para evitar rotas que não acomodem o caminhão.

## Planejamento de rotas

Você pode usar o Amazon Location Service para criar entradas para seu software de planejamento e otimização de rotas. Você pode criar resultados de rotas, incluindo tempo e distância de viagem, para rotas entre um conjunto de posições de partida e um conjunto de posições de destino. Isso é chamado de criar uma matriz de rotas.

### Note

O software de planejamento e otimização de rotas pode resolver muitos cenários diferentes. Por exemplo, o software de planejamento pode usar o conjunto de tempos e distâncias entre os pontos para calcular o caminho mais curto com parada em cada ponto, fornecendo uma rota eficiente para um único motorista. Alternativamente, o software de planejamento pode ser usado para dividir paradas entre vários caminhões, proporcionando eficiência em toda a frota ou para garantir que cada cliente seja visitado dentro do prazo necessário. O Amazon Location fornece eficientemente as funções de roteamento para permitir que o software de planejamento conclua sua tarefa.

Por exemplo, considerando as posições de partida A e B e as posições de destino X e Y, o Amazon Location Service retornará o tempo e a distância da viagem para rotas de A a X, A a Y, B a X e B a Y.

Assim como no cálculo de uma única rota, você pode calcular as rotas com diferentes modos de transporte, itens a evitar e condições de tráfego. Por exemplo, você pode especificar que o veículo é um caminhão com 11 metros de comprimento, e a rota calculada usará essas restrições para determinar o tempo e a distância da viagem. Você não pode incluir pontos de parada em um cálculo de matriz de rotas.

O número de resultados (e rotas calculadas) é o número de posições de partida multiplicado pelo número de posições de destino. Você é cobrado por cada rota calculada, não por cada solicitação ao serviço. Assim sendo, uma matriz de rotas com 10 partidas e 10 destinos será cobrada como 100 rotas.

## Terminologia de rotas

### Recurso da calculadora de rotas

Um AWS recurso que permite estimar o tempo de viagem, a distância e traçar rotas em um mapa com dados de tráfego e rede rodoviária provenientes do provedor de dados escolhido.

Usando os recursos da calculadora de rotas, você calcula rotas para diferentes modos de transporte, desvios e condições de tráfego.

### Rota

Uma rota contém detalhes usados ao viajar por um caminho a partir da posição de partida, posições dos pontos de parada e posição do destino.

Os exemplos de detalhes em uma rota incluem:

- A distância de uma posição até outra posição.
- O tempo necessário para viajar de uma posição à próxima posição.
- A LineString geometria que representa o caminho da rota.

Para obter mais informações sobre rotas, consulte a [sintaxe de resposta para a CalculateRoute operação](#) na API referência de rotas do Amazon Location Service.

### Matriz de rotas

Uma lista de rotas, de um conjunto de posições de partida até um conjunto de posições de destino. Útil como entradas no software de planejamento ou otimização de rotas.

Para obter mais informações sobre o cálculo de uma matriz de rotas, consulte a [sintaxe da CalculateRouteMatrix operação na referência](#) de rotas API do Amazon Location Service.

## Geometria **LineString**

Uma rota do Amazon Location consiste em um ou mais trechos (uma rota de um ponto de parada para outro dentro da rota geral). A geometria de cada perna é um polígono representado como uma **LineString**. Uma **LineString** é uma matriz ordenada de posições que pode ser usada para traçar uma rota em um mapa.

A seguir está um exemplo de uma **LineString** com três pontos:

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

## Ponto de parada

Os pontos de parada são posições intermediárias que servem de paradas ao longo de uma rota entre a posição de partida e a posição de destino. A ordem de parada na rota segue a ordem em que você fornece as posições dos pontos de parada na solicitação.

## Perna

Uma perna única é a viagem de uma posição para outra. Se as posições não estiverem localizadas em uma via, elas serão movidas para a via mais próxima. O número de pernas em uma rota é um a menos que o número total de posições.

Uma rota sem pontos de parada consiste em uma única etapa, da posição de partida até o destino. Uma rota com um ponto de parada consiste em dois trechos, da posição de partida até o ponto de parada e de lá até o destino.

## Etapa

Uma etapa é uma subseção de uma perna. Cada etapa fornece informações resumidas para aquela etapa da perna.

# Geocercas e rastreadores

Esta seção fornece uma visão geral dos conceitos de trabalho com geocercas e rastreadores do Amazon Location Service. As geocercas são limites poligonais que você pode usar para ser notificado quando dispositivos ou posições entram e saem das áreas. Os recursos do rastreador são usados para armazenar e atualizar as posições dos dispositivos enquanto eles se movem.

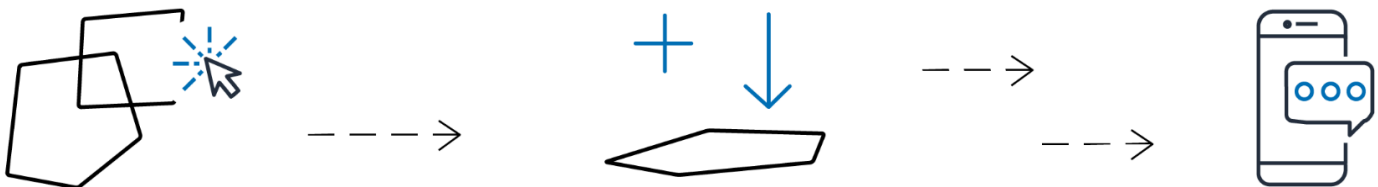
## Note

Para obter informações sobre como usar geocercas e rastreadores na prática, consulte [Colocando geocercas em uma área de interesse usando o Amazon Location](#).

## Geocercas

Os recursos de coleção de geocercas permitem que você armazene e gerencie geocercas, que são limites virtuais em um mapa. Você pode avaliar as localizações em relação a um recurso de coleção de geocercas e receber notificações quando a atualização de localização ultrapassar os limites de qualquer uma das geocercas na coleção de geocercas.

Veja a seguir como criar e usar recursos de coleção de geocercas:



1. Crie um recurso de coleta de cercas geográficas em sua AWS conta.
2. Adicione geocercas a essa coleção. Você pode fazer isso usando a ferramenta de upload de cercas geográficas no console de localização da Amazon ou usando as cercas geográficas de localização da Amazon. API Para obter mais informações sobre as opções disponíveis, consulte [Acesso ao Amazon Location](#).

As geocercas podem ser definidas por um polígono ou por um círculo. Use um polígono para descobrir quando um dispositivo entra em uma área específica. Use um círculo para descobrir quando um dispositivo está a uma certa distância (raio) de um ponto.

3. Você pode começar a avaliar as localizações em relação a todas as suas geocercas. Quando uma atualização de localização ultrapassa os limites de uma ou mais cercas geográficas, seu recurso de coleta de cercas geográficas emite um dos seguintes tipos de eventos de cercas geográficas na Amazon: EventBridge

- ENTER— Um evento é gerado para cada cerca geográfica em que a atualização de localização cruza seu limite ao entrar nela.
- EXIT— Um evento é gerado para cada cerca geográfica em que a atualização de localização cruza seus limites ao sair dela.

Para obter mais informações, consulte [the section called “Reagindo aos eventos com EventBridge”](#). Você também pode integrar o monitoramento usando serviços como Amazon CloudWatch AWS CloudTrail e. Para obter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

Por exemplo, se você estiver rastreando uma frota de caminhões e quiser ser notificado quando um caminhão chegar a uma determinada área de qualquer um dos seus armazéns. Você pode criar uma geocerca para a área ao redor de cada depósito. Então, quando os caminhões enviarem localizações atualizadas, você pode usar o Amazon Location Service para avaliar essas posições e ver se um caminhão entrou (ou saiu) de uma das áreas de geocercas.

#### Note

Você é cobrado pelo número de coleções de geocercas que você avalia. Sua fatura não é afetada pelo número de geocercas em cada coleção. Como cada coleção de geocercas pode conter até 50 mil geocercas, convém combinar suas geocercas em menos coleções, sempre que possível, para reduzir o custo das avaliações das geocercas. Os eventos gerados incluirão a ID da geocerca individual na coleção, bem como a ID da coleção.

## Eventos de geocercas

As localizações das posições que você está monitorando são referenciadas por uma ID chamada de DeviceId (e as posições são referenciadas como posições do dispositivo). Você pode enviar uma lista de posições do dispositivo para avaliação diretamente com o recurso de coleção de geocercas ou usar um rastreador. Consulte a próxima seção para mais informações sobre rastreadores.

Você recebe eventos (via Amazon EventBridge) somente quando um dispositivo entra ou sai de uma cerca geográfica, não para cada mudança de posição. Isso significa que normalmente você receberá eventos e precisará responder a eles com muito menos frequência do que cada atualização de posição do dispositivo.

#### Note

Para a primeira avaliação de localização de um DeviceID específico, presume-se que o dispositivo antes não estava em nenhuma geocerca. Portanto, a primeira atualização irá gerar um evento ENTER, se estiver dentro de uma geocerca na coleção, e nenhum evento se não estiver.

Para calcular se um dispositivo entrou ou saiu de uma geocerca, o Amazon Location Service deve manter o estado da posição anterior do dispositivo. Esse estado de posição é armazenado por 30 dias. Depois de 30 dias sem atualização para um dispositivo, uma nova atualização de localização será tratada como a primeira atualização de posição.

## Rastreadores

Um rastreador armazena atualizações de posição para uma coleção de dispositivos. O rastreador pode ser usado para consultar a localização atual ou o histórico de localização dos dispositivos. Ele armazena as atualizações, mas reduz o espaço de armazenamento e o ruído visual ao filtrar os locais antes de armazená-los.

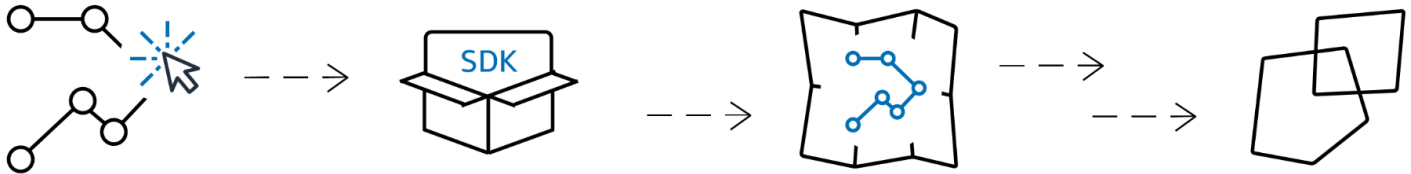
Cada atualização de posição armazenada nos recursos do seu rastreador pode incluir uma medida da precisão da posição e até 3 campos de metadados sobre a posição ou o dispositivo que você deseja armazenar. Os metadados são armazenados como pares chave-valor e podem armazenar informações como velocidade, direção, pressão dos pneus ou temperatura do motor.

#### Note

O armazenamento do rastreador é criptografado automaticamente com chaves AWS próprias. Você pode adicionar outra camada de criptografia usando KMS chaves que você gerencia, para garantir que somente você possa acessar seus dados. Para obter mais informações, consulte [Criptografia de dados em repouso para o Amazon Location Service](#).

A filtragem e o armazenamento da posição do rastreador são úteis por si só, mas os rastreadores são particularmente úteis quando combinados com geocercas. Você pode vincular rastreadores a um ou mais dos seus recursos de coleção de geocercas e as atualizações de posição são avaliadas automaticamente em relação às geocercas dessas coleções. O uso adequado da filtragem também pode reduzir consideravelmente os custos de suas avaliações de geocercas.

O diagrama abaixo mostra como criar e usar os recursos do rastreador:



1. Primeiro, você cria um recurso de rastreamento na sua AWS conta.
2. Em seguida, decida como você envia atualizações de localização para seus recursos do rastreador. Use [AWS SDKs](#) para integrar recursos de rastreamento em seus aplicativos móveis. Como alternativa, você pode usar MQTT seguindo as step-by-step instruções no [rastreamento de uso. MQTT](#)
3. Agora você pode usar seu recurso de rastreamento para registrar o histórico de localização e visualizá-lo em um mapa.
4. Você também pode vincular seu recurso de rastreador a uma ou mais coleções de geocercas para que cada atualização de posição enviada ao seu recurso de rastreamento seja avaliada automaticamente em relação a todas as coleções de geocercas vinculadas. Você pode vincular o recurso na página de detalhes dos recursos do rastreador do console de localização da Amazon ou usando os Amazon Location API Trackers.
5. Em seguida, você pode integrar o monitoramento usando serviços como Amazon CloudWatch AWS CloudTrail e. Para obter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

## Utilizar rastreadores com geocercas

Os rastreadores fornecem uma funcionalidade adicional quando combinados com geocercas. Você associa um rastreador a uma coleção de cercas geográficas, seja por meio do console de localização da Amazon ou do API, para avaliar automaticamente as localizações dos rastreadores. Cada vez que o rastreador recebe uma localização atualizada, essa localização será avaliada em relação a cada cerca geográfica da coleção, e os EXIT eventos apropriados ENTER serão gerados

na Amazon. EventBridge Você também pode aplicar a filtragem ao rastreador e, dependendo da filtragem, pode reduzir os custos das avaliações de geocercas avaliando somente as atualizações de localização significativas.

Se você associar o rastreador a uma coleção de geocercas depois que ele já tiver recebido algumas atualizações de posição, a primeira atualização de posição após a associação será tratada como uma atualização inicial para as avaliações da geocerca. Se estiver dentro de uma geocerca, você receberá um evento ENTER. Se não estiver dentro de nenhuma geocerca, você não receberá um evento EXIT, independentemente do estado anterior.

## Filtragem de posição

Os rastreadores podem filtrar automaticamente as posições que são enviadas a eles. Há várias razões pelas quais você pode querer filtrar algumas atualizações de localização do seu dispositivo. Se você tem um sistema que envia relatórios a cada minuto, talvez queira filtrar os dispositivos por tempo, armazenando e avaliando as posições somente a cada 30 segundos. Mesmo se você estiver monitorando com mais frequência, talvez queira filtrar as atualizações de posição para eliminar o ruído do GPS hardware. GPSs locais das posições são inerentemente ruidosos. Sua precisão não é 100% perfeita, então até mesmo um dispositivo estacionário parece estar se movendo um pouco. Em baixas velocidades, essa instabilidade causa confusão visual e pode causar eventos falsos de entrada e saída se o dispositivo estiver próximo à borda de uma geocerca.

A filtragem de posição funciona quando as atualizações de posição são recebidas por um rastreador, reduzindo o ruído visual nos caminhos do dispositivo (instabilidade), reduzindo o número de eventos falsos de entrada e saída da geocerca e ajudando a gerenciar custos ao reduzir o número de atualizações de posição armazenadas e avaliações de geocercas acionadas.

Os rastreadores oferecem três opções de filtragem de posições para ajudar a gerenciar custos e reduzir a instabilidade nas atualizações de localização.

- Baseado na precisão: use com qualquer dispositivo que forneça uma medição de precisão. A maioria dos GPS dispositivos móveis fornece essas informações. A precisão de cada medição de posição é afetada por muitos fatores ambientais, incluindo recepção de GPS satélite, paisagem e proximidade de dispositivos wi-fi e bluetooth. A maioria dos dispositivos, incluindo a maioria dos dispositivos móveis, pode fornecer uma estimativa da precisão da medição junto com a medição. Com a filtragem AccuracyBased, o Amazon Location ignora atualizações de localização se o dispositivo tiver se movido menos em relação à precisão medida. Por exemplo, se duas atualizações consecutivas de um dispositivo tiverem uma faixa de precisão de 5 m e 10 m, o



Amazon Location ignora a segunda atualização se o dispositivo tiver se movido menos de 15 m. O Amazon Location não avalia nem armazena as atualizações ignoradas em relação a geocercas.

Quando a precisão não é fornecida, ela é tratada como zero, a medição é considerada perfeitamente precisa e nenhuma filtragem será aplicada às atualizações.

### Note

Você pode usar a filtragem baseada em precisão para remover toda a filtragem. Se você selecionar a filtragem com base na precisão, mas substituir todos os dados de precisão para zero ou omitir totalmente a precisão, o Amazon Location não filtrará nenhuma atualização.

Na maioria dos cenários, a filtragem baseada em precisão é uma boa opção para filtrar atualizações de posição, fornecendo um equilíbrio da localização de rastreamento e filtrando atualizações desnecessárias, reduzindo assim os custos.

- Baseada na distância: use quando seus dispositivos não fornecem uma medição precisa, mas você quer aproveitar a filtragem para reduzir a instabilidade e gerenciar os custos mesmo assim. A filtragem `DistanceBased` ignora as atualizações de localização nas quais os dispositivos se moveram menos de 30 m (98,4 pés). Quando você usa a filtragem de posições `DistanceBased`, o Amazon Location não avalia nem armazena essas atualizações ignoradas em relação a geocercas.

A precisão da maioria dos dispositivos móveis, incluindo a precisão média dos dispositivos iOS e Android, é de cerca de 15 m. Na maioria dos aplicativos, a filtragem `DistanceBased` pode reduzir o efeito das imprecisões de localização ao exibir a trajetória do dispositivo em um mapa e o efeito de vai e vem de vários eventos consecutivos de entrada e saída quando os dispositivos estão próximos à borda de uma geocerca. Ela também pode ajudar a reduzir o custo do seu aplicativo, fazendo menos chamadas para avaliar as geocercas vinculadas ou recuperar as posições dos dispositivos.

A filtragem baseada em distância é útil se você quiser filtrar, mas seu dispositivo não fornecer medições de precisão, ou se você deseja filtrar um número maior de atualizações do que com base na precisão.

- Baseada no tempo: (padrão) use quando seus dispositivos enviam atualizações de posição com muita frequência (mais de uma vez a cada 30 segundos) e você deseja obter as avaliações da

geocerca em tempo quase real sem armazenar todas as atualizações. Na filtragem TimeBased, cada atualização de localização é avaliada em relação a coleções de geocercas vinculadas, mas nem todas as atualizações de localização são armazenadas. Se a frequência de atualização for superior a 30 segundos, apenas uma atualização a cada 30 segundos será armazenada para cada ID de dispositivo exclusivo.

A filtragem baseada em tempo é particularmente útil quando você deseja armazenar menos posições, mas deseja que cada atualização de posição seja avaliada em relação às coleções de geocercas associadas.

### Note

Esteja atento aos custos do seu aplicativo de rastreamento ao decidir seu método de filtragem e a frequência das atualizações de posição. Você é cobrado por cada atualização de localização e uma vez para avaliar a atualização de posição em relação a cada coleção de geocercas vinculada. Por exemplo, ao usar a filtragem baseada no tempo, se seu rastreador estiver vinculado a duas coleções de geocercas, cada atualização de posição contará como uma solicitação de atualização de localização e duas avaliações da coleção de geocercas. Se você estiver registrando atualizações de posição a cada cinco segundos para seus dispositivos e usando a filtragem com base no tempo, você será cobrado por 720 atualizações de localização e 1.440 avaliações da geocerca por hora para cada dispositivo.

## Terminologia de geocercas

### Coleção de geocercas

Contém zero ou mais geocercas. É capaz de monitorar a geocerca emitindo eventos de entrada e saída, quando solicitado, para avaliar a posição de um dispositivo em relação às suas geocercas.

### Geocerca

Geometria poligonal ou circular que define um limite virtual em um mapa.

## Geometria poligonal

Uma geocerca do Amazon Location é um limite virtual para uma área geográfica e é representada como um polígono ou como um círculo.

Um círculo é um ponto com uma distância ao seu redor. Use um círculo quando quiser ser notificado se um dispositivo estiver a uma certa distância de um local.

Um polígono é uma matriz composta por 1 ou mais anéis lineares. Use um polígono quando quiser definir um limite específico para notificações do dispositivo. Um anel linear é uma matriz de quatro ou mais vértices, onde o primeiro e o último vértice são iguais para formar um limite fechado. Cada vértice é um ponto bidimensional da forma `[longitude, latitude]`, onde as unidades de longitude e latitude são graus. Os vértices devem ser listados no sentido anti-horário ao redor do polígono.

### Note

O Amazon Location Service não é compatível com polígonos com mais de um anel. Isso inclui furos, ilhas ou multipolígonos. O Amazon Location também não é compatível com polígonos enrolados no sentido horário ou que cruzam o antimeridiano.

Veja a seguir um exemplo de um único anel externo linear:

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

## Terminologia do rastreador

### Recurso de rastreador

Um AWS recurso que recebe atualizações de localização dos dispositivos. O recurso de rastreador fornece suporte para consultas de localização, como localização atual e histórica do dispositivo. A vinculação de um recurso de rastreador a uma coleção de geocercas avalia

automaticamente as atualizações de localização em relação a todas as geocercas na coleção de geocercas vinculadas.

## Dados de posição rastreados

Um recurso de rastreador armazena informações sobre seus dispositivos ao longo do tempo. As informações incluem uma série de atualizações de posição, em que cada atualização inclui localização, horário e metadados opcionais. Os metadados podem incluir a precisão de uma posição e até três pares de valores-chave para ajudar você a rastrear informações importantes sobre cada posição, como velocidade, direção, pressão dos pneus, combustível restante ou temperatura do motor do veículo que você está rastreando. Os rastreadores mantêm o histórico de localização do dispositivo por 30 dias.

## Filtragem de posição

A filtragem de posição pode ajudar você a controlar os custos e melhorar a qualidade do seu aplicativo de rastreamento, filtrando as atualizações de posição que não fornecem informações valiosas antes que as atualizações sejam armazenadas ou avaliadas em relação às geocercas.

Você pode escolher as filtragens `AccuracyBased`, `DistanceBased` ou `TimeBased`. Por padrão, a filtragem de posição está definida como `TimeBased`.

Você pode configurar a filtragem de posição ao criar ou atualizar recursos do rastreador.

## RFCFormato de carimbo de data/hora 3339

O Amazon Location Service Trackers usa o formato [RFC3339](#), que segue o formato 8601 da [International Organization for Standardization \(ISO\) 8601 para](#) datas e horas.

O formato é “YYYY-MM-:mm:ss.SSSZ+ 00:00”: DDThh

- YYYY-MM-DD: representa o formato da data.
- T: indica que os valores de tempo seguirão.
- hh:mm:ss.sss: representa o horário no formato de 24 horas.
- Z— Indica que o fuso horário usado éUTC, que pode ser seguido por desvios do fuso UTC horário.
- +00:00— Opcionalmente, indique desvios do fuso UTC horário. Por exemplo, + 01:00 indica UTC + 1 hora.

## Exemplo

Para 2 de julho de 2020, às 12:15:20 da tarde, com um ajuste de mais 1 hora no fuso UTC horário.

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

## Casos de uso comuns para o uso do Amazon Location Service

O Amazon Location Service permite que você crie uma variedade de aplicativos, desde rastreamento de ativos até marketing baseado em localização. Veja a seguir alguns casos de uso comuns:

### Engajamento do usuário e geomarketing

Usa dados de localização para criar soluções que melhorem o engajamento do usuário com o marketing para atingir clientes-alvo. Por exemplo, o Amazon Location pode acionar um evento que solicita uma notificação quando um cliente que pediu um café em seu aplicativo móvel está por perto. Além disso, você pode criar recursos de segmentação geográfica para que os varejistas possam enviar códigos de desconto ou folhetos digitais para clientes próximos às lojas-alvo.

### Rastreamento de ativos

Cria recursos de rastreamento de ativos para ajudar as empresas a entender a localização atual e histórica de seus produtos, funcionários e infraestrutura. Com os recursos de rastreamento de ativos, você pode criar várias soluções que otimizam a equipe remota, protegem a remessa durante o trajeto e maximizam a eficácia do despacho.

### Entrega

Integra recursos de localização em aplicativos de entrega para armazenar, rastrear e coordenar o local de partida, os veículos de entrega e seus destinos. Por exemplo, um aplicativo de entrega de comida com atributos do Amazon Location integrados tem funcionalidades de rastreamento de localização e delimitação geográfica que podem notificar automaticamente um restaurante quando um entregador estiver por perto. Isso reduz o tempo de espera e ajuda a manter a qualidade dos alimentos entregues.

Este tópico apresenta uma visão geral da arquitetura e das etapas dos aplicativos que você pode criar com o Amazon Location.

### Tópicos

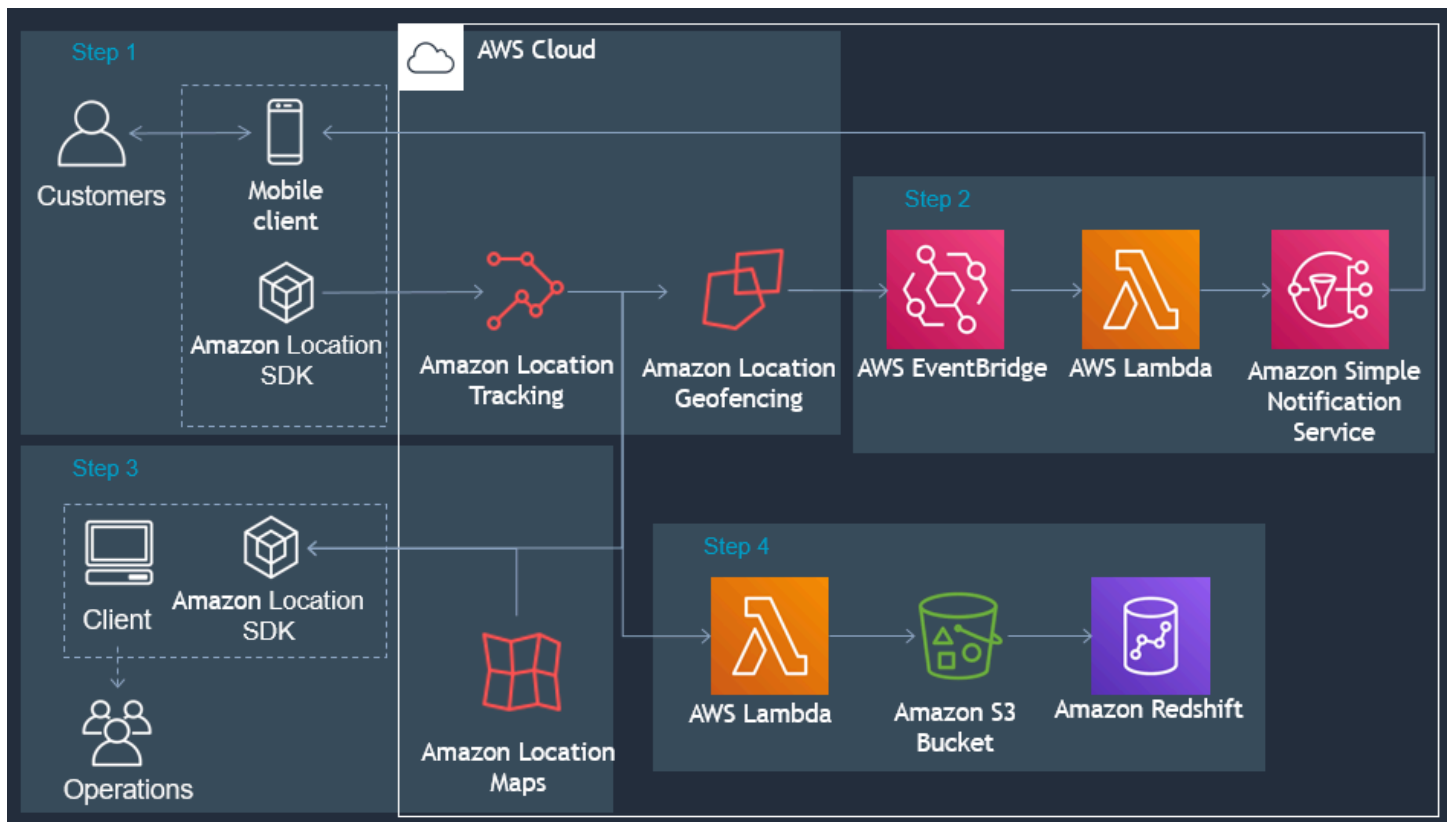
- [Aplicativos de engajamento do usuário e geomarketing](#)
- [Aplicativos de rastreamento de ativos](#)
- [Aplicativos de entrega](#)

## Aplicativos de engajamento do usuário e geomarketing

A seguir está uma ilustração de uma arquitetura de aplicativo de geomarketing e engajamento de usuários com o Amazon Location:

Com essa arquitetura, você pode:

- Iniciar eventos com base na proximidade de um alvo para que você possa enviar ofertas a clientes próximos ou interagir com aqueles que deixaram seu estabelecimento recentemente (chamado de segmentação geográfica).
- Visualizar a localização dos dispositivos do cliente em um mapa para monitorar as tendências ao longo do tempo.
- Armazenar as localizações dos dispositivos dos clientes que você pode analisar ao longo do tempo.
- Analisar o histórico de localização para identificar tendências e oportunidades de otimização.



Abaixo é apresentada uma visão geral das etapas necessárias para criar um aplicativo de engajamento de usuários e geomarketing:

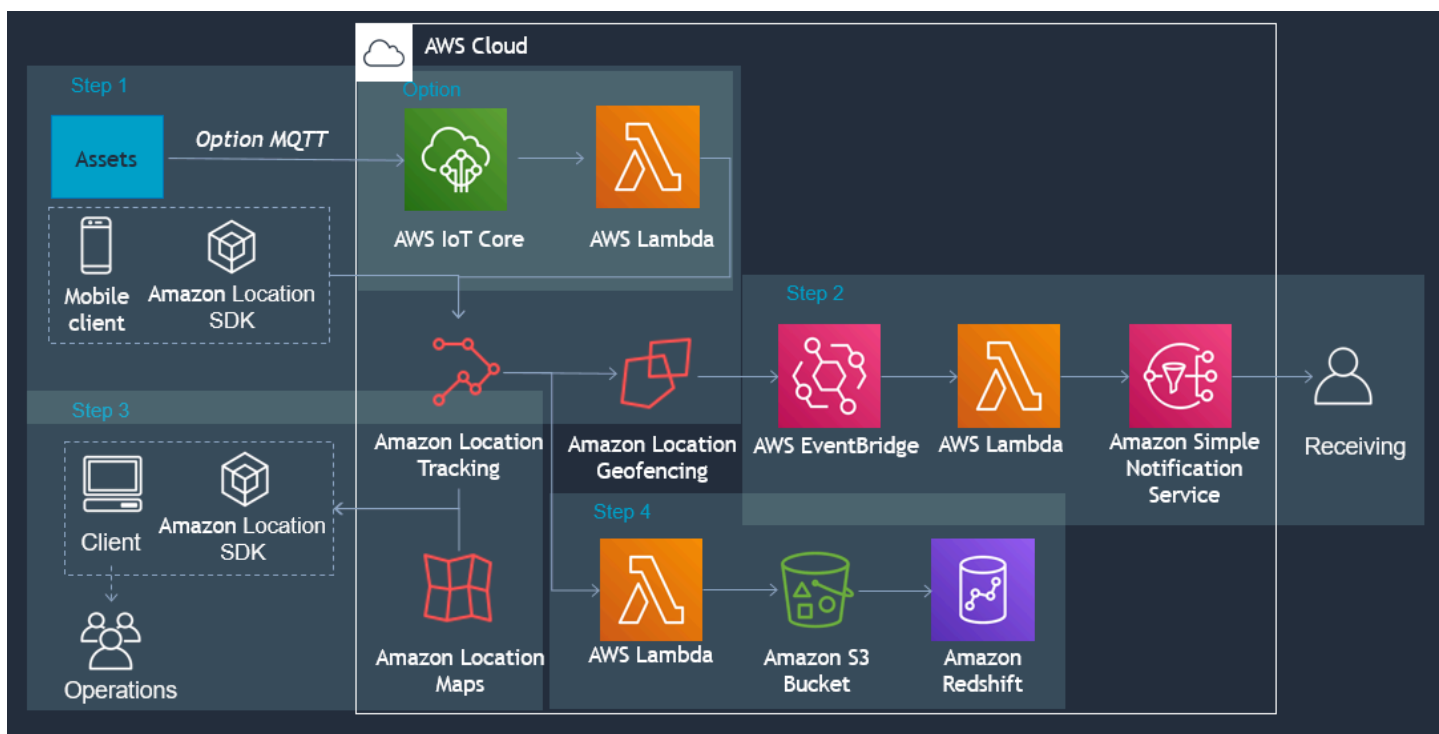
1. Crie suas geocercas em Coleções de Geocercas e vincule Rastreadores a elas. Para obter mais informações, consulte [the section called “Geocercamento e rastreamento”](#).
2. Configure EventBridge a Amazon para enviar uma notificação aos clientes que entram ou saem de uma área geograficamente delimitada de interesse. Para obter mais informações, consulte [the section called “Reagindo aos eventos com EventBridge”](#).
3. Exiba localizações de clientes e geocercas em um mapa. Para obter mais informações, consulte [Utilizar mapas](#).
4. Salve os dados de localização para armazenamento de longo prazo para análise posterior.
5. Depois de criar seu aplicativo, você pode usar a Amazon CloudWatch e AWS CloudTrail gerenciar seu aplicativo. Para ter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

## Aplicativos de rastreamento de ativos

A seguir está uma ilustração de uma arquitetura de aplicativo de rastreamento de ativo usando o Amazon Location:

Com essa arquitetura, você pode:

- Exibir a localização dos ativos em um mapa para ilustrar o panorama geral. Por exemplo, mostrar um mapa de calor usando localizações ou eventos históricos para ajudar uma equipe de operações ou planejamento.
- Iniciar eventos com base na proximidade do ativo para notificar o departamento de recebimento para se preparar para a chegada da remessa e reduzir o tempo de processamento.
- Armazenar a localização dos ativos para iniciar ações em seus aplicativos de backend ou para analisar dados ao longo do tempo.
- Analisar o histórico de localização para identificar tendências e oportunidades de otimização.



Abaixo, é apresentada uma visão geral das etapas necessárias para criar um aplicativo de rastreamento de ativos:

1. Crie suas geocercas em Coleções de Geocercas e vincule Rastreadores a elas. Para obter mais informações, consulte [the section called “Geocercamento e rastreamento”](#).



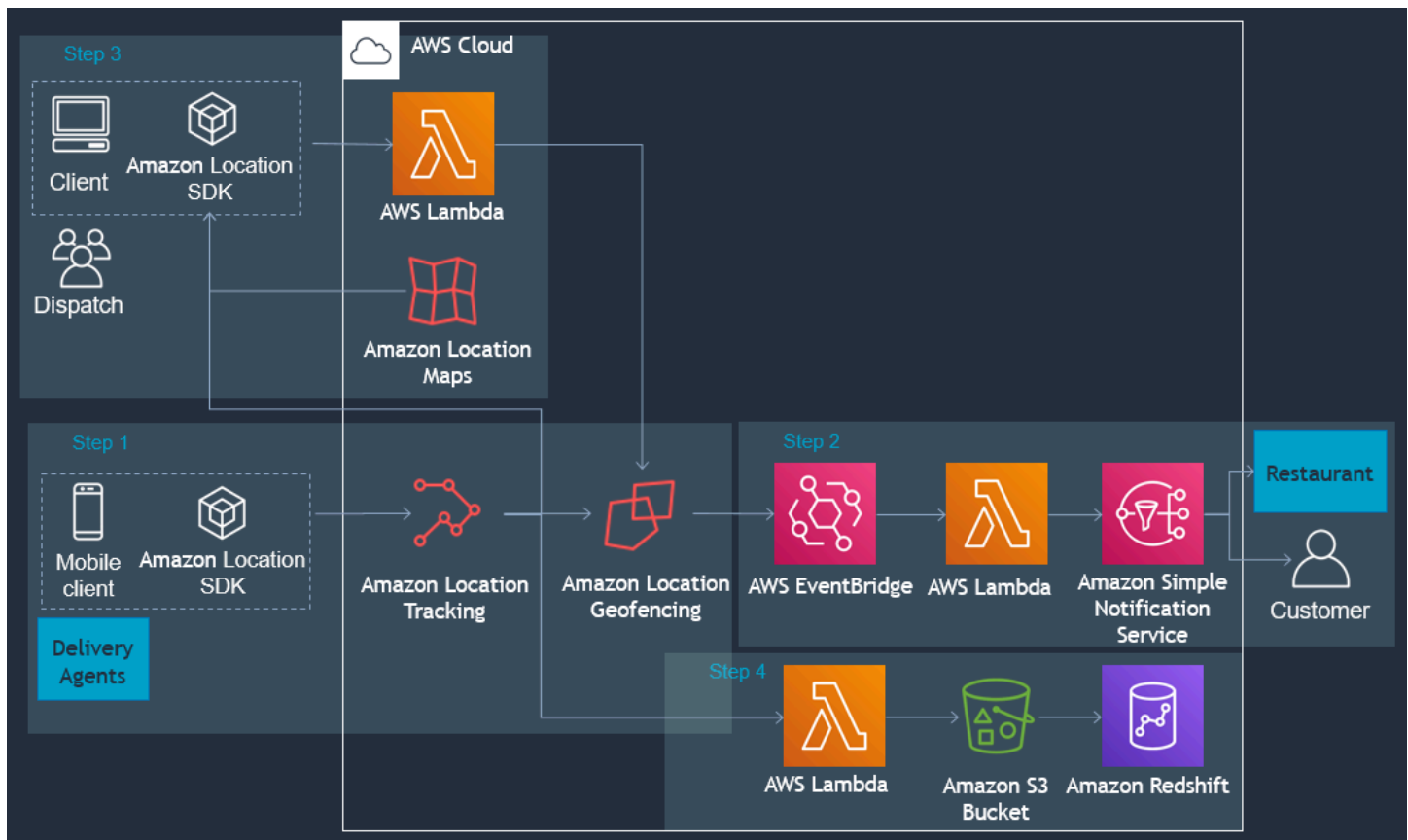
2. Configure EventBridge a Amazon para enviar uma notificação ou iniciar um processo. Para obter mais informações, consulte [the section called “Reagindo aos eventos com EventBridge”](#).
3. Exibir seus ativos rastreados e suas geocercas ativas em um mapa. Para obter mais informações, consulte [Utilizar mapas](#).
4. Salve os dados de localização para armazenamento de longo prazo para análise posterior.
5. Depois de criar seu aplicativo, você pode usar a Amazon CloudWatch e AWS CloudTrail gerenciar seu aplicativo. Para ter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

## Aplicativos de entrega

Veja abaixo a ilustração de uma arquitetura de aplicativo de entrega usando o Amazon Location.

Com essa arquitetura, você pode:

- Iniciar eventos com base na proximidade dos agentes de entrega para que as coletas estejam prontas a tempo e os clientes possam ser notificados quando a entrega chegar.
- Exibir a localização dos motoristas, bem como os locais de coleta e entrega em tempo quase real em um mapa para mostrar às equipes de despacho uma visão geral.
- Armazenar os locais dos agentes de entrega para que você possa utilizá-los em seu aplicativo de back-end ou analisá-los ao longo do tempo.
- Analisar o histórico de localização para identificar tendências e oportunidades de otimização.



Veja abaixo uma visão geral das etapas necessárias para criar um aplicativo de entrega:

1. Crie suas coleções de geocercas e vincule dispositivos rastreados à coleção. Para obter mais informações, consulte [the section called “Geocercamento e rastreamento”](#).
2. Crie uma AWS Lambda função para adicionar e remover automaticamente delimitações geográficas à medida que seus pedidos são reservados.
3. Configure EventBridge a Amazon para enviar notificações ou iniciar um processo. Para obter mais informações, consulte [the section called “Reagindo aos eventos com EventBridge”](#).
4. Exibir ativos rastreados e geocercas ativas em um mapa. Para obter mais informações, consulte [Utilizar mapas](#).
5. Salve os dados de localização para armazenamento de longo prazo para análise posterior.
6. Depois de criar seu aplicativo, você pode usar a Amazon CloudWatch e AWS CloudTrail gerenciar seu aplicativo. Para ter mais informações, consulte [the section called “Monitoramento com CloudWatch”](#) e [the section called “Usando CloudTrail com a Amazon Location”](#).

## O que é um provedor de dados?

Use o Amazon Location Service para acessar recursos de geolocalização de vários provedores de dados por meio de sua conta da AWS sem necessidade de contratos ou integrações de terceiros. Isso pode ajudar você a se concentrar na criação do seu aplicativo, sem precisar gerenciar contas, credenciais, licenças e faturamento de terceiros.

Os seguintes serviços do Amazon Location usam provedores de dados.

- Mapas: escolha estilos de diferentes provedores de mapas ao [criar um recurso de mapa](#). Você pode usar recursos de mapa para criar um mapa interativo para visualizar dados.
- Locais: escolha um provedor de dados ao [criar um recurso de índice de local](#) para dar suporte a consultas de geocodificação, geocodificação reversa e pesquisas.
- Rotas: escolha um provedor de dados para dar suporte a consultas para cálculos de rotas em diferentes geografias e aplicativos ao [criar um recurso de calculadora de rotas](#). Com o provedor de dados escolhido, o Amazon Location Service permite que você calcule rotas com base em dados up-to-date da rede rodoviária, dados de tráfego ao vivo, fechamentos planejados e padrões históricos de tráfego.

Cada provedor coleta e organiza seus dados usando diferentes meios. Eles também podem ter diferente expertise em diferentes regiões do mundo. Esta seção apresenta detalhes sobre nossos provedores de dados. Você pode selecionar qualquer provedor de dados com base na sua preferência.

Lembre-se de ler os termos e condições ao usar os provedores de dados do Amazon Location Service. Para obter mais informações, consulte os [Termos AWS de Serviço](#). Consulte também a seção [the section called “Privacidade de dados”](#) para obter mais informações sobre como o Amazon Location protege sua privacidade.

## Cobertura e atributos do provedor de dados

A tabela a seguir mostra a cobertura e os atributos em alto nível para cada provedor de dados.

Provedor de dados	Cobertura geográfica	Cobertura de atributos	Região da AWS
Esri	Global	Mapas, locais, rotas	<a href="#">Todas as regiões</a> em que o Amazon

Provedor de dados	Cobertura geográfica	Cobertura de atributos	Região da AWS
			Location está disponível.
Grab	<a href="#">Sudeste Asiático</a>	Mapas, locais, rotas	Ásia-Pacífico (Singapura), ap-southeast-1 , somente.
HERE	Global	Mapas, locais, rotas	<a href="#">Todas as regiões</a> em que o Amazon Location está disponível.
Open Data	Global	Mapas	<a href="#">Todas as regiões</a> em que o Amazon Location está disponível.

Para obter mais informações sobre os atributos específicos de cada provedor de dados, consulte [Recursos por provedor de dados](#).

Cada provedor de dados coleta e produz dados de diferentes maneiras. Você pode aprender mais sobre suas áreas de cobertura nos seguintes tópicos:

- [Cobertura: Esri](#)
- [Cobertura: Grab](#)
- [Cobertura: HERE](#)
- [Cobertura: Open Data](#)

Se você encontrar um problema com os dados e quiser relatar um erro ao provedor de dados, consulte os tópicos a seguir:

- [Relatório de erros para a Esri](#)
- [Relatório de erros para GrabMaps dados](#)
- [Relatório de erros para HERE](#)

- [Relatórios de erros e contribuição para Open Data](#)

## Estilos de mapa

Cada provedor de dados fornece um conjunto de estilos de mapa para renderizar os dados do mapa que fornecem. Por exemplo, um estilo pode incluir imagens de satélite ou pode ser otimizado para mostrar as vias para navegação. Você encontrará a lista e os exemplos dos estilos de cada provedor nos tópicos a seguir.

- [Estilos de mapa da Esri](#)
- [Estilos de mapas da Grab](#)
- [HEREestilos de mapa](#)
- [Estilos de mapas Open Data](#)

## Mais informações sobre cada provedor de dados

Os links a seguir fornecem informações sobre a cada provedor de dados.

- [Esri](#)
- [GrabMaps](#)
- [HERETecnologias](#)
- [Open Data](#)

## Esri

O Amazon Location Service usa os serviços de localização da Esri para ajudar AWS os clientes a usar mapas, geocodificar e calcular rotas de forma eficaz. Os serviços de localização da Esri são criados com dados de alta qualidade, confiáveis e de ready-to-use localização, organizados por equipes especializadas de cartógrafos, geógrafos e demógrafos.

Para obter informações adicionais, consulte [Esri](#) no Guia do desenvolvedor do Amazon Location Service.

### Tópicos

- [Estilos de mapa da Esri](#)
- [Cobertura: Esri](#)

- [Termos de uso e atribuição de dados: Esri](#)
- [Relatório de erros para a Esri](#)

## Estilos de mapa da Esri

O Amazon Location Service é compatível com os seguintes estilos de mapa da Esri quando [cria um recurso de mapa](#).

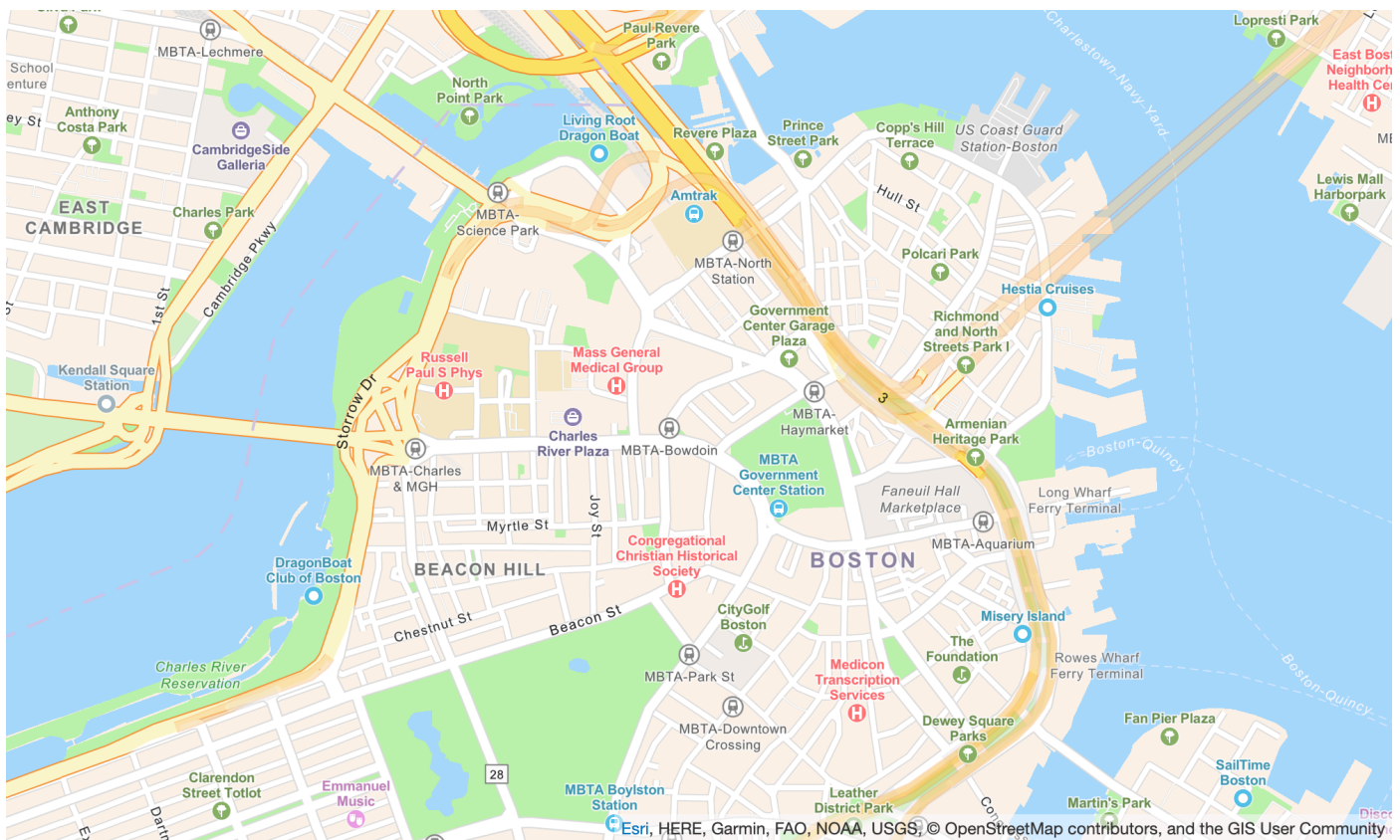
### Note

Estilos de mapa da Esri não estão listados nesta seção não são compatíveis.

Os estilos vetoriais da Esri suportam alternativas [Visões políticas](#).

## Esri Navigation

### Navegação da Esri



Nome do estilo do mapa: VectorEsriNavigation

Este mapa fornece um mapa de base detalhado do mundo no estilo de um mapa de navegação personalizado e projetado para uso durante o dia em dispositivos móveis.

Este mapa de ruas abrangente inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas. A camada de mosaico vetorial neste mapa é construída usando as mesmas fontes de dados usadas para o World Street Map e outros mapas de base da Esri. Ative a POI camada configurando-a [CustomLayers](#) para aproveitar os dados de locais adicionais.

Para mais informações, consulte [Esri World Navigation](#) no website da Esri.

#### Note

O mapa `VectorEsriNavigation` ilustrado acima tem a camada POI ativada.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Arial Italic
- Arial Regular
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regular

## Esri Imagery

### Esri Imagery

Nome do estilo do mapa: `RasterEsriImagery`

Este mapa fornece imagens aéreas e de satélite de um metro ou menos em várias partes do mundo e imagens de satélite de menor resolução em todo o mundo.

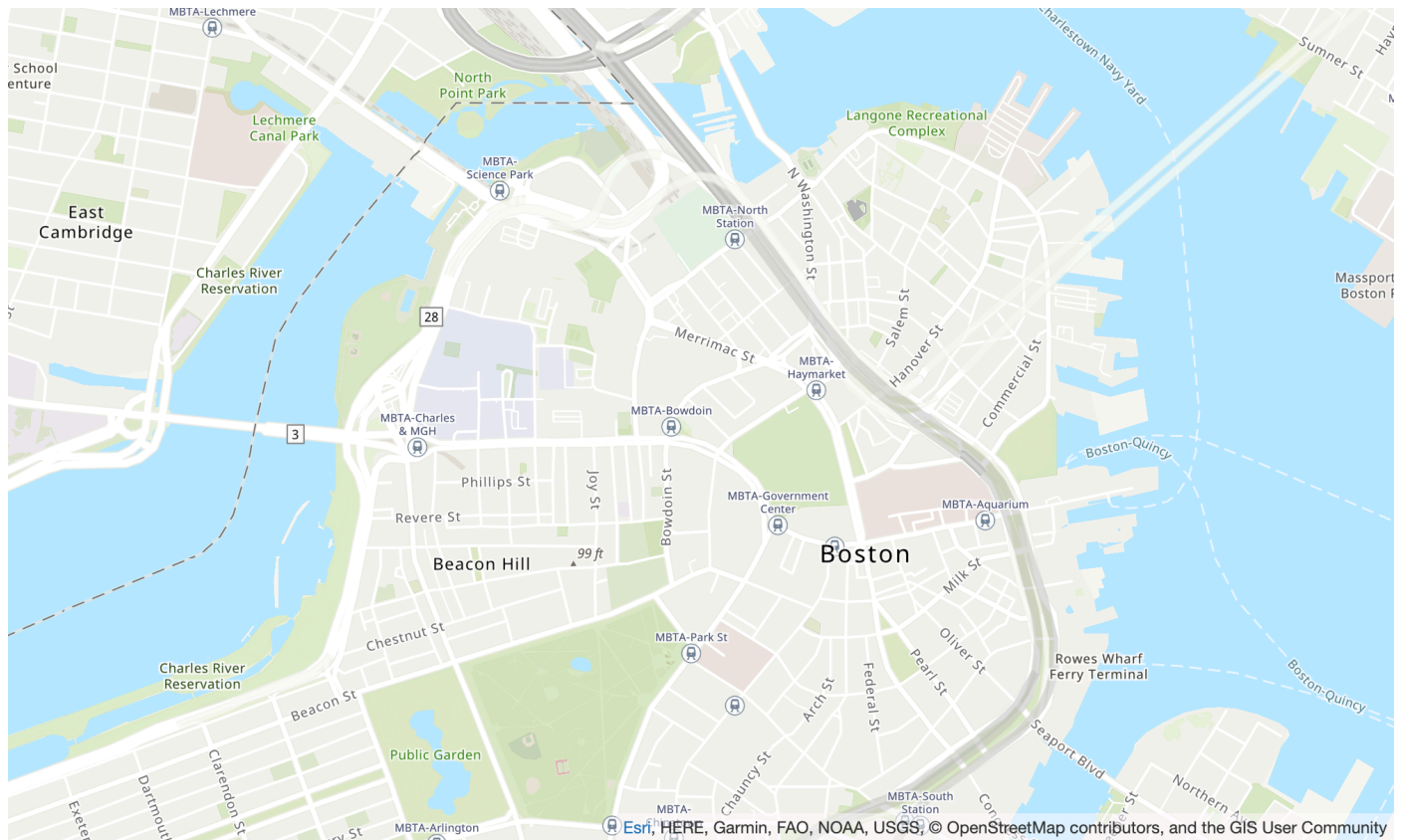
O mapa inclui imagens de 15 milhões em pequenas e médias escalas (~ 1:591 M até ~ 1:72 k) e 2,5 milhões de SPOT imagens (~ 1:288 k a ~ 1:72 k) para o mundo. O mapa apresenta

imagens com resolução de 0,5 m no território continental dos Estados Unidos e partes da Europa Ocidental da Maxar. Este mapa apresenta imagens adicionais do submeter Maxar em muitas partes do mundo. Em outras partes do mundo, a comunidade de GIS usuários contribuiu com imagens em diferentes resoluções. Em comunidades selecionadas, imagens de altíssima resolução (até 0,03m) estão disponíveis na escala de ~1:280.

Para obter mais informações, consulte [Esri World Imagery](#) no site da Esri.

## Esri Light

### Esri Light



Nome do estilo do mapa: VectorEsriTopographic

fornece um mapa de base detalhado do mundo no estilo clássico de mapa da Esri. Isso inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas.

Este mapa base é compilado a partir de uma variedade de fontes confiáveis de vários provedores de dados, incluindo o Serviço Geológico dos EUA (USGS), a Agência de Proteção Ambiental dos EUA (EPA), o Serviço Nacional de Parques dos EUA (NPS), a Organização das Nações Unidas para Alimentação e Agricultura (FAO), o Departamento de Recursos Naturais do Canadá (NRCan)



e a Esri. EPA HERE Os dados de áreas selecionadas são provenientes de OpenStreetMap colaboradores. Além disso, os dados são fornecidos pela GIS comunidade.

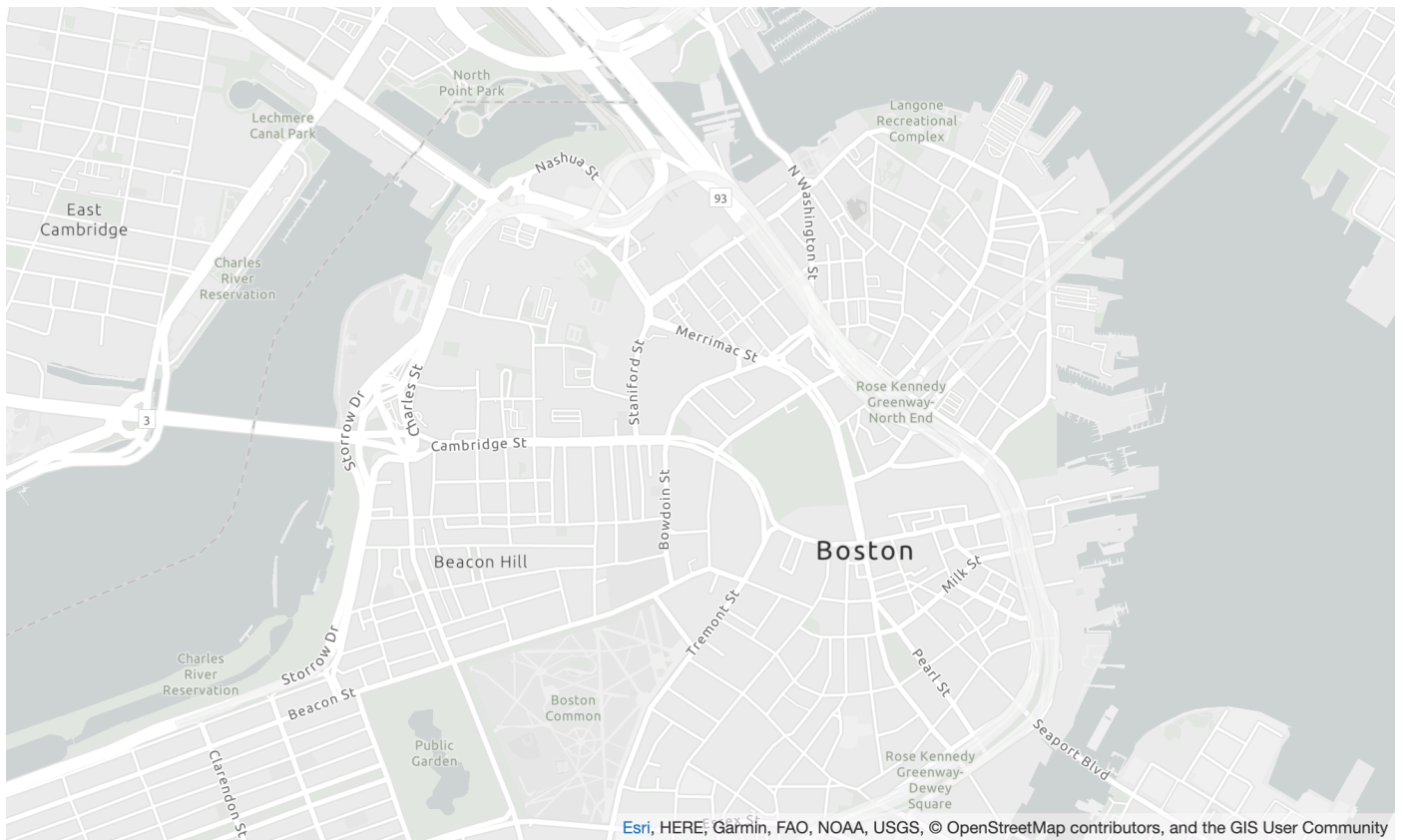
## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Noto Sans Italic
- Noto Sans Regular
- Noto Sans Bold
- Noto Serif Regular
- Roboto Condensed Light Italic

## Esri Light Gray Canvas

### Esri Light Gray Canvas



Nome do estilo do mapa: `VectorEsriLightGrayCanvas`

este mapa fornece um mapa de base detalhado do mundo simbolizado com um estilo de fundo neutro, em cinza claro, com cores mínimas, rótulos e funções projetadas para destacar o seu conteúdo temático.

Essa camada de mosaico vetorial é construída usando as mesmas fontes de dados usadas para o Light Grey Canvas e outros mapas-base da Esri. O mapa inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas.

Para obter mais informações, consulte [Esri Light Grey Canvas](#) no site da Esri.

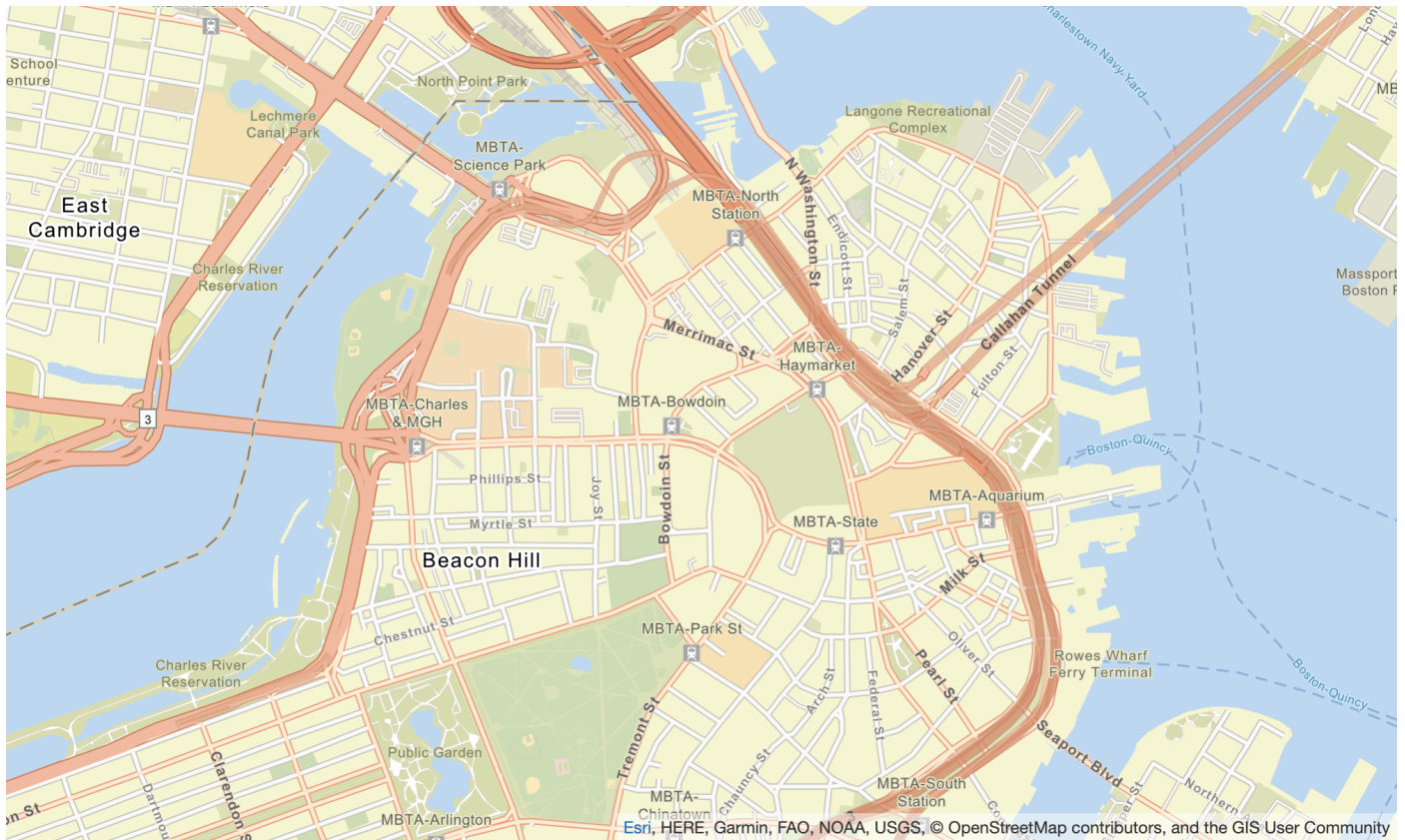
## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Ubuntu Italic
- Ubuntu Regular
- Ubuntu Light
- Ubuntu Bold

## Esri Street Map

Mapa de ruas da Esri



## Nome do estilo do mapa: VectorEsriStreets

Este mapa fornece um mapa de base detalhado do mundo no estilo de um mapa de navegação personalizado e projetado para uso durante o dia em dispositivos móveis.

Este mapa de ruas abrangente inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas. Também inclui um conjunto mais rico de lugares, como lojas, serviços, restaurantes, atrações e outros pontos de interesse. A camada de mosaico vetorial neste mapa é construída usando as mesmas fontes de dados usadas para o World Street Map e outros mapas de base da Esri.

Para obter mais informações, consulte [Esri World Street](#) no site da Esri.

## Fontes

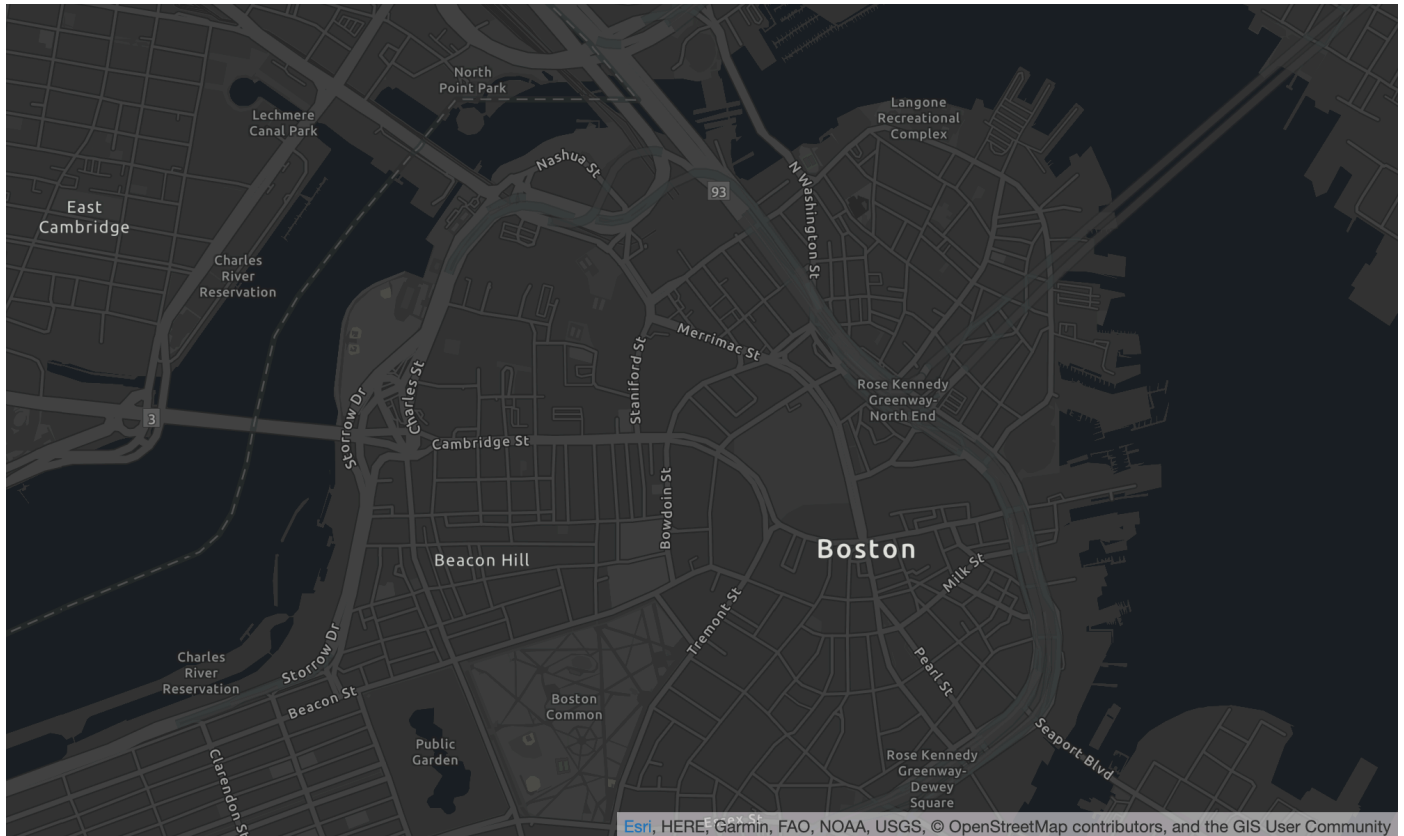
O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Arial Italic

- Arial Regular
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regular

## Esri Dark Gray Canvas

### Esri Dark Gray Canvas



Nome do estilo do mapa: `VectorEsriDarkGrayCanvas`

este mapa fornece um mapa de base vetorial e detalhado do mundo em fundo cinza escuro, fundo neutro com cores mínimas, rótulos e recursos, projetado para destacar o conteúdo temático.

O mapa inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas. As camadas de mosaico vetorial neste mapa são construídas usando as mesmas fontes de dados usadas para o mapa raster Dark Grey Canvas e outros mapas de base da Esri.

Para obter mais informações, consulte [Esri Dark Grey Canvas](#) no site da Esri.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Ubuntu Medium Italic
- Ubuntu Medium
- Ubuntu Italic
- Ubuntu Regular
- Ubuntu Bold

## Cobertura: Esri

Você pode usar a Esri como provedor de dados para dar suporte a consultas para geocodificação, geocodificação reversa e pesquisas ao [criar um recurso de índice de local](#) ou para dar suporte a consultas para calcular uma rota ao [criar um recurso de calculadora de rotas](#).

A Esri fornece diferentes níveis de qualidade de dados em diferentes regiões do mundo. Para obter mais informações sobre a cobertura em sua região de interesse, consulte:

- [Detalhes da Esri sobre cobertura de geocodificação](#)
- [Detalhes da Esri sobre redes de ruas e cobertura de tráfego](#)

## Termos de uso e atribuição de dados: Esri

Antes de usar os dados da Esri, certifique-se de que você pode cumprir todos os requisitos legais aplicáveis, incluindo os termos de licença aplicáveis à Esri e AWS.

Para obter mais informações sobre os AWS requisitos, consulte os [Termos AWS de Serviço](#).

Para obter informações sobre as diretrizes de atribuição da Esri, consulte [Atribuições de Dados e Termos de Uso](#) da Esri.

## Relatório de erros para a Esri

Se você encontrar um problema com os dados e quiser relatar erros e discrepâncias à Esri, siga o artigo de suporte técnico da Esri sobre [Como: Fornecer feedback sobre mapas-base e geocodificação](#).

## GrabMaps

A Grab é a maior empresa de entregas do Sudeste Asiático, com milhões de motoristas parceiros e clientes. Sua subsidiária, [GrabMaps](#), cria dados de up-to-date mapeamento nesses países/regiões para seu próprio uso e em outros. O Amazon Location Service usa GrabMaps serviços de localização para ajudar AWS os clientes a usar mapas, geocodificar e calcular rotas de forma eficaz. GrabMaps'os serviços de localização são criados para fornecer dados de alta qualidade, confiáveis e de ready-to-use localização, especificamente para países do sudeste asiático.

Para obter informações sobre recursos adicionais, consulte os provedores [GrabMaps](#) de dados do Amazon Location Service.

### Important

O Grab só fornece mapas para o sudeste asiático e só está disponível na região da Ásia-Pacífico (Singapura) (ap-southeast-1). Para obter mais informações, consulte [Países/regiões e área de cobertura](#).

## Tópicos

- [Estilos de mapas da Grab](#)
- [Cobertura: Grab](#)
- [Países/regiões e área de cobertura](#)
- [Termos de uso e atribuição de dados: Grab](#)
- [Relatório de erros para GrabMaps dados](#)

## Estilos de mapas da Grab

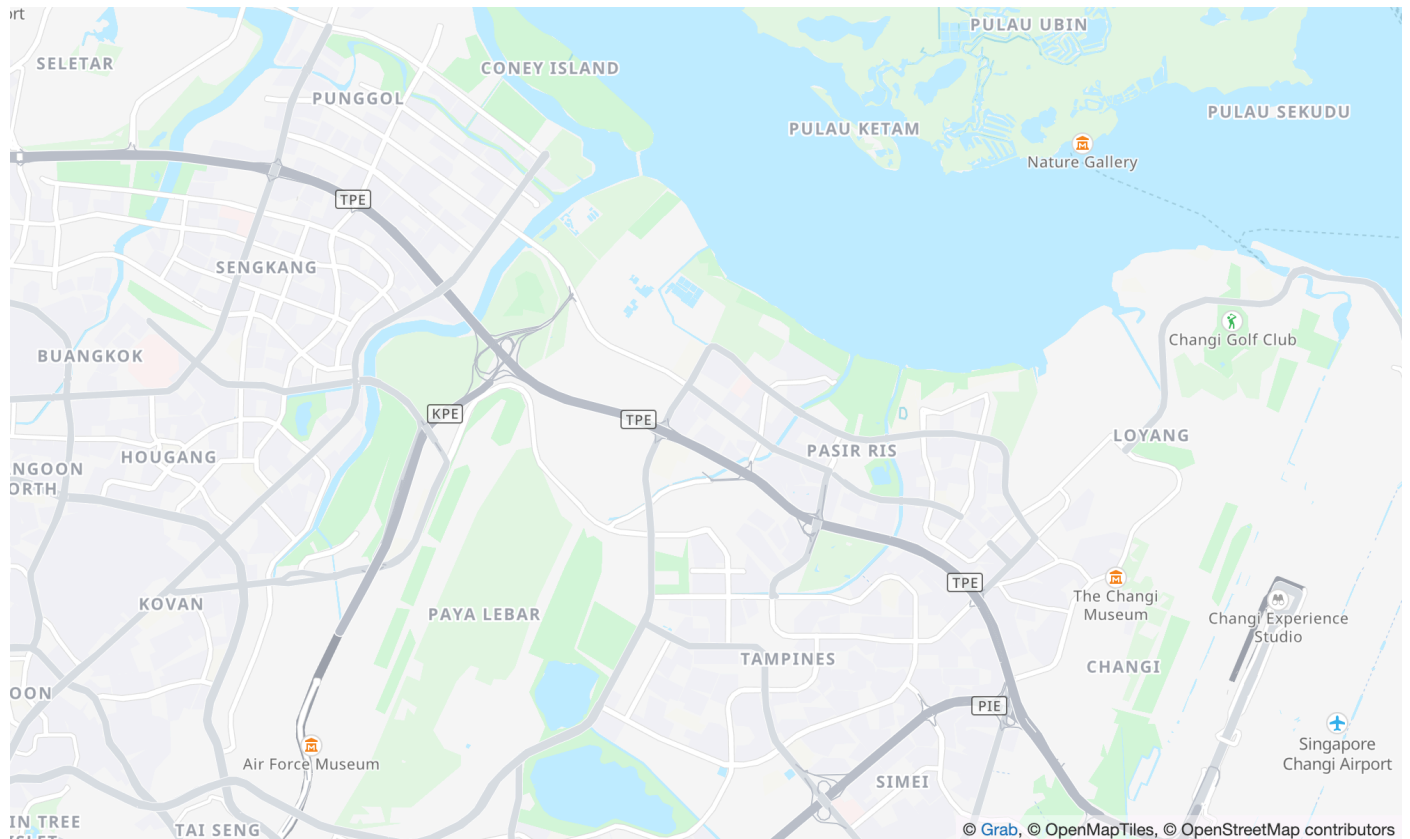
O Amazon Location Service é compatível com os seguintes estilos de mapas da Grab ao [criar um recurso de mapa](#):

**Note**

Estilos de mapa da Grab que não estão listados nesta seção atualmente não são compatíveis.

## Grab Standard Light Map

### Mapa Grab Standard Light



Nome do estilo do mapa: `VectorGrabStandardLight`

estilo de mapa de base da Grab com detalhes em cores do uso do terreno, nomes de áreas, vias, pontos de referência e pontos de interesse, abrangendo o Sudeste Asiático.

### Fontes

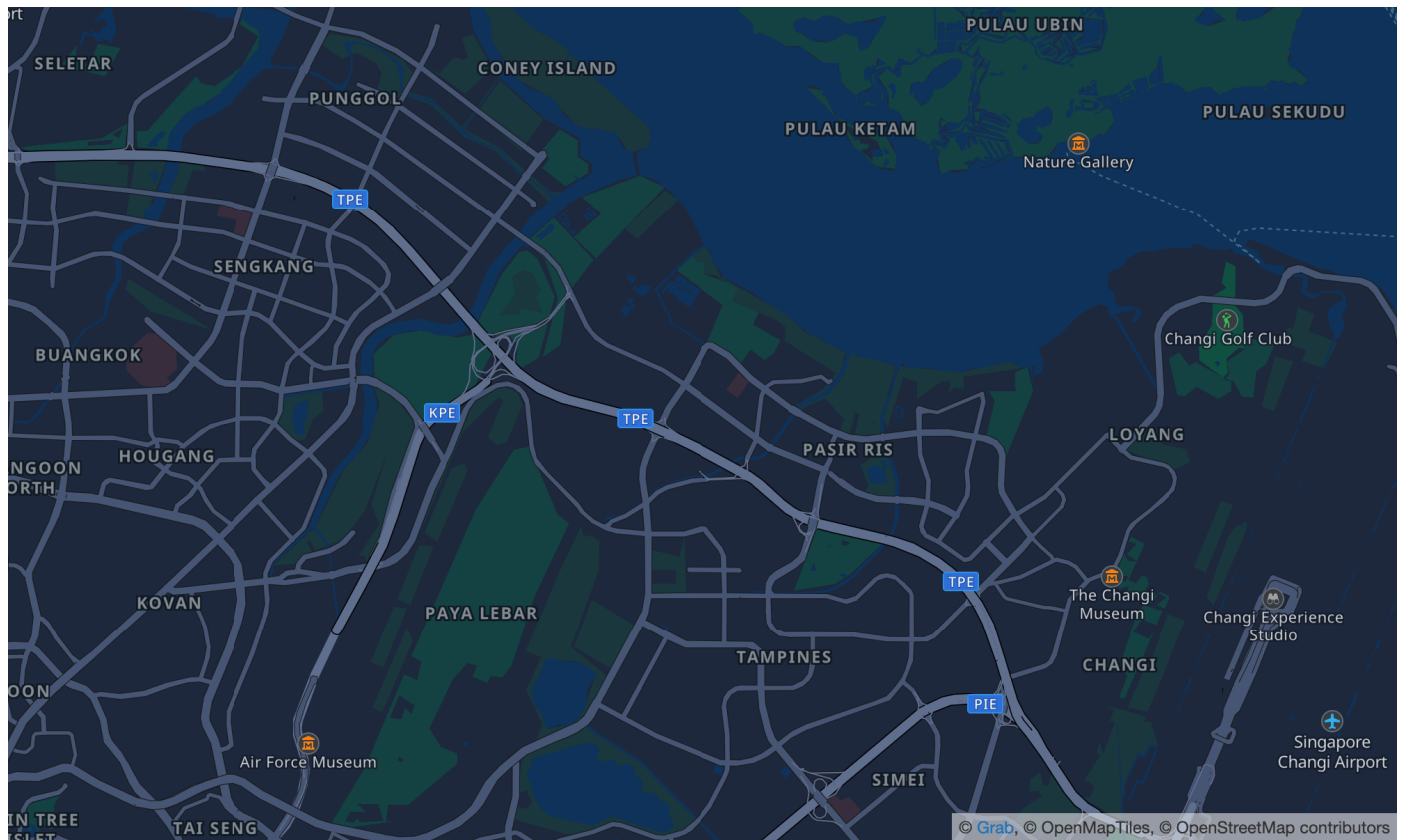
O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Noto Sans Regular

- Noto Sans Medium
- Noto Sans Bold

## Grab Standard Dark Map

### Standard Dark Map da Grab



Nome do estilo do mapa: VectorGrabStandardDark

estilo escuro do mapa de base da Grab, com detalhes em cores do uso do terreno, nomes de áreas, vias, pontos de referência e pontos de interesse abrangendo o Sudeste Asiático.

### Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

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



## Cobertura: Grab

Você pode usar a Grab como provedora de dados para dar suporte a consultas de geocodificação, geocodificação reversa e pesquisas ao [criar um recurso de índice de local](#), ou para dar suporte a consultas para calcular uma rota ao [criar um recurso de calculadora de rotas](#).

### Países/regiões e área de cobertura

A Grab só fornece mapas para o sudeste asiático e só está disponível na região da Ásia-Pacífico (Singapura) (ap-southeast-1).

A Grab fornece dados detalhados dos seguintes países/regiões:

- Malásia
- Filipinas
- Tailândia
- Cingapura
- Vietnã
- Indonésia
- Mianmar
- Camboja

#### Note

Fora dessas áreas, os recursos do Amazon Location Service criados com a Grab como provedora de dados não fornecerão nenhum resultado. Isto inclui resultados de pesquisa ou rotas.

Os mapas da Grab estão dentro dos seguintes limites:

- Sul: latitude -21,943045533438166
- Oeste: longitude 90,0
- Norte: latitude 31,952162238024968
- Leste: longitude 146,25

Para níveis de zoom de 1 a 4, a Grab inclui cobertura global. Para níveis de zoom 5 e abaixo, os blocos do mapa são fornecidos somente dentro dessa caixa delimitada.

#### Note

Fora dessa caixa delimitada, os recursos de mapa do Amazon Location Service criados com a Grab como provedora de dados não retornarão mosaicos de mapas. Para evitar ver erros 404 em seu aplicativo, você pode limitar o mapa com uma caixa delimitadora, conforme descrito em [Definindo extensões para um mapa usando MapLibre](#).

Modos de viagem por roteamento da Grab.

Para roteamento, a Grab fornece roteamento de carros e motocicletas para todos os países/regiões listados anteriormente.

A Grab não oferece suporte ao roteamento de caminhões.

Para ciclovias e percursos de pedestres, a Grab oferece suporte às seguintes cidades:

- Cingapura
- Jacarta
- Manila
- Vale do Klang
- Bangkok
- Cidade de Ho Chi Minh
- Hanói

## Termos de uso e atribuição de dados: Grab

Ao usar os dados do Grab, você deve cumprir todos os requisitos legais aplicáveis, incluindo os termos de licença aplicáveis ao Grab AWS e.

Para obter mais informações sobre os AWS requisitos, consulte os [Termos AWS de Serviço](#).

Para obter informações sobre GrabMaps “diretrizes de atribuição”, consulte a Seção 9.23 das [Atribuições de Dados](#) e Termos de Uso da Grab.

## Relatório de erros para GrabMaps dados

Se você encontrar um problema com os dados e quiser relatar erros ou discrepâncias, [entre em contato com o AWS suporte técnico](#). GrabMaps

## HERETecnologias

O Amazon Location Service usa os serviços de localização HERE da Technologies para ajudar AWS os clientes a usar mapas, geocodificar e calcular rotas de forma eficaz. HEREOs dados de localização da oferecem uma plataforma centrada na localização que é aberta, segura e privada. Ao selecionar dados de HERE localização, você está selecionando dados precisos, atualizados e robustos que são implantados nativamente na AWS nuvem.

Para obter informações adicionais sobre capacidades, consulte os [HERE](#) provedores de dados do Amazon Location Service.

### Tópicos

- [HEREestilos de mapa](#)
- [Cobertura: HERE](#)
- [Termos de uso e atribuição de dados: HERE](#)
- [Relatório de erros para HERE](#)

## HEREestilos de mapa

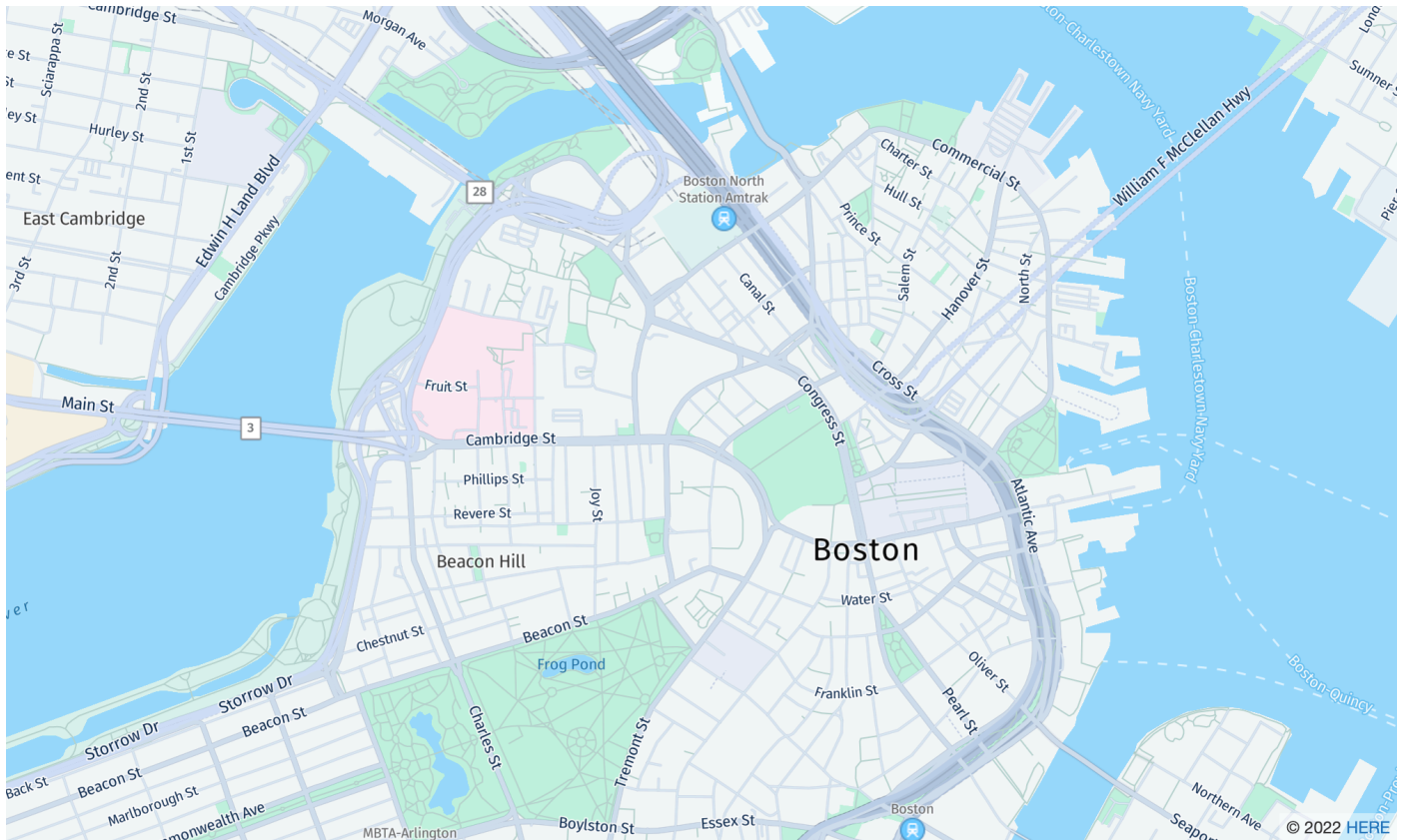
O Amazon Location Service suporta os seguintes estilos de HERE mapa ao [criar um recurso de mapa](#):

### Note

HEREestilos de mapa que não estão listados nesta seção atualmente não são suportados.

## HERE Explore

HEREExplorar



Nome do estilo do mapa: VectorHereExplore

HEREExplorar

Um mapa de base, detalhado e neutro do mundo. O mapa inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas. Inclui também um mapa totalmente projetado do Japão.

Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Normal
- Noto Sans CJK JP Bold

## HERE Imagery

### HEREImagens



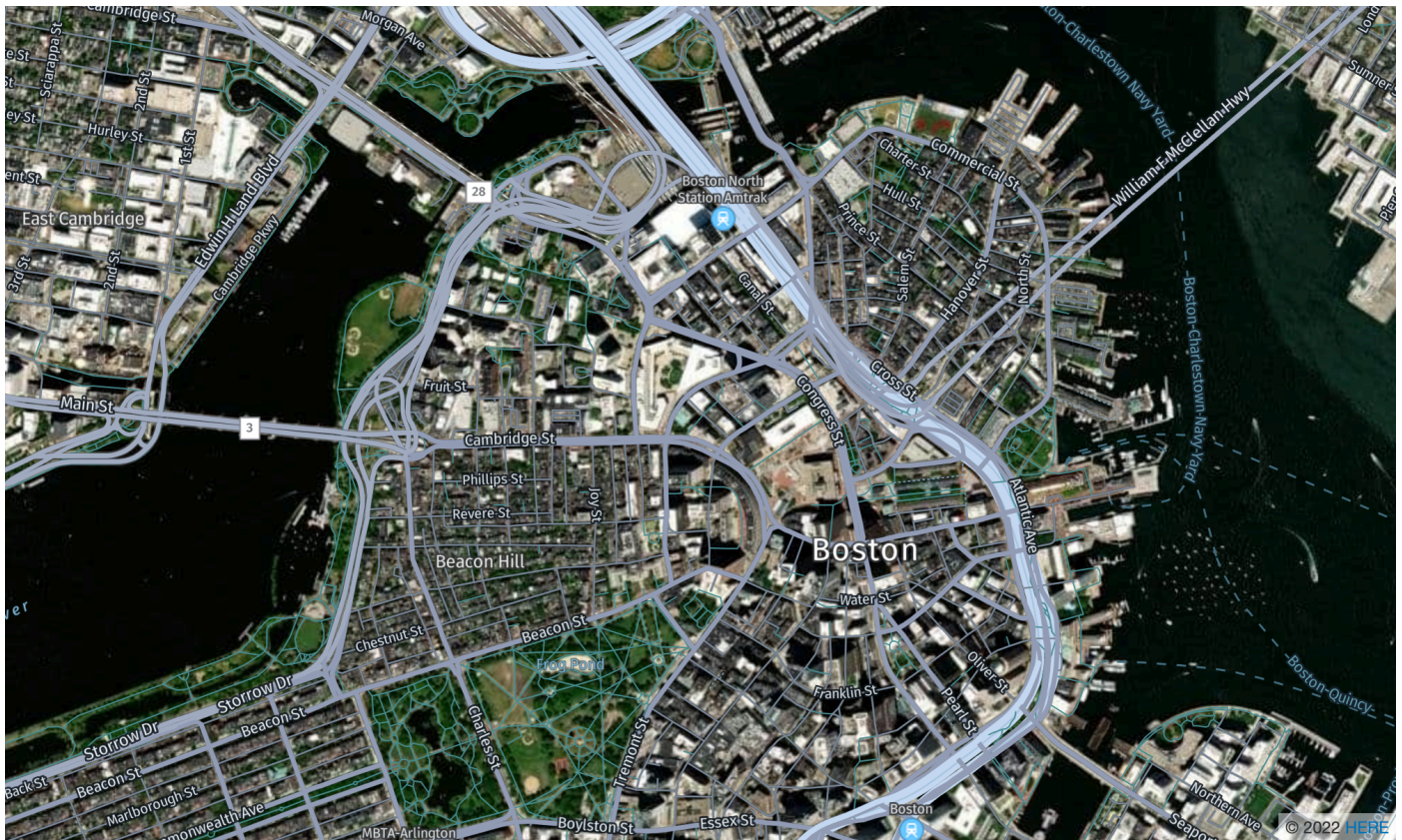
Nome do estilo do mapa: `RasterHereExploreSatellite`

### HEREImagens

HEREAs imagens fornecem imagens de satélite de alta resolução com cobertura global.

### HERE Hybrid

### HEREHíbrido



Nome do estilo do mapa: HybridHereExploreSatellite

HEREHíbrido

HEREO estilo híbrido exibe a rede viária, os nomes das ruas e os rótulos das cidades em imagens de satélite. Este estilo sobrepõe dois blocos de mapa: a imagem de satélite (bloco raster) no fundo e a rede rodoviária e os rótulos (bloco vetorial) na parte superior. Esse estilo recuperará automaticamente os blocos rasterizados e os vetoriais necessários para renderizar o mapa.

#### Note

Os estilos híbridos usam blocos vetoriais e rasterizados ao renderizar o mapa que você vê. Isso significa que mais blocos são recuperados do que quando se usa apenas blocos vetoriais ou rasterizados. Suas cobranças incluirão todos os blocos recuperados.

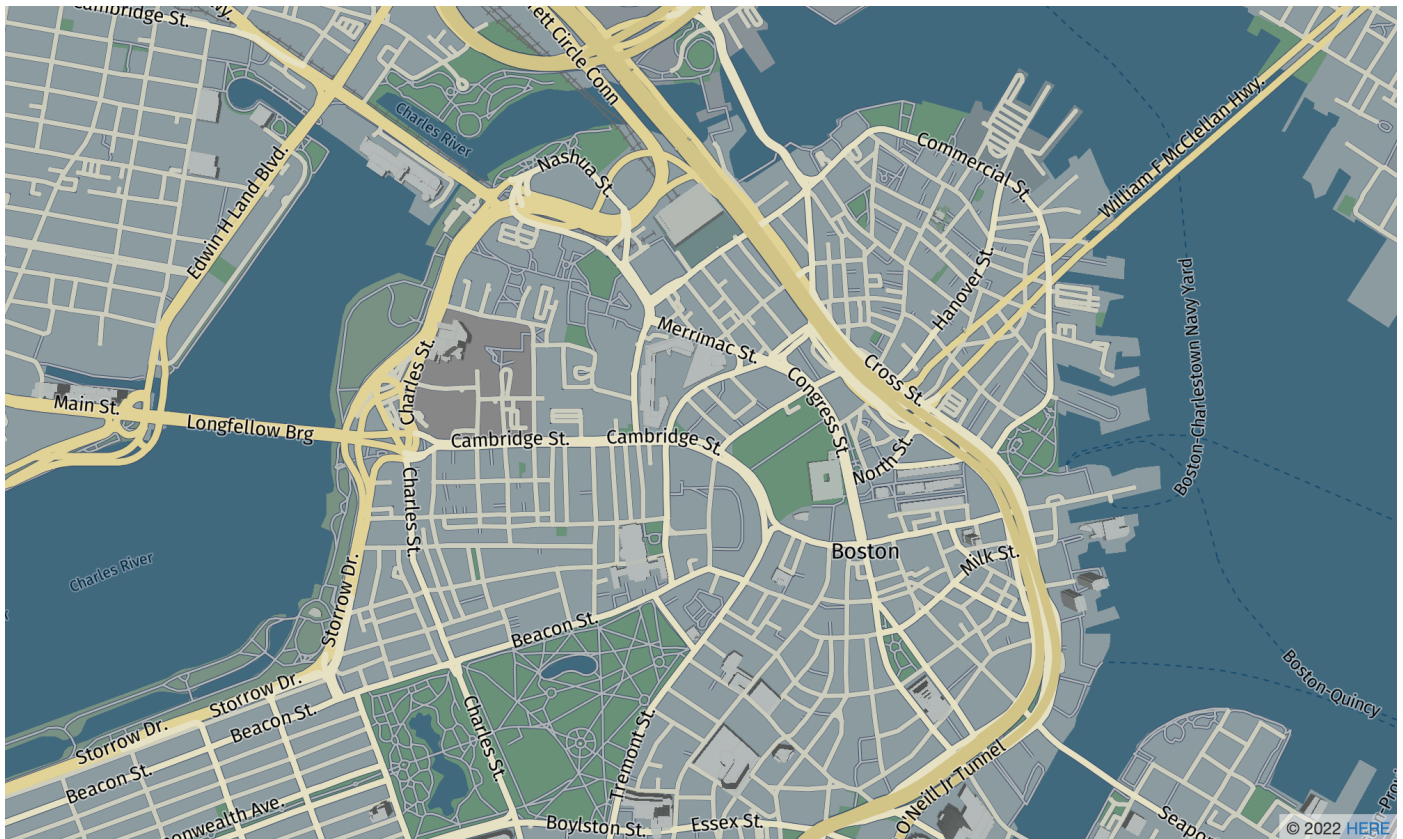
Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Normal
- Noto Sans CJK JP Bold

HERE Contrast (Berlin)

HEREContrast (Berlim)



Nome do estilo do mapa: VectorHereContrast

HEREContrast (Berlim)

Um mapa base detalhado do mundo que combina renderização 3D e 2D. O mapa de alto contraste inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Fira GO Regular
- Fira GO Bold

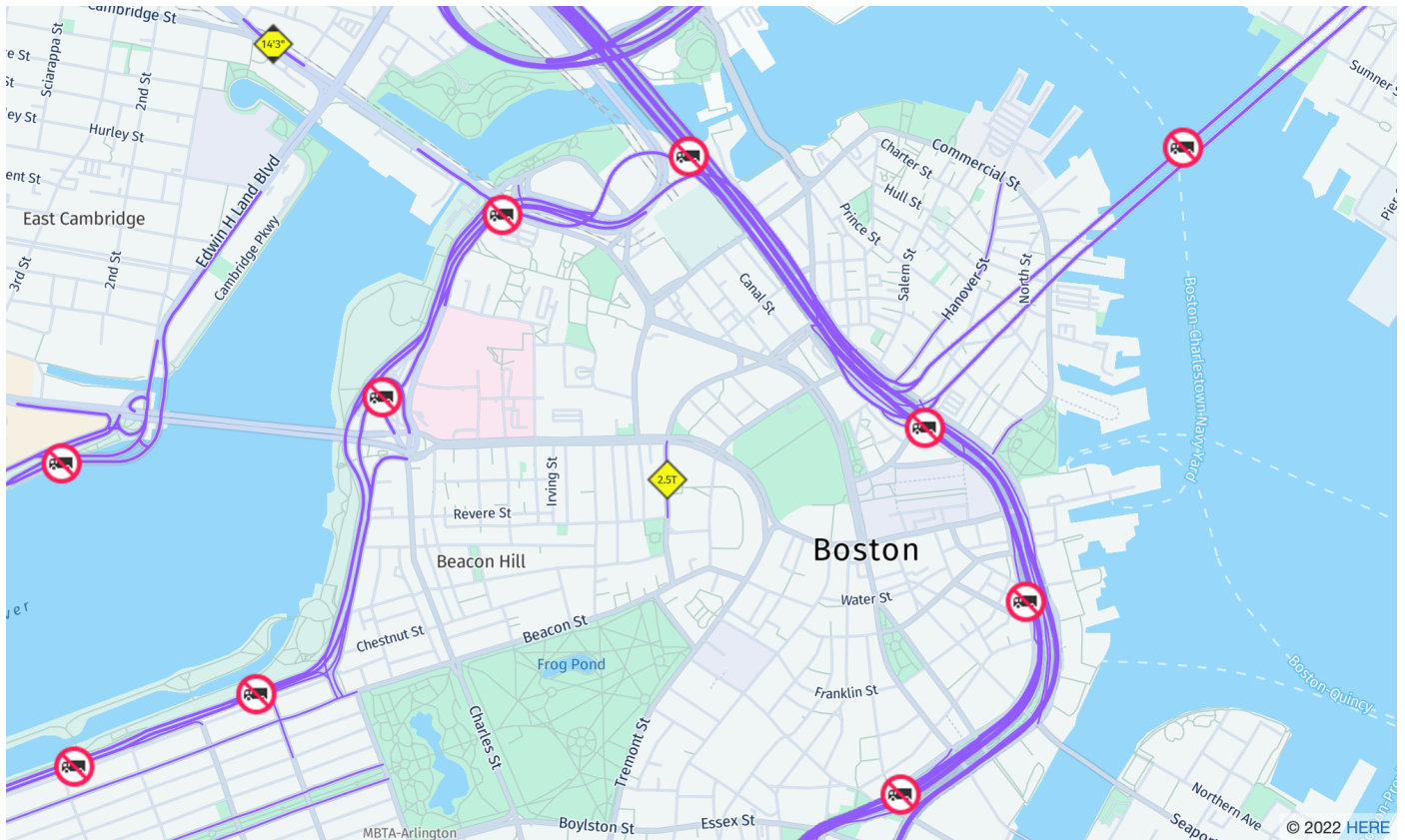
### Note

Esse estilo foi renomeado de `VectorHereBerlin` (mapas de HERE Berlim). `VectorHereBerlin` está obsoleto, mas continuará funcionando em aplicativos que o usam.

## HERE Explore Truck

HEREExplore o caminhão





Nome do estilo do mapa: VectorHereExploreTruck

HEREExplore o caminhão

Um mapa de base, detalhado e neutro do mundo. O mapa de ruas se baseia no estilo HERE Explore e destaca as restrições e os atributos da trilha (incluindo largura, altura e HAZMAT) com símbolos e ícones, para apoiar casos de uso em transporte e logística.

Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

- Fira GO Italic
- Fira GO Regular
- Fira GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Normal
- Noto Sans CJK JP Bold

Para obter informações adicionais sobre a qualidade dos dados do mapa em diferentes regiões do mundo, consulte a [cobertura do HERE mapa](#).

## Cobertura: HERE

[Você pode usar HERE como provedor de dados para dar suporte a consultas de geocodificação, geocodificação reversa e pesquisas ao criar um recurso de índice de local ou para apoiar consultas para calcular uma rota ao criar um recurso de calculadora de rotas.](#)

HERE fornece diferentes níveis de qualidade de dados em diferentes regiões do mundo. Para obter mais informações sobre a cobertura na sua região de interesse, consulte os itens a seguir:

- [HEREcobertura de geocodificação](#)
- [HEREcobertura de roteamento de automóveis](#)
- [HEREcobertura de roteamento de caminhões](#)

## Termos de uso e atribuição de dados: HERE

Antes de usar HERE os dados, verifique se você pode cumprir todos os requisitos legais aplicáveis, incluindo os termos de licença aplicáveis a HERE AWS e. Devido às limitações de licenciamento, você não pode usar HERE para armazenar resultados de geocodificação para locais no Japão.

Para obter informações sobre os AWS requisitos, consulte os [Termos AWS de Serviço](#).

Para obter informações adicionais sobre as diretrizes HERE de atribuição, consulte a Seção 2 dos [Termos de Fornecedores de HERE Tecnologias Aplicáveis à Localização](#) e Outros Conteúdos.

## Relatório de erros para HERE

Para relatar erros e discrepâncias no mapaHERE, acesse <https://www.here.com/contact> e escolha Relatar um erro de mapa.

## Open Data

O Amazon Location Service fornece acesso a dados de mapas de código aberto por meio do provedor Open Data. O Open Data fornece mapas base globais criados a partir da [distribuição de mapas Daylight](#) de [OpenStreetMap \(OSM\)](#), [Natural Earth](#) e outras fontes de dados abertas. Os mapas fornecidos foram projetados para oferecer suporte a diferentes aplicativos e casos de uso, incluindo logística, entrega e visualização de dados em ambientes web e móveis. Com mais de um

milhão de criadores de mapas, a OSM comunidade atualiza centenas de milhares de recursos por dia. O Amazon Location Service incorpora regularmente essas edições.

Para obter informações adicionais sobre a funcionalidade, consulte [Open Data](#) nos provedores de dados do Amazon Location Service.

## Tópicos

- [Estilos de mapas Open Data](#)
- [Cobertura: Open Data](#)
- [Termos de uso e atribuição de dados: Open Data](#)
- [Relatórios de erros e contribuição para Open Data](#)

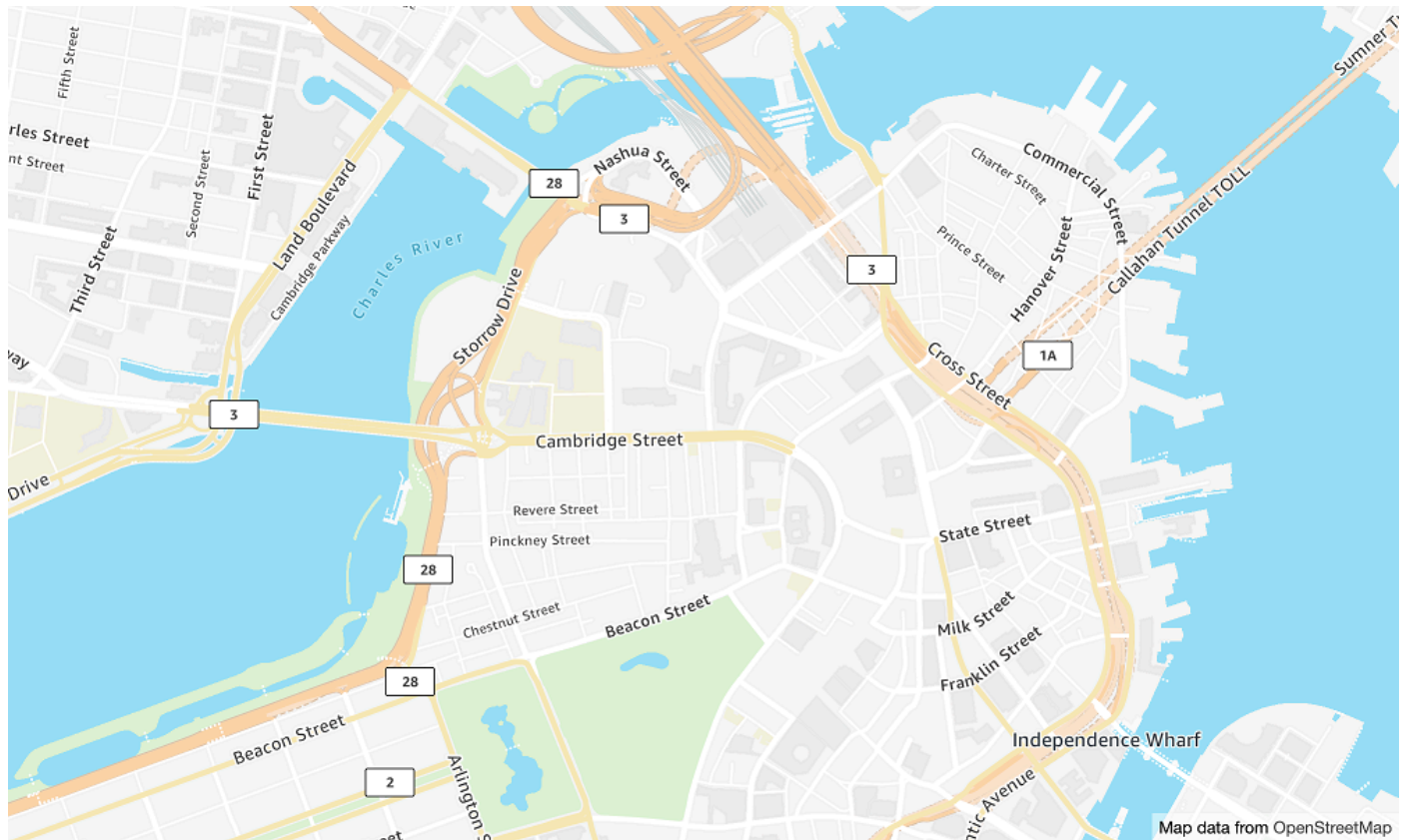
## Estilos de mapas Open Data

O Amazon Location Service é compatível com os seguintes estilos de mapas ao [criar um recurso de mapa](#):

Os estilos de mapa Open Data oferecem suporte alternativo [Visões políticas](#).

### Open Data Standard Light

Padrão claro de dados abertos



Nome do estilo do mapa: `VectorOpenDataStandardLight`

fornece um mapa de base detalhado do mundo em um estilo de mapa leve e adequado para uso em sites e aplicativos móveis. Isso inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas.


Esse mapa base é baseado na [distribuição OSM do mapa Daylight compilada a partir dos contribuidores](#) OpenStreetMap (OSM). A OSM comunidade inclui mais de 1,8 milhão de colaboradores que atualizam mais de 500.000 recursos diariamente. O Amazon Location Service incorpora essas edições regularmente.

### Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

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

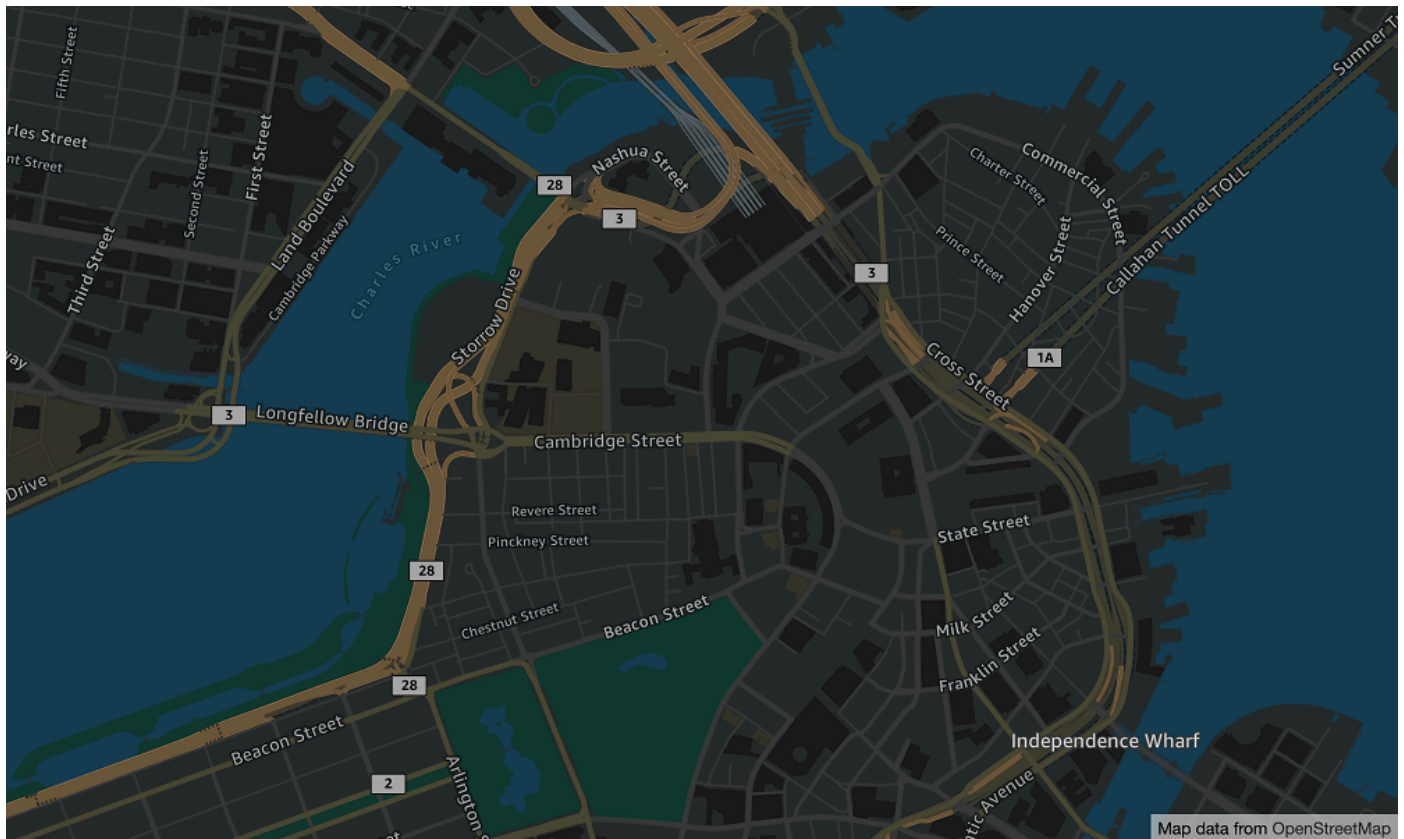
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold,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

 Note

As fontes usadas em `VectorOpenDataStandardLight` são fontes combinadas que utilizam Amazon Ember para a maioria dos glifos, menos Noto Sans para glifos não compatíveis com Amazon Ember.

## Open Data Standard Dark

Padrão escuro de dados abertos



Nome do estilo do mapa: VectorOpenDataStandardDark

este é um estilo de mapa com tema escuro que fornece um mapa de base detalhado do mundo, adequado para uso em sites e aplicações móveis. Isso inclui rodovias, vias principais, vias secundárias, ferrovias, recursos hídricos, cidades, parques, pontos de referência, áreas construídas e fronteiras administrativas.


Esse mapa base é baseado na [distribuição OSM do mapa Daylight compilada a partir dos contribuidores](#) OpenStreetMap (OSM). A OSM comunidade inclui mais de 1,8 milhão de colaboradores que atualizam mais de 500.000 recursos diariamente. O Amazon Location Service incorpora essas edições regularmente.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

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

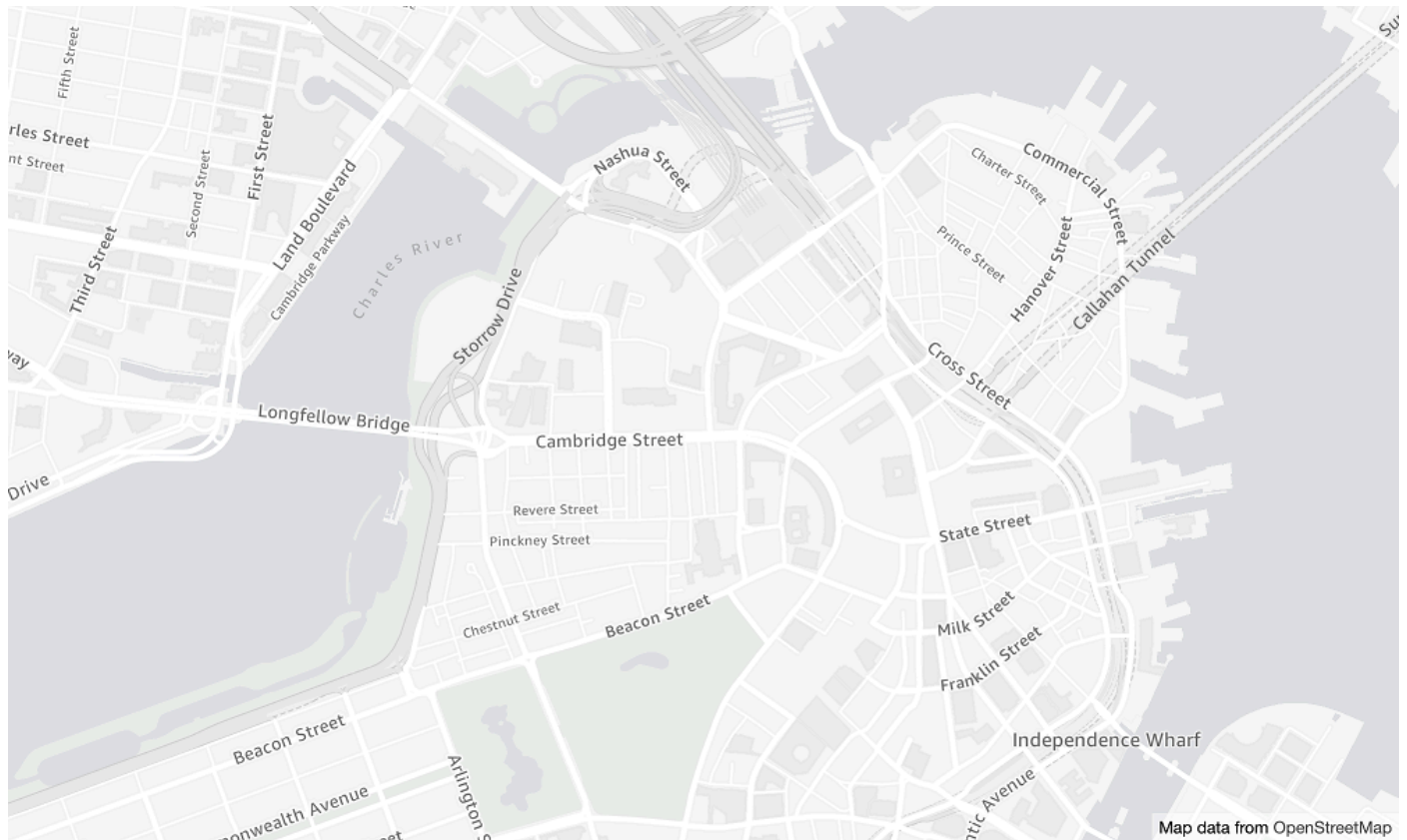
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold,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

 Note

As fontes usadas em `VectorOpenDataStandardDark` são fontes combinadas que utilizam Amazon Ember para a maioria dos glifos, menos Noto Sans para glifos não compatíveis com Amazon Ember.

## Open Data Visualization Light

Visualização clara de dados abertos



Nome do estilo do mapa: `VectorOpenDataVisualizationLight`

este é um estilo com tema claro, com cores suaves e menos recursos que ajudam a entender os dados sobrepostos.

Esse mapa base é baseado na [distribuição OSM do mapa Daylight compilada a partir dos contribuidores](#) OpenStreetMap (OSM). A OSM comunidade inclui mais de 1,8 milhão de colaboradores que atualizam mais de 500.000 recursos diariamente. O Amazon Location Service incorpora essas edições regularmente.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

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



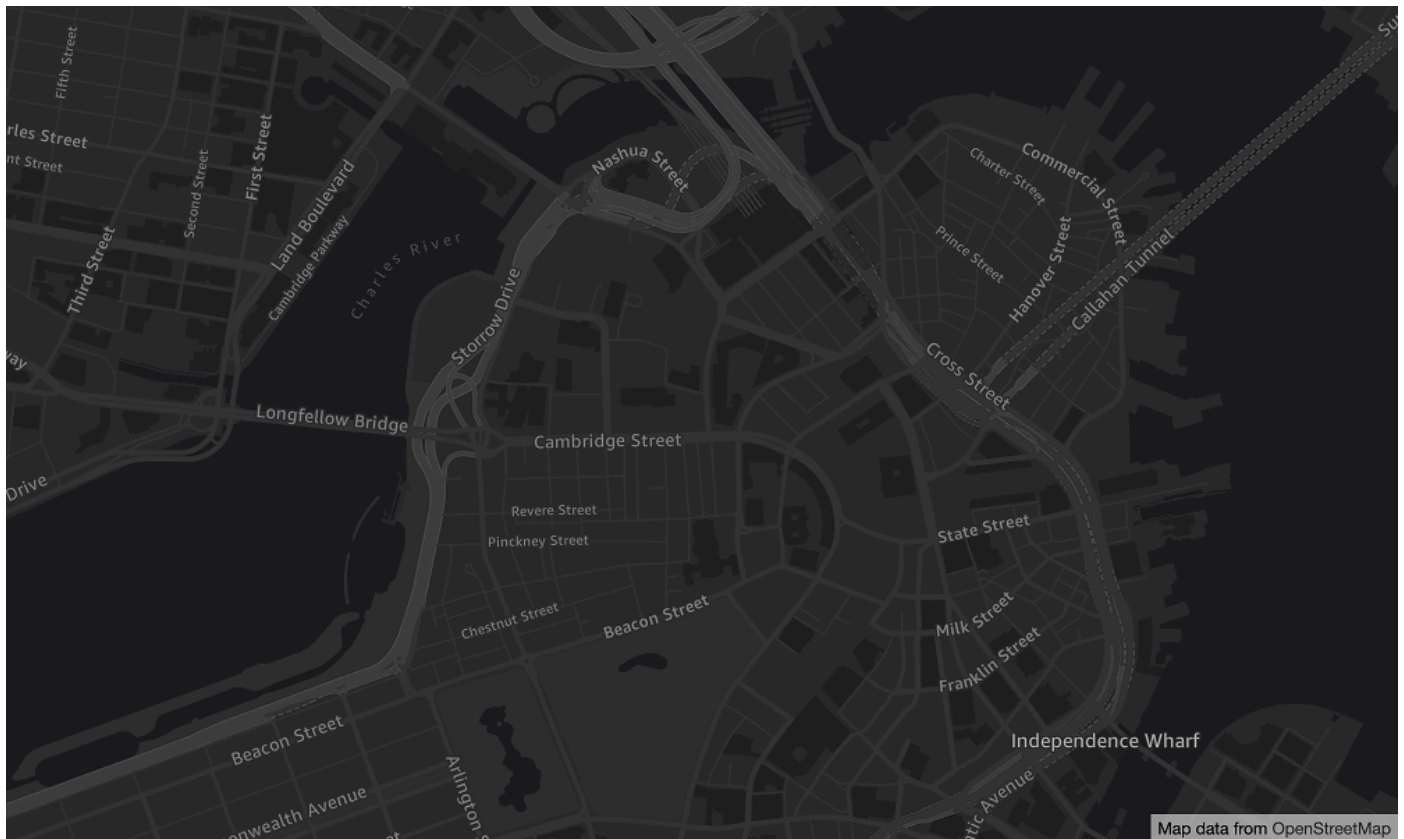
- Amazon Ember Regular Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
-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

 Note

As fontes usadas em `VectorOpenDataVisualizationLight` são fontes combinadas que utilizam Amazon Ember para a maioria dos glifos, menos Noto Sans para glifos não compatíveis com Amazon Ember.

## Open Data Visualization Dark

Visualização escura de dados abertos



Nome do estilo do mapa: `VectorOpenDataVisualizationDark`

este é um estilo com tema escuro com cores suaves e menos atributos que ajudam a entender os dados sobrepostos.

Esse mapa base é baseado na [distribuição OSM do mapa Daylight compilada a partir dos contribuidores](#) OpenStreetMap (OSM). A OSM comunidade inclui mais de 1,8 milhão de colaboradores que atualizam mais de 500.000 recursos diariamente. O Amazon Location Service incorpora essas edições regularmente.

## Fontes

O Amazon Location oferece fontes usando [GetMapGlyphs](#). Abaixo estão as pilhas de fontes disponíveis para este mapa:

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

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

#### Note

As fontes usadas em `VectorOpenDataVisualizationDark` são fontes combinadas que utilizam Amazon Ember para a maioria dos glifos, menos Noto Sans para glifos não compatíveis com Amazon Ember.

## Cobertura: Open Data

A Open Data inclui mapas com cobertura global para renderização com um [recurso de mapas do Amazon Location Service](#).

#### Note

Os Dados Abertos devem ser usados somente com os recursos de mapas do Amazon Location Service. Você não pode usar a Open Data como provedor de dados para dar suporte a consultas de geocodificação, geocodificação reversa e pesquisas, e nem para dar suporte a consultas para calcular uma rota.

## Termos de uso e atribuição de dados: Open Data

Antes de usar os Dados Abertos, certifique-se de cumprir todos os requisitos legais aplicáveis, incluindo os termos de licença aplicáveis aos Dados Abertos AWS e.

Para obter mais informações sobre os AWS requisitos, consulte os [Termos AWS de Serviço](#).

Para obter informações sobre as diretrizes de atribuição de dados abertos, consulte [Direitos autorais e Licença](#) e [OpenStreetMap Diretrizes OpenStreetMap de Licença/Atribuição](#).

## Relatórios de erros e contribuição para Open Data

OpenStreetMap (OSM) e Natural Earth são projetos de dados abertos voltados para a comunidade. Se encontrar algum problema com os dados, você poderá relatar os erros ou contribuir diretamente com correções ou sugestões.

- Para relatar um erro ou oferecer uma sugestão OSM, você pode criar uma nota no mapa. Trata-se de um comentário no mapa que ajuda os colaboradores a fazer correções no mesmo. Você cria notas por meio do [OpenStreetMap site](#). Para obter mais informações sobre notas, consulte [Notas](#) no OpenStreetMap wiki.
- Para obter mais informações sobre como contribuir diretamente para OpenStreetMap, incluindo adicionar locais e corrigir erros, consulte [Contribuir com dados do mapa](#) no OpenStreetMap wiki.
- Para enviar uma solicitação de correção de dados no Natural Earth, você pode enviá-la pelo [site do Natural Earth](#).

### Note

A correção de erros OpenStreetMap pode ocorrer rapidamente, no entanto, pode levar algum tempo para que as correções apareçam na distribuição do mapa Daylight dos OSM dados usados pelo provedor de dados abertos. O site [Daylight Map Distribution](#) fornece mais informações sobre o processo. Além disso, o Amazon Location Service atualiza os dados do mapa usados no Amazon Location Service a cada mês, aproximadamente.

## Recursos por provedor de dados

Esta seção descreve os atributos disponíveis no Amazon Location Service, categorizados por provedor de dados.

A tabela a seguir fornece uma visão geral de alto nível dos atributos.

Provedor de dados	Cobertura geográfica	Cobertura de atributos	Região da AWS
Esri	Global	Mapas, locais, rotas	<a href="#">Todas as regiões</a> em que o Amazon

Provedor de dados	Cobertura geográfica	Cobertura de atributos	Região da AWS
			Location está disponível.
Grab	<a href="#">Sudeste Asiático</a>	Mapas, locais, rotas	Ásia-Pacífico (Singapura), ap-southeast-1 , somente.
HERE	Global	Mapas, locais, rotas	<a href="#">Todas as regiões</a> em que o Amazon Location está disponível.
Open Data	Global	Mapas	<a href="#">Todas as regiões</a> em que o Amazon Location está disponível.


As abas a seguir mostram detalhes em cada área de atributo.

## Map Features

A tabela a seguir mostra os atributos do mapa por provedor de dados. Para obter mais informações sobre conceitos de mapas, consulte [Mapas](#).

Provedor de dados	Tipos de mapa compatíveis	Níveis de zoom vetorial	Níveis de zoom raster
Esri	Vetor  Raster (imagens)  Para obter mais informações, consulte <a href="#">Estilos de mapa da Esri</a> .	0-15	0-23

Provedor de dados	Tipos de mapa compatíveis	Níveis de zoom vetorial	Níveis de zoom raster
Grab	<p>Vetor</p> <p>(Somente no <a href="#">sudeste asiático</a>)</p> <p>Para obter mais informações, consulte <a href="#">Estilos de mapas da Grab</a>.</p>	0-14	nenhuma
HERE	<p>Vetor</p> <p>Raster (imagens)</p> <p>Hybrid</p> <p>Para obter mais informações, consulte <a href="#">HEREestilos de mapa</a>.</p>	1-17	0-19
Open Data	<p>Vetor</p> <p>Para obter mais informações, consulte <a href="#">Estilos de mapas Open Data</a>.</p>	0-15	nenhuma

 Note

Os níveis de zoom representam as configurações máxima e mínima, conforme definido nas configurações de cada provedor APIs. Áreas diferentes do mapa podem ter máximos diferentes. Por exemplo, blocos oceânicos podem ter menos níveis de zoom detalhados do que áreas nas grandes cidades.

MapLibre (e outros mecanismos de renderização de mapas) permitem que você defina os níveis mínimo e máximo de zoom e também respeitem os níveis de zoom do provedor de dados em uma área, para que você não precise escrever código para lidar com essas discrepâncias.

## Places and Search

A tabela a seguir mostra os atributos de local e busca por provedor de dados. Para obter mais informações sobre os conceitos de local, consulte [Pesquisa de lugares](#).

Provedor de dados	Geocodificação	Geocodificação reversa	Autocompletar	GetPlace
Esri	Todos os atributos, exceto:  PlaceId	Todos os atributos, exceto:  TimeZone  PlaceId	Todos os atributos	Todos os atributos
Grab	Todos os atributos, exceto:  unit type  Categorias não compatíveis	Todos os atributos	Todos os atributos	Todos os atributos, exceto:  tipo de unidade  SubMunicipality
HERE	Todos os atributos, exceto:  número de unidade	Todos os atributos	Todos os atributos	Todos os atributos, exceto:  número de unidade

Provedor de dados	Geocodificação	Geocodificação reversa	Autocompletar	GetPlace
	tipo de unidade			tipo de unidade
	relevância			SubMunicipality
	<a href="#">Limitações adicionais na filtragem</a>			
Open Data	Sem compatibilidade	Sem compatibilidade	Sem compatibilidade	Suporta apenas: SubMunicipality

## Route features

A tabela a seguir mostra os atributos de rotas por provedor de dados. Para obter mais informações sobre conceitos de rotas, consulte [Rotas](#). Para obter descrições mais detalhadas das limitações da matriz de rotas, consulte [Restrições nas posições de partida e destino](#).

Provedor de dados	Modos de viagem	Calcular rota	Matriz de rotas
Esri	Carro, caminhão, caminhada	A origem e o destino devem estar a 400 km um do outro. O tempo total de viagem não pode ser superior a 400 minutos.  ArrivalTime não é suportado.	Até 10 posições de origem e destino.  Não é compatível na Coreia.  Todos os pares de origem e destino devem estar a 400 km um do outro.
Grab	Carro, motocicleta.	Sem limites de distância.	Até 350 posições de origem e destino.



Provedor de dados	Modos de viagem	Calcular rota	Matriz de rotas
	Caminhada e bicicleta em <a href="#">cidades selecionadas</a> .		
HERE	Carro, caminhão, caminhada	Sem limite de distância. As rotas que percorrem mais de 10 km fora de um perímetro ao redor das posições de origem e destino não serão calculadas.	Até 350 posições de origem e destino.  Todas as posições de origem e destino devem estar em um perímetro de 180 km.  Há suporte para rotas mais longas, mas com <a href="#">restrições adicionais</a> .
Open Data	Sem compatibilidade	Sem compatibilidade	Sem compatibilidade

## Termos de uso e atribuição de dados para provedores de dados

Antes de usar um provedor de dados, certifique-se de cumprir todos os requisitos legais aplicáveis, incluindo os termos de licença aplicáveis ao uso do provedor.

Para obter mais informações sobre os AWS requisitos, consulte os [Termos AWS de Serviço](#).

Ao usar um provedor de dados com seus recursos do Amazon Location para seu aplicativo ou documentação, lembre-se de fornecer atribuições para cada provedor de dados que você usa.

Para obter mais informações sobre conformidade e atribuição para cada provedor de dados, consulte os tópicos abaixo.

- Esri: [Termos de uso e atribuição de dados: Esri](#)
- Grab: [Termos de uso e atribuição de dados: Grab](#)
- HERE – [Termos de uso e atribuição de dados: HERE](#)
- Open data: [Termos de uso e atribuição de dados: Open Data](#)


## Regiões e endpoints do Amazon Location

A localização da Amazon está disponível nas seguintes AWS regiões:

### Regiões

Nome da região	Região	Endpoint	Protocolo
Leste dos EUA (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS
Leste dos EUA (Norte da Virgínia)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Ásia-Pacífico (Mumbai)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Ásia-Pacífico (Tóquio)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS

Nome da região	Região	Endpoint	Protocolo
Canadá (Central)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Europa (Londres)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
América do Sul (São Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

 Note

Para obter mais informações sobre como usar os endpoints nesta tabela, consulte a seção a seguir.

## Endpoints

A sintaxe geral de um endpoint regional do Amazon Location é a seguinte:

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

Dentro dessa sintaxe, o Amazon Location usa os seguintes códigos de serviço:

Serviço	Código de serviço
Amazon Location Maps	mapas
Amazon Location Places	locais
Amazon Location Geofences	geocercas
Amazon Location Trackers	rastreamento
Amazon Location Routes	rotas

Por exemplo, o endpoint regional do Amazon Location Maps para o Leste dos EUA (Norte da Virgínia) seria: <https://maps.geo.us-east-1.amazonaws.com>.

## API Endpoints de operação

A sintaxe de um endpoint do ambiente de gerenciamento do Amazon Location Service é a seguinte:

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

As ações do ambiente de gerenciamento do Amazon Location Service são:

Serviço	Endpoint	API operação
Amazon Location Maps	<a href="https://cp.maps.geo.us-east-1.amazonaws.com">https://cp.maps.geo.us-east-1.amazonaws.com</a>	<a href="#">CreateMap</a> <a href="#">DeleteMap</a> <a href="#">DescribeMap</a>

Serviço	Endpoint	APIoperação
		<a href="#">ListMaps</a> <a href="#">UpdateMap</a>
Amazon Location Places	https://cp.places. geo. <i>region</i> .amazonaws.com	<a href="#">CreatePlaceIndex</a> <a href="#">DeletePlaceIndex</a> <a href="#">DescribePlaceIndex</a> <a href="#">ListPlaceIndexes</a> <a href="#">UpdatePlaceIndex</a>
Amazon Location Geofences	https://cp.geofenc ing.geo. <i>region</i> .amazonaw s.com	<a href="#">CreateGeofenceCollection</a> <a href="#">DeleteGeofenceCollection</a> <a href="#">DescribeGeofenceCollection</a> <a href="#">ListGeofenceCollections</a> <a href="#">UpdateGeofenceCollection</a>
Amazon Location Trackers	https://cp.trackin g.geo. <i>region</i> .amazonaw s.com	<a href="#">CreateTracker</a> <a href="#">DeleteTracker</a> <a href="#">DescribeTracker</a> <a href="#">UpdateTracker</a> <a href="#">ListTrackers</a> <a href="#">AssociateTrackerConsumer</a> <a href="#">DisassociateTrackerConsumer</a> <a href="#">ListTrackerConsumers</a>

Serviço	Endpoint	APIoperação
Amazon Location Routes	https://cp.routes. geo. <i>region</i> .amazonaws.com	<a href="#">CreateRouteCalculator</a> <a href="#">DeleteRouteCalculator</a> <a href="#">DescribeRouteCalculator</a> <a href="#">ListRouteCalculators</a> <a href="#">UpdateRouteCalculator</a>
Amazon Location Metadata	https://cp.metadat a.geo. <i>region</i> .amazonaw s.com	<a href="#">CreateKey</a> <a href="#">DeleteKey</a> <a href="#">DescribeKey</a> <a href="#">ListKeys</a> <a href="#">UpdateKey</a>

## Amazon Location Service quotas

Este tópico apresenta um resumo dos limites de tarifas e cotas do Amazon Location Service.

### Note

Se precisar de cotas mais altas, você poderá usar o console do Service Quotas para [solicitar aumentos das cotas](#) para cotas ajustáveis. Ao solicitar um aumento de cota, selecione a região na qual você deseja aumentar a cota, pois a maioria das cotas é específica da região. AWS

As Cotas de Serviço são o número máximo de recursos que você pode ter por AWS conta e AWS região. O Amazon Location Service nega solicitações adicionais que excedam a Cota de serviço.

Limites de taxa (cotas que começam com Taxa de... ) são o número máximo de solicitações por segundo, com uma taxa de intermitência de 80% do limite em qualquer parte do segundo, definida

para cada API operação. O Amazon Location Service controla a utilização de solicitações que excedem o limite de taxa da operação.

Nome	Padrão	Ajustá	Descrição
API Principais recursos por conta	Cada região com suporte: 500	Não	O número máximo de API recursos-chave (ativos ou expirados) que você pode ter por conta.
Recursos da Geofence Collection (Coleção de Geocercas) por conta	Cada região compatível: 1.500	<a href="#">Yes</a> (Sim)	Número máximo de recursos da Geofence Collection (Coleção de Geocercas) que você pode criar por conta.
Geocercas por Geofence Collection	Cada região compatível: 50.000	Não	O número máximo de geocercas que você pode criar por Geofence Collection (Coleção de Geocercas).
Mapear recursos por conta	Cada região compatível: 40	<a href="#">Sim</a>	O número máximo de recursos do Mapa que você pode criar por conta.
Recursos Índice de local por conta	Cada região compatível: 40	<a href="#">Sim</a>	Número máximo de recursos de Place Index (Índice de Local) que você pode criar em cada conta.
Taxa de AssociateTrackerConsumer API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de AssociateTrackerConsumer solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de BatchDeleteDevicePositionHistory API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchDeleteDevicePositionHistory solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de BatchDeleteGeofence API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchDeleteGeofence solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de BatchEvaluateGeofences API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchEvaluateGeofences solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de BatchGetDevicePosition API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchGetDevicePosition solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de BatchPutGeofence API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchPutGeofence solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.



Nome	Padrão	Ajuste	Descrição
Taxa de BatchUpdateDevicePosition API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de BatchUpdateDevicePosition solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CalculateRoute API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CalculateRoute solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CalculateRouteMatrix API solicitações	Cada região compatível: 5 por segundo	<a href="#">Sim</a>	O número máximo de CalculateRouteMatrix solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CreateGeofenceCollection API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreateGeofenceCollection solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CreateKey API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreateKey solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de CreateMap API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreateMap solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CreatePlaceIndex API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreatePlaceIndex solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CreateRouteCalculator API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreateRouteCalculator solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de CreateTracker API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de CreateTracker solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DeleteGeofenceCollection API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeleteGeofenceCollection solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de DeleteKey API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeleteKey solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DeleteMap API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeleteMap solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DeletePlaceIndex API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeletePlaceIndex solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DeleteRouteCalculator API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeleteRouteCalculator solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DeleteTracker API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DeleteTracker solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajusté	Descrição
Taxa de DescribeGeofenceCollection API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribeGeofenceCollection solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DescribeKey API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribeKey solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DescribeMap API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribeMap solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DescribePlaceIndex API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribePlaceIndex solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DescribeRouteCalculator API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribeRouteCalculator solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de DescribeTracker API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DescribeTracker solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de DisassociateTrackerConsumer API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de DisassociateTrackerConsumer solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ForecastGeofenceEvents API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de ForecastGeofenceEvents solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetDevicePosition API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetDevicePosition solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetDevicePositionHistory API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetDevicePositionHistory solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajusté	Descrição
Taxa de GetGeofence API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetGeofence solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetMapGlyphs API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetMapGlyphs solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetMapSprites API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetMapSprites solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetMapStyleDescriptor API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetMapStyleDescriptor solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetMapTile API solicitações	Cada região compatível: 500 por segundo	<a href="#">Sim</a>	O número máximo de GetMapTile solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de GetPlace API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de GetPlace solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de ListDevicePositions API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de ListDevicePositions solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListGeofenceCollections API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListGeofenceCollections solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListGeofences API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de ListGeofences solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListKeys API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListKeys solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListMaps API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListMaps solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajuste	Descrição
Taxa de ListPlaceIndexes API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListPlaceIndexes solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListRouteCalculators API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListRouteCalculators solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListTagsForResource API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListTagsForResource solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListTrackerConsumers API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListTrackerConsumers solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de ListTrackers API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de ListTrackers solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.



Nome	Padrão	Ajuste	Descrição
Taxa de PutGeofence API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de PutGeofence solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de SearchPlaceIndexForPosition API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de SearchPlaceIndexForPosition solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de SearchPlaceIndexForSuggestions API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de SearchPlaceIndexForSuggestions solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de SearchPlaceIndexForText API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de SearchPlaceIndexForText solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de TagResource API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de TagResource solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajusté	Descrição
Taxa de UntagResource API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UntagResource solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de UpdateGeofenceCollection API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdateGeofenceCollection solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de UpdateKey API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdateKey solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de UpdateMap API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdateMap solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de UpdatePlaceIndex API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdatePlaceIndex solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.

Nome	Padrão	Ajusté	Descrição
Taxa de UpdateRouteCalculator API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdateRouteCalculator solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de UpdateTracker API solicitações	Cada região compatível: 10 por segundo	<a href="#">Sim</a>	O número máximo de UpdateTracker solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Taxa de VerifyDevicePosition API solicitações	Cada região compatível: 50 por segundo	<a href="#">Sim</a>	O número máximo de VerifyDevicePosition solicitações que você pode fazer por segundo. Solicitações adicionais são limitadas.
Recursos da Calculadora de rotas por conta	Cada região compatível: 40	<a href="#">Sim</a>	Número máximo de recursos Calculadora de rotas que você pode criar por conta.
Consumidores de rastreador por rastreador	Cada região compatível: 5	Não	Número máximo do Geofence Collection ao qual o recurso do rastreador pode ser associado.
Recursos de Rastreamento por conta	Cada região com suporte: 500	<a href="#">Sim</a>	Número máximo de recursos do rastreador que você pode criar por conta.

**Note**

Você pode monitorar o seu uso em relação às suas cotas com o CloudWatch. Para obter mais informações, consulte [Usando CloudWatch para monitorar o uso em relação às cotas](#).

## Gerenciar suas cotas do Amazon Location Service

O Amazon Location Service é integrado ao Service Quotas, um AWS serviço que permite que você visualize e gerencie suas cotas a partir de um local central. Para obter mais informações, consulte [What Is Service Quotas?](#) (O que é o Service Quotas?) no Guia do usuário do Service Quotas.

O Service Quotas facilita a pesquisa do valor das cotas do seu Amazon Location Service.

### AWS Management Console

Para ver as cotas do Amazon Location Service usando o console

1. Abra o console Service Quotas em. <https://console.aws.amazon.com/servicequotas/>
2. No painel de navegação, selecione serviços da AWS .
3. Na lista de AWS serviços, procure e selecione Amazon Location.

Na lista Service quotas é possível ver o nome da service quota, o valor aplicado (se estiver disponível), a cota padrão da AWS e se o valor da cota é ajustável.

4. Para visualizar informações adicionais sobre uma Service Quota, como a descrição, escolha o nome da cota.
5. (Opcional) Para solicitar um aumento de cota, selecione a cota que deseja aumentar, selecione Requesting a quota increase (Solicitar aumento de cota), insira ou selecione as informações necessárias e selecione Request (Solicitar).

Para trabalhar mais com service quotas usando o console, consulte o [Guia do usuário de Service Quotas](#). Para solicitar o aumento da quota, consulte [Solicitar um aumento de quota](#) no Guia do usuário do Service Quotas.

### AWS CLI

Visualizar as quotas do Amazon Location Service usando o AWS CLI

Execute o comando a seguir para visualizar as quotas padrão do Amazon Location.

```
aws service-quotas list-aws-default-service-quotas \
  --query 'Quotas[*]'.
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
  --service-code geo \
  --output table
```

Para trabalhar mais com cotas de serviço usando o AWS CLI, consulte a Referência de Comandos de [AWS CLI Cotas de Serviço](#). Para solicitar um aumento de quotas, consulte o comando [request-service-quota-increase](#) na [Referência de comandos do AWS CLI](#).

# Conceitos básicos para desenvolvedores usando o Amazon Location Service

Você pode usar o Amazon Location Service para fornecer funcionalidade geográfica para aplicativos em vários formatos e sistemas diferentes, incluindo serviços web de back-end, aplicativos web e aplicativos móveis. Há muitas ferramentas fornecidas para ajudá-lo a criar seus aplicativos, incluindo SDKs, bibliotecas e exemplos de código.

Esta seção fornece informações e links para ajudar você a começar a trabalhar com o Amazon Location. Principalmente, os tópicos a seguir fornecem informações que podem ser muito úteis para você:

- [Cenários e casos de uso](#) – Uma lista de cenários de desenvolvimento e como o Amazon Location Service pode ajudá-lo a concluí-los.
- [SDKs e ferramentas do Amazon Location](#) – os kits de desenvolvimento de software (SDKs) e bibliotecas que ajudarão você a programar com o Amazon Location.
- [Referência da API do Amazon Location Service](#) — Uma referência às principais APIs de localização da Amazon que são fornecidas com o AWS SDK.
- [Exemplos de código](#) – esta seção fornece exemplos que ajudarão você a começar ou a adicionar funcionalidade ao seu aplicativo existente.
- [Tutorial de início rápido](#) – este tutorial mostra como criar seu primeiro aplicativo. Há versões do tutorial para criar um aplicativo web ou um aplicativo móvel baseado em Android.
- [Conceitos do Amazon Location Service](#) – Esta seção deste guia descreve os conceitos básicos do Amazon Location, incluindo seções sobre mapas, pesquisas por lugares, rotas, geocercas e rastreadores.
- [Amplify](#) – o Amplify é uma solução completa que engloba muitas das funcionalidades necessárias para criar aplicativos web e móveis usando o Nuvem AWS. Se você já estiver usando o Amplify ou optar por usar o Amplify, ele tem uma biblioteca geográfica, que tem o Amazon Location Service embutido, que você pode usar. Para começar a usar o Amplify Geo, consulte a documentação [aqui](#).

## Cenários e casos de uso

O Amazon Location Service é um serviço executado na Nuvem AWS. Você pode chamá-lo a partir de suas próprias instâncias do Amazon EC2 na nuvem, mas muitos aplicativos de mapeamento serão executados em dispositivos ou em uma combinação de dispositivos e nuvem. A seguir estão listados apenas alguns cenários típicos e como você pode abordar o desenvolvimento deles.

- Um aplicativo de back-end que ajuda você a otimizar rotas para motoristas em sua frota.

Você pode escrever um aplicativo executado no [Amazon EC2](#) na Nuvem AWS que usa o Amazon Location Service para [calcular matrizes de rotas](#) como uma entrada para um otimizador de rotas para sua frota. Use o [AWS SDK](#) para fazer chamadas para o Amazon Location.

- Um aplicativo da web que permite que seus clientes encontrem os locais de sua empresa.

Você pode criar um site que seja executado em instâncias do Amazon EC2, incluindo um aplicativo baseado em localização. Use o [AWS SDK JavaScript para](#) desenvolver um aplicativo web para pesquisar locais usando a [pesquisa de lugares](#) e exibir os resultados em um [mapa](#) usando MapLibre. Use o SDK do Amazon Location para facilitar a programação com localização.

- Adicione atributos de localização a um aplicativo para iOS ou Android existente.

Você pode usar o AWS SDK para Swift (iOS) ou [Kotlin](#) (Android) para fazer chamadas para a Amazon Location e adicionar a funcionalidade de [pesquisa de lugares](#) e [mapas](#) ao seu aplicativo. Use MapLibre para renderizar mapas. Há outros [AWS SDKs](#) disponíveis para outros idiomas.

- Rastreie ativos (dispositivos ou veículos) e receba atualizações quando eles entrarem ou saírem das áreas definidas por você.

Um aplicativo para rastrear dispositivos consiste em várias partes.

- Cada dispositivo que você está rastreando deve ter um recurso de [rastreador](#) criado para rastreá-lo. Ele deve enviar atualizações de posição para o Amazon Location Service, por exemplo, usando o [MQTT](#).
- Crie [geocercas](#) para definir áreas nas quais você deseja obter eventos de entrada e saída para seus ativos.
- Você pode usar o [Amazon EC2](#) ou [AWS Lambda](#) responder aos seus eventos à medida que os ativos entram ou saem das áreas da geocerca.
- Você pode expandir isso para criar aplicativos web ou em dispositivos para rastrear e exibir a localização de seus ativos em mapas.

A seção a seguir fornece detalhes sobre ferramentas e bibliotecas disponíveis para uso com cada aspecto do Amazon Location Service.

## SDKs e ferramentas para usar o Amazon Location Service

Existem várias ferramentas que ajudarão você a usar o Amazon Location Service.

- **AWS SDKs** — Os kits de desenvolvimento de AWS software (SDKs) estão disponíveis em muitas linguagens de programação populares, fornecendo uma API, exemplos de código e documentação que facilitam a criação de aplicativos em sua linguagem preferida. Os AWS SDKs incluem as principais APIs e funcionalidades de localização da Amazon, incluindo acesso a mapas, pesquisa de lugares, rotas, cercas geográficas e rastreadores. Para saber mais sobre os SDKs disponíveis para uso com o Amazon Location Service em diferentes aplicativos e idiomas, consulte [SDKs por linguagem](#).
- **MapLibre**— O Amazon Location Service recomenda a renderização de mapas usando o mecanismo de [MapLibre](#) renderização. MapLibre é um mecanismo para exibir mapas em aplicativos da web ou móveis. MapLibre também tem um modelo de plug-in e suporta interface de usuário para pesquisas e rotas em alguns idiomas e plataformas. Para saber mais sobre o uso MapLibre e a funcionalidade que ele fornece, consulte [MapLibre](#).
- **SDK do Amazon Location** – o SDK do Amazon Location é um conjunto de bibliotecas de código aberto que facilita o desenvolvimento de aplicativos com o Amazon Location Service. As bibliotecas fornecem funcionalidade para oferecer suporte à autenticação para aplicativos móveis e web, rastreamento de localização para aplicativos móveis, conversão entre tipos de dados de localização da Amazon e [GeoJSON](#), bem como um pacote hospedado do cliente Amazon Location para AWS o SDK v3. Para saber mais sobre o SDK do Amazon Location, consulte [SDKs do Amazon Location](#).

### SDKs por linguagem

As tabelas a seguir fornecem informações sobre AWS SDKs e MapLibre versões para linguagens e estruturas, por tipo de aplicativo: aplicativo web, móvel ou back-end.

#### Versões do SDK

Recomendamos que você use a versão mais recente do AWS SDK e quaisquer outros SDKs usados em seus projetos e que mantenha os SDKs atualizados. O AWS SDK fornece



os recursos e funcionalidades mais recentes, além de atualizações de segurança. Para encontrar a versão mais recente do AWS SDK JavaScript, por exemplo, consulte o tópico de [instalação do navegador](#) no AWS SDK para JavaScript ver a documentação.

## Web frontend

Os seguintes AWS SDKs e MapLibre versões estão disponíveis para desenvolvimento de aplicativos front-end na web.

Linguagem/Estrutura	SDK da AWS	Estrutura de renderização
Suporte total		
JavaScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://maplibre.org/projects/maplibre-gl-js/">https://maplibre.org/projects/maplibre-gl-js/</a>
ReactJS	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://github.com/maplibre/maplibre-react-native">https://github.com/maplibre/maplibre-react-native</a>
TypeScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://maplibre.org/projects/maplibre-gl-js/">https://maplibre.org/projects/maplibre-gl-js/</a>
Suporte parcial		
Flutter	<a href="https://docs.amplify.aws/start/q/integration/flutter/">https://docs.amplify.aws/start/q/integration/flutter/</a>  O Flutter ainda não é totalmente suportado pelo AWS, mas um suporte limitado é oferecido via Amplify.	<a href="https://github.com/maplibre/flutter-maplibre-gl">https://github.com/maplibre/flutter-maplibre-gl</a>  A biblioteca MapLibre Flutter é considerada experimental.
Node.js	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	Não há MapLibre suporte para Node.js.
PHP	<a href="https://aws.amazon.com/sdk-for-php/">https://aws.amazon.com/sdk-for-php/</a>	Não há MapLibre suporte para PHP.

## Mobile frontend

Os seguintes AWS SDKs e MapLibre versões estão disponíveis para desenvolvimento de aplicativos front-end móveis.

Linguagem/Estrutura	AWS SDK	Estrutura de renderização
Suporte total		
Java	<a href="https://aws.amazon.com/sdk-for-java/">https://aws.amazon.com/sdk-for-java/</a>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>
Kotlin	<p><a href="https://aws.amazon.com/sdk-for-kotlin/">https://aws.amazon.com/sdk-for-kotlin/</a></p> <p>SDK de autenticação móvel do Amazon Location Service para Android: <a href="https://github.com/aws-geospatial/sdk-android">amazon-location-mobile-auth</a> <a href="https://github.com/aws-geospatial/sdk-android">https://github.com/aws-geospatial/sdk-android</a></p> <p>SDK de rastreamento móvel do Amazon Location Service para Android: <a href="https://github.com/aws-geospatial/sdk-android">amazon-location-mobile-tracking</a> <a href="https://github.com/aws-geospatial/sdk-android">https://github.com/aws-geospatial/sdk-android</a></p>	<p><a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a></p> <p>Requer vinculações personalizadas, assim como MapLibre é baseado em Java.</p>
ObjectiveC	<a href="https://github.com/aws-amplify/aws-sdk-ios">https://github.com/aws-amplify/aws-sdk-ios</a>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>
ReactNative	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>	<a href="https://github.com/maplibre/maplibre-react-native">https://github.com/maplibre/maplibre-react-native</a>
Swift	<p><a href="https://aws.amazon.com/sdk-for-swift/">https://aws.amazon.com/sdk-for-swift/</a></p> <p>SDK de autenticação móvel do Amazon Location</p>	<a href="https://maplibre.org/projects/maplibre-native/">https://maplibre.org/projects/maplibre-native/</a>

Linguagem/Estrutura	AWS SDK	Estrutura de renderização
	<p>Service para iOS: <a href="https://github.com/aws-geospatial/sdk-ios">amazon-location-mobile-auth</a> <a href="https://github.com/aws-geospatial/sdk-ios">https://github.com/aws-geospatial/sdk-ios</a></p> <p>SDK de rastreamento móvel do Amazon Location Service para iOS: <a href="https://github.com/aws-geospatial/sdk-ios">amazon-location-mobile-tracking</a> <a href="https://github.com/aws-geospatial/sdk-ios">https://github.com/aws-geospatial/sdk-ios</a></p>	
Suporte parcial		
Flutter	<p><a href="https://docs.amplify.aws/start/q/integration/flutter/">https://docs.amplify.aws/start/q/integration/flutter/</a></p> <p>O Flutter ainda não é totalmente suportado pelo AWS, mas um suporte limitado é oferecido via Amplify.</p>	<p><a href="https://github.com/maplibre/flutter-maplibre-gl">https://github.com/maplibre/flutter-maplibre-gl</a></p> <p>A biblioteca MapLibre Flutter é considerada experimental.</p>

## Backend application

Os seguintes AWS SDKs estão disponíveis para desenvolvimento de aplicativos de back-end. MapLibre não está listado aqui, porque a renderização de mapas normalmente não é necessária para aplicativos de back-end.

Idioma	SDK da AWS
.NET	<a href="https://aws.amazon.com/sdk-for-net/">https://aws.amazon.com/sdk-for-net/</a>
C++	<a href="https://aws.amazon.com/sdk-for-cpp/">https://aws.amazon.com/sdk-for-cpp/</a>
Go	<a href="https://aws.amazon.com/sdk-for-go/">https://aws.amazon.com/sdk-for-go/</a>

Idioma	SDK da AWS
Java	<a href="https://aws.amazon.com/sdk-for-java/">https://aws.amazon.com/sdk-for-java/</a>
JavaScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>
Node.js	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>
TypeScript	<a href="https://aws.amazon.com/sdk-for-javascript/">https://aws.amazon.com/sdk-for-javascript/</a>
Kotlin	<a href="https://aws.amazon.com/sdk-for-kotlin/">https://aws.amazon.com/sdk-for-kotlin/</a>
PHP	<a href="https://aws.amazon.com/sdk-for-php/">https://aws.amazon.com/sdk-for-php/</a>
Python	<a href="https://aws.amazon.com/sdk-for-python/">https://aws.amazon.com/sdk-for-python/</a>
Ruby	<a href="https://aws.amazon.com/sdk-for-ruby/">https://aws.amazon.com/sdk-for-ruby/</a>
Rust	<a href="https://aws.amazon.com/sdk-for-rust/">https://aws.amazon.com/sdk-for-rust/</a>  O AWS SDK para Rust está em versão prévia para desenvolvedores.

## Usando MapLibre ferramentas e bibliotecas com a Amazon Location

Uma das ferramentas importantes para criar aplicativos interativos com o Amazon Location é MapLibre. [MapLibre](#) é principalmente um mecanismo de renderização para exibir mapas em um aplicativo web ou móvel. No entanto, ele também inclui suporte para plug-ins e fornece funcionalidade para trabalhar com outros aspectos do Amazon Location. A seguir, descrevemos as ferramentas que você pode usar com base na área de localização com a qual você deseja trabalhar.

### Note

Para usar qualquer aspecto do Amazon Location, instale o [AWS SDK para a linguagem que você deseja usar](#).

- Mapas

Para exibir mapas em seu aplicativo, você precisa de um mecanismo de renderização de mapas que use os dados fornecidos pela Amazon Location e desenhe na tela. Os mecanismos de renderização de mapas também fornecem funcionalidade para deslocar e ampliar o mapa ou para adicionar marcadores, alfinetes e outras anotações ao mapa.

O Amazon Location Service recomenda a renderização de mapas usando o mecanismo de [MapLibre](#) renderização. MapLibre O GL JS é um mecanismo para exibir mapas em JavaScript, enquanto o MapLibre Native fornece mapas para iOS ou Android.

MapLibre também tem um ecossistema de plug-ins para estender a funcionalidade principal. Para obter mais informações, visite <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

- Pesquisa de lugares

Para simplificar a criação de uma interface de usuário de pesquisa, você pode usar o [MapLibre geocodificador](#) para web (os aplicativos Android podem usar o plug-in [Android Places](#)).

Use a [biblioteca de geocodificadores Amazon Location for Maplibre](#) para simplificar o processo de uso do Amazon Location em aplicativos. `amazon-location-for-maplibre-gl-geocoder`  
JavaScript

- Rotas

Para exibir rotas no mapa, use as [MapLibredireções](#).

- Geocercas e rastreadores

MapLibre não tem nenhuma renderização ou ferramenta específica para cercas geográficas e rastreamento, mas você pode usar a funcionalidade de renderização e os [plug-ins](#) para mostrar as cercas geográficas e os dispositivos rastreados no mapa.

Os dispositivos que estão sendo rastreados podem usar o [MQTT](#) ou enviar atualizações manualmente para o Amazon Location Service. Os eventos de geocercas podem ser respondidos usando [AWS Lambda](#).

Muitas bibliotecas de código aberto estão disponíveis para fornecer funcionalidade adicional para o Amazon Location Service, por exemplo, o [Turf](#), que fornece funcionalidade de análise espacial.

Muitas bibliotecas usam os dados formatados [GeoJSON](#) de padrão aberto. O Amazon Location Service fornece uma biblioteca para apoiar o uso do GeoJSON em aplicativos JavaScript. Para obter mais informações, consulte a próxima sessão, [SDKs e bibliotecas do Amazon Location](#).

## Plug-in Amazon Location MapLibre Geocoder

O plug-in do MapLibre geocodificador Amazon Location foi projetado para facilitar a incorporação da funcionalidade Amazon Location em seus JavaScript aplicativos, ao trabalhar com renderização de mapas e geocodificação usando a biblioteca. [maplibre-gl-geocoder](#)

### Instalação

Você pode instalar o plug-in Amazon Location MapLibre geocoder do NPM para uso com módulos, com este comando:

```
npm install @aws/amazon-location-for-maplibre-gl-geocoder
```

Você pode importar para um arquivo HTML para uso diretamente no navegador, com um script:

```
<script src="https://www.unpkg.com/@aws/amazon-location-for-maplibre-gl-geocoder@1"/>/script<
```

### Uso com o módulo

Esse código configura um mapa Maplibre GL JavaScript com recursos de geocodificação de localização da Amazon. Ele usa a autenticação via Pool de Identidade do Amazon Cognito para acessar os recursos de localização da Amazon. O mapa é renderizado com um estilo e coordenadas centrais especificados e permite pesquisar lugares no mapa.

```
// 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());
```

## Uso com um navegador

Este exemplo usa o Amazon Location Client para fazer uma solicitação que se autentica usando o Amazon Cognito.

### Note

Alguns desses exemplos usam o Amazon Location Client. O Amazon Location Client é baseado no [AWS SDK para JavaScript V3](#) e permite fazer chamadas para o Amazon Location por meio de um script referenciado em um arquivo HTML.

Inclua o seguinte em um arquivo HTML:

```
< 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>
```

Inclua o seguinte em um JavaScript arquivo:

```
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());
```

Abaixo estão listadas as funções e os comandos usados no plug-in de MapLibre geocodificador Amazon Location:

- **buildAmazonLocationMaplibreGeocoder**

Essa classe cria uma instância da `AmazonLocationMaplibreGeocder` que é o ponto de entrada para todas as outras chamadas:



```
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
  placesIndex, {enableAll: true});
```

### • **getPlacesGeocoder**

Retorna um objeto IControl pronto para uso que pode ser adicionado diretamente a um mapa.

```
const geocoder = getPlacesGeocoder();

// Initialize map
let map = await initializeMap();

// Add the geocoder to the map.
map.addControl(geocoder);
```

## SDKs e bibliotecas do Amazon Location

O SDK do Amazon Location é um conjunto de bibliotecas de código aberto que fornecem funcionalidades úteis para o desenvolvimento de aplicativos do Amazon Location. As seguintes funcionalidades estão incluídas:

- Cliente Amazon Location — Os objetos Amazon Location no AWS SDK v3 são agrupados e empacotados para facilitar o uso no desenvolvimento web.
- Autenticação — O utilitário de autenticação simplifica a autenticação (usando o Amazon Cognito ou chaves de API) ao criar uma página da web [JavaScript](#), um [aplicativo iOS](#) ou [Android](#) para o Amazon Location Service.
- Rastreamento — Os SDKs de rastreamento móvel estão disponíveis para [iOS](#) e [Android](#). Esse SDK facilita a interação de aplicativos móveis com os Amazon Location Trackers.
- Funções Amazon Location GeoJSON — [Os utilitários de conversão GeoJSON facilitam a conversão entre os dados formatados GeoJSON padrão do setor e os formatos da Amazon Location API.](#)

### Tópicos

- [Como começar a usar o SDK do Amazon Location](#)
- [Cliente do Amazon Location](#)
- [JavaScript Auxiliar de autenticação](#)

- [Auxiliares de conversão GeoJSON](#)
- [SDK de autenticação móvel para Android](#)
- [SDK de autenticação móvel para iOS](#)
- [SDK de rastreamento móvel para Android](#)
- [SDK de rastreamento móvel para iOS](#)

## Como começar a usar o SDK do Amazon Location

O SDK do Amazon Location é um conjunto de funções que pode simplificar o uso do Amazon Location Service em um aplicativo. Você pode instalar e importar essas funções para o seu JavaScript aplicativo. As seções a seguir descrevem o cliente Amazon Location e as bibliotecas auxiliares de autenticação e GeoJSON.

### Cliente do Amazon Location

Com o AWS SDK v3, o SDK é separado por serviço. Você pode instalar apenas as peças necessárias. Por exemplo, para instalar o cliente do Amazon Location e o provedor de credenciais para o Amazon Cognito, use os seguintes comandos.

```
npm install @aws-sdk/client-location
npm install @aws-sdk/credential-providers
```

Para facilitar o uso do Amazon Location Service em aplicativos de front-end JavaScript da web, AWS fornece um pacote hospedado da biblioteca Amazon Location e do provedor de credenciais. Para usar o cliente do pacote, adicione-o ao seu HTML em uma tag de script, da seguinte forma:

```
<script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
```

#### Note

O pacote é mantido atualizado e compatível com versões anteriores para facilitar o uso. Usar essa tag de script ou a instalação do NPM sempre obterá a versão mais recente.

## JavaScript Auxiliar de autenticação

O assistente de JavaScript autenticação de localização da Amazon simplifica a autenticação ao fazer chamadas à API de localização da Amazon a partir do seu aplicativo. JavaScript Esse auxiliar de autenticação ajuda você especificamente ao usar o [Amazon Cognito](#) [ou as chaves de API](#) como método de autenticação. Esta é uma biblioteca de código aberto que está disponível aqui: <https://github.com/aws-geospatial/amazon-location-utilities-auth-helper-js>. GitHub

### Note

O suporte ao Amazon Cognito no auxiliar de autenticação não oferece suporte ao atributo de identidades federadas do Amazon Cognito.

## Instalação

Você pode usar as bibliotecas com uma instalação local, se usar um sistema de compilação como o webpack, ou incluindo JavaScript pacotes pré-construídos com `<script>` tags em seu html.

- Use o comando a seguir para instalar a biblioteca, usando NPM:

```
npm install @aws/amazon-location-utilities-auth-helper
```

- Use o comando a seguir em seu arquivo HTML para carregar o script:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/dist/amazonLocationAuthHelper.js"></script>
```

## Importar

Para usar uma função específica em seu JavaScript aplicativo, você deve importar essa função. O código a seguir é usado para importar a função `withIdentityPoolId` para seu aplicativo.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';
```

## Opções de autenticação

Os auxiliares de autenticação de localização da Amazon incluem as seguintes funções que retornam um objeto `AuthHelper`:

- `async withIdentityPoolId( identityPoolId: string): AuthHelper`— Essa função retorna um `AuthHelper` objeto, inicializado para funcionar com o Amazon Cognito
- `async withAPIKey( API_KEY: string): AuthHelper`— Essa função retorna um `AuthHelper` objeto, inicializado para funcionar com chaves de API.

O objeto `AuthHelper` fornece as seguintes funções:

- `AuthHelper.getMapAuthenticationOptions()`— Essa função do `AuthHelper` objeto retorna um JavaScript objeto com o `transformRequest` que pode ser usado com as opções de mapa em MapLibre JS. Fornecido somente quando inicializado com um banco de identidades.
- `AuthHelper.getLocationClientConfig()`— Essa função do `AuthHelper` objeto retorna um JavaScript objeto com o `credentials` que pode ser usado para inicializar um `LocationClient`.
- `AuthHelper.getCredentials()`— Essa função do `AuthHelper` objeto retorna as credenciais internas do Amazon Cognito. Fornecido somente quando inicializado com um banco de identidades.

Exemplo: inicializando um objeto de MapLibre mapa com o Amazon Cognito, usando um `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
});
```

Exemplo: inicializar um objeto de MapLibre mapa com uma chave de API (não `AuthHelper` é necessário neste caso)

```
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,
});
```

Exemplo: inicialize o cliente Location a partir do AWS SDK para JS, usando o Amazon Cognito e AuthHelper

Este exemplo usa o AWS SDK para JavaScript v3.

```
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
}));
```

Exemplo: inicialize o cliente Location a partir do AWS SDK para JS, usando uma chave de API e AuthHelper

Este exemplo usa o AWS SDK para JavaScript v3.

```
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
}));
```

## Auxiliares de conversão GeoJSON

Os auxiliares de conversão GeoJSON do Amazon Location fornecem ferramentas para converter tipos de dados do Amazon Location Service de e para o formato [GeoJSON](#) padrão do setor. O GeoJSON é usado, por exemplo, MapLibre com para renderizar dados geográficos no mapa. Esta é uma biblioteca de código aberto que está disponível GitHub aqui: <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

### Instalação

Você pode usar as bibliotecas com uma instalação local, como webpack, ou incluindo JavaScript pacotes pré-criados com `<script>` tags em seu html.

- Use o comando a seguir para instalar a biblioteca, usando NPM.

```
npm install @aws/amazon-location-utilities-datatypes
```

- Use o comando a seguir em seu arquivo HTML para carregar o script:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-datatypes@1.x/dist/
amazonLocationDataConverter.js"></script>
```

### Importar

Para usar uma função específica em seu JavaScript aplicativo, você deve importar essa função. O código a seguir é usado para importar a função `placeToFeatureCollection` para seu aplicativo.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utilities-datatypes';
```

## Funções de conversão GeoJSON

Os auxiliares de conversão GeoJSON do Amazon Location incluem as seguintes funções:

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse, keepNull: boolean): FeatureCollection`— Esta função converte respostas das funções de busca de local em um `FeatureCollection` GeoJSON com 1 ou mais recursos de ponto.
- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)`— Esta função converte respostas das funções de posição do dispositivo rastreador em um `GeoJSON FeatureCollection` com 1 ou mais recursos de ponto.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection`— Essa função converte respostas da função calcular rota em um `FeatureCollection` GeoJSON com um único recurso. `MultiStringLine` Cada trecho da rota é representado por uma `LineString` entrada no `MultiStringLine`.
- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection`— Esta função converte a solicitação ou resposta de funções de geofence em um `GeoJSON` com recursos de polígono. `FeatureCollection` Ele pode converter cercas geográficas na resposta e na solicitação, permitindo que você mostre cercas geográficas em um mapa antes de carregá-las com `PutGeofence` ou `BatchPutGeofence`.

Essa função converterá uma geocerca circular em um atributo com um polígono aproximado, mas também terá propriedades de “centro” e “raio” para recriar a geocerca circular, se necessário (veja a próxima função).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]`— Essa função converte um `FeatureCollection` GeoJSON com recursos de polígono em uma matriz `BatchPutGeofenceRequestEntry` de objetos, para que o resultado possa ser usado para criar uma solicitação para `BatchPutGeofence`.

Se um recurso no `FeatureCollection` tiver propriedades de “centro” e “raio”, ele será convertido em uma entrada de solicitação de cerca geográfica circular, ignorando a geometria do polígono.

Exemplo: converter os resultados da pesquisa em uma camada de pontos no MapLibre

Este exemplo usa o AWS SDK para JavaScript v3.

```
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 de autenticação móvel para Android

Esses utilitários ajudam você a se autenticar ao fazer chamadas de API do Amazon Location Service a partir de seus aplicativos Android. Isso ajuda especificamente ao usar o [Amazon Cognito](#) ou [as chaves de API](#) como método de autenticação.

O SDK de autenticação móvel Android está disponível no github: [Amazon Location Service Mobile Authentication SDK](#) for Android. Além disso, tanto o SDK de autenticação móvel quanto o AWS SDK estão disponíveis no repositório [AWS Maven](#).

### Instalação

Para usar o SDK de autenticação móvel, adicione as seguintes instruções de importação ao seu `build.gradle` arquivo no Android Studio.

```
implementation("software.amazon.location:auth:0.0.1")
```



```
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

## Funções de autenticação

O SDK auxiliar de autenticação tem as seguintes funções:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")`: `LocationCredentialsProvider`: essa função retorna uma `LocationCredentialsProvider` inicializada para funcionar com uma chave de API.
- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Essa função retorna um `LocationCredentialsProvider` inicializado para funcionar com um pool de identidades do Amazon Cognito.

## Uso

Para usar o SDK em seu código, importe as seguintes classes:

```
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

Você tem duas opções ao criar as instâncias do auxiliar de autenticação e do provedor do cliente de localização. Você pode criar uma instância usando as [chaves da API de localização da Amazon](#) ou o [Amazon Cognito](#).

- Para criar uma instância auxiliar de autenticação usando uma chave de API de localização da Amazon, declare a classe auxiliar da seguinte forma:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key")
```

- Para criar uma instância auxiliar de autenticação usando o Amazon Cognito, declare a classe auxiliar da seguinte forma:

```
var authHelper = AuthHelper(applicationContext)
var locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

Você pode criar uma instância do cliente Amazon Location usando o provedor de credenciais de localização e fazer chamadas para o serviço Amazon Location. O exemplo a seguir pesquisa lugares próximos a uma latitude e longitude especificadas.

```
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 de autenticação móvel para iOS

Esses utilitários ajudam você a se autenticar ao fazer chamadas de API do Amazon Location Service a partir de seus aplicativos iOS. Isso ajuda especificamente ao usar o [Amazon Cognito](#) ou [as chaves de API](#) como método de autenticação.

O SDK de autenticação móvel do iOS está disponível no github: [Amazon Location Service Mobile Authentication SDK](#) for iOS.

### Instalação

Instale o SDK em um projeto do Xcode:

1. Vá para Arquivo e selecione Add Package Dependencies no seu projeto do XCode.
2. Digite o URL do pacote: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/> na barra de pesquisa e pressione a tecla enter.
3. Selecione o amazon-location-mobile-auth-sdk-ios pacote e pressione Add Package.
4. Selecione o produto AmazonLocationiOSAuthSDK embalado e pressione Add Package.

### Funções de autenticação

O SDK auxiliar de autenticação tem as seguintes funções:

- `authHelper.authenticateWithApiKey("My-Amazon-Location-API-Key"): LocationCredentialsProvider`: essa função retorna uma `LocationCredentialsProvider` inicializada para funcionar com uma chave de API.

- `authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")`: `LocationCredentialsProvider`: Essa função retorna um `LocationCredentialsProvider` inicializado para funcionar com um pool de identidades do Amazon Cognito.

## Uso

Para usar o SDK de autenticação móvel, adicione as seguintes declarações à sua atividade:

```
import AmazonLocationiOSAuthSDK
import AWSLocationXCF
```

Você tem duas opções ao criar as instâncias do auxiliar de autenticação e do provedor do cliente de localização. Você pode criar uma instância usando as [chaves da API de localização da Amazon](#) ou o [Amazon Cognito](#).

- Para criar uma instância auxiliar de autenticação usando uma chave de API de localização da Amazon, declare a classe auxiliar da seguinte forma:

```
let authHelper = AuthHelper()
let locationCredentialsProvider = authHelper.authenticateWithAPIKey(apiKey: "My-Amazon-Location-API-Key", region: "account-region")
```

- Para criar uma instância auxiliar de autenticação usando o Amazon Cognito, declare a classe auxiliar da seguinte forma:

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
    authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Amazon-Location-API-Key", region: "account-region")
```

Você pode criar uma instância do cliente Amazon Location usando o provedor de credenciais de localização e fazer chamadas para o serviço Amazon Location. O exemplo a seguir pesquisa lugares próximos a uma latitude e longitude especificadas.

```
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)
```

## SDK de rastreamento móvel para Android

O SDK de rastreamento móvel Amazon Location fornece utilitários que ajudam a autenticar, capturar posições de dispositivos e enviar atualizações de posição para Amazon Location Trackers com facilidade. O SDK oferece suporte à filtragem local de atualizações de localização com intervalos de atualização configuráveis. Isso reduz os custos de dados e otimiza a conectividade intermitente para seus aplicativos Android.

O SDK de rastreamento do Android está disponível em GitHub: [Amazon Location Mobile Tracking SDK](#) para Android. Além disso, tanto o SDK de autenticação móvel quanto o AWS SDK estão disponíveis no repositório [AWS Maven](#). O SDK de rastreamento do Android foi projetado para funcionar com o AWS SDK geral.

Esta seção aborda os seguintes tópicos do SDK para Android de rastreamento móvel de localização da Amazon:

### Tópicos

- [Instalação](#)
- [Uso](#)
- [Filtros](#)
- [Funções de rastreamento do Android Mobile SDK](#)
- [Exemplos](#)

### Instalação

Para instalar o SDK, adicione as seguintes linhas à seção de dependências do seu arquivo build.gradle no Android Studio:

```
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")
```

### Uso

Este procedimento mostra como usar o SDK para autenticar e criar o LocationTracker objeto:

**Note**

Esse procedimento pressupõe que você tenha importado a biblioteca mencionada na [Instalação](#) seção.

1. Importe as seguintes classes em seu código:

```
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
```

2. Em seguida, crie um `AuthHelper`, pois o `LocationCredentialsProvider` parâmetro é necessário para criar um `LocationTracker` objeto:

```
// Create an authentication helper using credentials from Cognito
val authHelper = AuthHelper(applicationContext)
val locationCredentialsProvider : LocationCredentialsProvider =
    authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

3. Agora, use `LocationCredentialsProvider` e `LocationTrackerConfig` para criar um `LocationTracker` objeto:

```
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,
)
```

## Filtros

O SDK para Android de rastreamento móvel de localização da Amazon tem três filtros de localização embutidos.

- `TimeLocationFilter`: filtra a localização atual a ser carregada com base em um intervalo de tempo definido.
- `DistanceLocationFilter`: filtra as atualizações de localização com base em um limite de distância especificado.
- `AccuracyLocationFilter`: filtra as atualizações de localização comparando a distância percorrida desde a última atualização com a precisão da localização atual.

Este exemplo adiciona filtros no `LocationTracker` momento da criação:

```
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,
)
```

Este exemplo ativa e desativa o filtro em tempo de execução com `LocationTracker`:

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

## Funções de rastreamento do Android Mobile SDK

O SDK de rastreamento móvel Amazon Location para Android inclui as seguintes funções:

- Classe: `LocationTracker`

```
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, trackerName: String), ou
constructor(context: Context, locationCredentialsProvider:
LocationCredentialsProvider, clientConfig: LocationTrackerConfig)
```

Essa é uma função inicializadora para criar um `LocationTracker` objeto. Requer instâncias de `trackerName` e `LocationCredentialsProvider`, opcionalmente, uma instância de `LocationTrackingConfig`. Se a configuração não for fornecida, ela será inicializada com valores padrão.

- Classe: `LocationTracker`

```
start(locationTrackingCallback: LocationTrackingCallback)
```

Inicia o processo de acessar a localização do usuário e enviá-la para um rastreador de localização da Amazon.

- Classe: `LocationTracker`

```
isTrackingInForeground()
```

Verifica se o rastreamento de localização está em andamento.

- Classe: `LocationTracker`

```
stop()
```

Interrompe o processo de rastreamento da localização do usuário.

- Classe: `LocationTracker`

```
startTracking()
```

Inicia o processo de acessar a localização do usuário e enviá-la ao AWS rastreador.

- Classe: `LocationTracker`

```
startBackground(mode: BackgroundTrackingMode, serviceCallback:
ServiceCallback)
```

Inicia o processo de acessar a localização do usuário e enviá-la ao AWS rastreador enquanto o aplicativo está em segundo plano. `BackgroundTrackingMode` tem as seguintes opções:

- `ACTIVE_TRACKING`: essa opção rastreia ativamente as atualizações de localização de um usuário.
- `BATTERY_SAVER_TRACKING`: essa opção rastreia as atualizações de localização do usuário a cada 15 minutos.
- Classe: `LocationTracker`

```
stopBackgroundService()
```

Interrompe o processo de acessar a localização do usuário e enviá-la ao AWS rastreador enquanto o aplicativo está em segundo plano.

- Classe: `LocationTracker`

```
getTrackerDeviceLocation()
```

Recupera a localização do dispositivo dos serviços de localização da Amazon.

- Classe: `LocationTracker`

```
getDeviceLocation(locationTrackingCallback: LocationTrackingCallback?)
```

Recupera a localização atual do dispositivo do cliente do provedor de localização fundido e a carrega no Amazon Location Tracker.

- Classe: `LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback:  
LocationTrackingCallback?)
```

Carrega a localização do dispositivo para os serviços de localização da Amazon após a filtragem com base nos filtros de localização configurados.

- Classe: `LocationTracker`

```
enableFilter(filter: LocationFilter)
```

Ativa um filtro de localização específico.

- Classe: `LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```



Desative o filtro de localização específico.

- Classe: `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())
```

Isso inicializa o `LocationTrackerConfig` com valores de parâmetros definidos pelo usuário. Se um valor de parâmetro não for fornecido, ele será definido como um valor padrão.

- Classe: `LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation:
LocationEntry?): Boolean
```

`LocationFilter` é um protocolo que os usuários podem implementar para a implementação de filtros personalizados. Você precisa implementar a `shouldUpload` função para comparar a localização anterior e atual e retornar se a localização atual deve ser carregada.

## Exemplos

O exemplo de código a seguir mostra a funcionalidade do SDK de rastreamento móvel.

Este exemplo usa o `LocationTracker` para iniciar e interromper o rastreamento em segundo plano:

```
// 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()
```

## SDK de rastreamento móvel para iOS

O SDK de rastreamento móvel Amazon Location fornece utilitários que ajudam a autenticar, capturar posições de dispositivos e enviar atualizações de posição para Amazon Location Trackers com facilidade. O SDK oferece suporte à filtragem local de atualizações de localização com intervalos de atualização configuráveis. Isso reduz os custos de dados e otimiza a conectividade intermitente para seus aplicativos iOS.

O SDK de rastreamento do iOS está disponível em GitHub: [Amazon Location Mobile Tracking SDK](#) para iOS.

Esta seção aborda os seguintes tópicos do SDK para iOS de rastreamento móvel de localização da Amazon:

### Tópicos

- [Instalação](#)
- [Uso](#)
- [Filtros](#)
- [Funções de rastreamento do SDK móvel para iOS](#)
- [Exemplos](#)

### Instalação

Use o procedimento a seguir para instalar o SDK de rastreamento móvel para iOS:

1. No seu projeto do Xcode, acesse Arquivo e selecione Add Package Dependencies.
2. Digite o seguinte URL: <https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/> na barra de pesquisa e pressione a tecla enter.
3. Selecione o amazon-location-mobile-tracking-sdk-ios pacote e clique em Add Package.

4. Selecione o produto AmazonLocationiOSTrackingSDK do pacote e clique em Add Package.

## Uso

O procedimento a seguir mostra como criar um auxiliar de autenticação usando credenciais do Cognito.

1. Depois de instalar a biblioteca, você precisa adicionar uma ou ambas as descrições ao seu `info.plist` arquivo:

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

2. Em seguida, importe o AuthHelper em sua classe:

```
import AmazonLocationiOSError import AmazonLocationiOSTrackingSDK
```

3. Em seguida, você criará um AuthHelper objeto e o usará com o AWS SDK, criando um auxiliar de autenticação usando credenciais do Amazon Cognito.

```
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)
```

## Filtros

O SDK para iOS de rastreamento móvel de localização da Amazon tem três filtros de localização embutidos.

- **TimeLocationFilter**: filtra a localização atual a ser carregada com base em um intervalo de tempo definido.
- **DistanceLocationFilter**: filtra as atualizações de localização com base em um limite de distância especificado.
- **AccuracyLocationFilter**: filtra as atualizações de localização comparando a distância percorrida desde a última atualização com a precisão da localização atual.

Este exemplo adiciona filtros no `LocationTracker` momento da criação:

```
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,
)
```

Este exemplo ativa e desativa o filtro em tempo de execução com `LocationTracker`:

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

## Funções de rastreamento do SDK móvel para iOS

O SDK de rastreamento móvel Amazon Location para iOS inclui as seguintes funções:

- Classe: `LocationTracker`

```
init(provider: LocationCredentialsProvider, trackerName: String, config: LocationTrackerConfig? = nil)
```

Essa é uma função inicializadora para criar um `LocationTracker` objeto. Requer instâncias de `trackerName` e `LocationCredentialsProvider`, opcionalmente, uma instância de `LocationTrackingConfig`. Se a configuração não for fornecida, ela será inicializada com valores padrão.

- Classe: `LocationTracker`

```
setTrackerConfig(config: LocationTrackerConfig)
```

Isso define a configuração do Tracker para entrar em vigor a qualquer momento após a inicialização do rastreador de localização

- Classe: `LocationTracker`

```
getTrackerConfig()
```

Isso faz com que a configuração de rastreamento de localização seja usada ou modificada em seu aplicativo.

Devoluções: `LocationTrackerConfig`

- Classe: `LocationTracker`

```
getDeviceId()
```

Obtém o ID do dispositivo gerado pelo rastreador de localização.

Devoluções: `String?`

- Classe: `LocationTracker`

```
startTracking()
```

Inicia o processo de acessar a localização do usuário e enviá-la ao AWS rastreador.

- Classe: `LocationTracker`

```
resumeTracking()
```

Retoma o processo de acessar a localização do usuário e enviá-la ao AWS rastreador.

- Classe: `LocationTracker`

```
stopTracking()
```

Interrompe o processo de rastreamento da localização do usuário.

- Classe: `LocationTracker`

```
startBackgroundTracking(mode: BackgroundTrackingMode)
```

Inicia o processo de acessar a localização do usuário e enviá-la ao AWS rastreador enquanto o aplicativo está em segundo plano. `BackgroundTrackingMode` tem as seguintes opções:

- `Active`: Essa opção não pausa automaticamente as atualizações de localização.
  - `BatterySaving`: Essa opção pausa automaticamente as atualizações de localização.
  - `None`: Em geral, essa opção desativa as atualizações de localização em segundo plano.
- Classe: `LocationTracker`

```
resumeBackgroundTracking(mode: BackgroundTrackingMode)
```

Retoma o processo de acessar a localização do usuário e enviá-la ao AWS rastreador enquanto o aplicativo está em segundo plano.

- Classe: `LocationTracker`

```
stopBackgroundTracking()
```

Interrompe o processo de acessar a localização do usuário e enviá-la ao AWS rastreador enquanto o aplicativo está em segundo plano.

- Classe: `LocationTracker`

```
getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil,  
endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse,  
Error>)
```

Recupera os locais de rastreamento enviados para o dispositivo do usuário entre a data e hora de início e término.

Devoluções: Void

- Classe: LocationTrackerConfig

```
init()
```

Isso inicializa o LocationTrackerConfig com valores padrão.

- Classe: 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)
```

Isso inicializa o LocationTrackerConfig com valores de parâmetros definidos pelo usuário. Se um valor de parâmetro não for fornecido, ele será definido como um valor padrão.

- Classe: LocationFilter

```
shouldUpload(currentLocation: LocationEntity, previousLocation: LocationEntity?, trackerConfig: LocationTrackerConfig)
```

LocationFilter é um protocolo que os usuários podem implementar para a implementação de filtros personalizados. Um usuário precisaria implementar uma shouldUpload função para comparar a localização anterior e atual e retornar se a localização atual deveria ser carregada.

## Exemplos

Esta seção detalha exemplos do uso do Amazon Location Mobile Tracking SDK para iOS.

### Note

Certifique-se de que as permissões necessárias estejam definidas no info.plist arquivo. Essas são as mesmas permissões listadas na [Uso](#) seção.

O exemplo a seguir demonstra a funcionalidade para rastrear a localização do dispositivo e recuperar os locais rastreados:

Privacy - Location When In Use Usage Description  
Privacy - Location Always and When In Use Usage Description

Comece a rastrear a localização:

```
do {
    try locationTracker.startTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app
    settings
    }
}
```

Retome o rastreamento da localização:

```
do {
    try locationTracker.resumeTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
}
```

Pare de rastrear a localização:

```
locationTracker.stopTracking()
```

Inicie o rastreamento em segundo plano:

```
do {
    locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
}
```

Retomar o rastreamento em segundo plano:

```
do {
```



```
locationTracker.resumeBackgroundTracking(mode: .Active)
}
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
}
```

Para interromper o rastreamento em segundo plano:

```
locationTracker.stopBackgroundTracking()
```

Recupere os locais rastreados do dispositivo a partir do rastreador:

```
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 Location

O Amazon Location Service fornece operações de API para acessar programaticamente a funcionalidade de localização. Isso inclui APIs para mapas, locais, rotas, rastreadores, geocercas e marcação de seus recursos. Para obter informações sobre as ações de API disponíveis, consulte a [referência da API do Amazon Location Service](#).

Você pode encontrar exemplos no capítulo [Exemplos de código](#) deste guia.

## Usando o Amazon Location com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que facilitam aos desenvolvedores a criação de AWS aplicativos em seu idioma preferido.

Para obter mais informações sobre os SDKs disponíveis para uso com o Amazon Location Service por idioma, consulte [SDKs por linguagem](#) neste guia.

### Versões do SDK

Recomendamos que você use a versão mais recente do AWS SDK e quaisquer outros SDKs usados em seus projetos e que mantenha os SDKs atualizados. O AWS SDK fornece os recursos e funcionalidades mais recentes, além de atualizações de segurança. Para encontrar a versão mais recente do AWS SDK JavaScript, por exemplo, consulte o tópico de [instalação do navegador](#) no AWS SDK para JavaScript ver a documentação.

## Atualizações de mensagens de erro da API do Amazon Location

A partir de 1.º de agosto de 2023, a equipe do Amazon Location está alterando as mensagens de erro da API conforme descrito nas tabelas a seguir. Os códigos de erro não serão alterados. Se seus aplicativos dependerem de strings de mensagens de erro exatas, você deverá atualizar seus aplicativos com as novas strings de caracteres. Para obter ajuda com dúvidas ou problemas, entre em contato com AWS Support.

### Tópicos

- [Locais](#)
- [Mapas](#)
- [Rastreadores](#)
- [Rotas](#)
- [Metadados](#)
- [Geocercas](#)

## Locais

### Locais

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 has IntendedUse <IntendedUse>	A place index with 'IntendedUse' set to Storage does not support 'SearchPlaceIndexForSuggestions' operation.
400	ValidationException	only one of 'BiasPosition' or 'FilterBBox' may be set	Only one of 'BiasPosition' or 'FilterBBox' may be set.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://wikipedia.org/wiki/IETF_language_tag">https://wikipedia.org/wiki/IETF_language_tag</a> .
400	ValidationException	'placeID' is invalid	'PlaceId' 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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Place index resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Place index already exists: <PlaceIndexName>.

## Mapas

### Mapas

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
		unable to find style template  Error fetching style  was not able to serialize the map style file	
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html</a> .



Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html</a> .
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html">https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html</a> .
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'.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html">https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html</a> .
402	ServiceQuotaExceededException	Map resources per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Map resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Map already exists: <MapName>.

## Rastreadores

### Rastreadores

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
500	InternalServerErrorException	Internal Server Exception  internal server error  unable to retrieve point from the storage  unable to verify tracker  Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	tracker not found: <TrackerName>  Tracker with name <TrackerName> was not found	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	'ConsumerArn' must refer to a resource

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
		in the same region as the tracker it is associated to	in the same region as the tracker it is associated with.
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is 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

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
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 <a href="https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html</a> .



Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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 <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
402	ServiceQuotaExceededException	Trackers per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Tracking resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	association already exists: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn>	An association already exists between tracker <TrackerName> and consumer <ConsumerArn>.
409	ConflictException	Tracker already exists: <TrackerName>	Tracker already exists: <TrackerName>.

## Rotas

### Rotas

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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
400	ValidationException	'DepartNow' may not be true if 'DepartureTime' is set	Only one of 'DepartNow' or 'DepartureTime' may be set.
400	ValidationException	'<TravelModeOption>' may not be set when 'TravelMode' has value <TravelModeOption>	'<TravelModeOption>' must not be set when 'TravelMode' has value <TravelModeOption>.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.Dimensions: Height, Length, Width.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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>.



Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
400	ValidationException	No route found	No route found. For more information, see <a href="https://developer.amazon.com/documentation/routing-api/dev_guide/topics/notice.html">https://developer.amazon.com/documentation/routing-api/dev_guide/topics/notice.html</a> .
400	ValidationException	No route found	No route found.
402	ServiceQuotaExceededException	Route calculators per account exceeded quota limits. For more info, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Route calculator resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .
409	ConflictException	Resource already exists	Route calculator already exists: <RouteCalculatorName>.

## Metadados

### Metadados

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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	'ExpireTime' must not be in the past.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
		be in the past when creating a key	
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

## Geocercas

### Geocercas

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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>	Internal server error. Try again later.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
404	ResourceNotFoundException	collection not found: <GeofenceCollectionName>  <GeofenceCollectionName> geofence collection not found  Resource not found error  no geofence with given name found	Geofence Collection not found: <GeofenceCollectionName>.
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 <a href="https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk">https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk</a> .
400	ValidationException	Invalid token	'NextToken' must be a valid token.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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'
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 <a href="https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html</a> .
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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>.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.



Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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 <a href="https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3">https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3</a> .
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.
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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
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.
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.R adius' 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.

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see <a href="https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html">https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html</a> .
400	ValidationException	arn: not enough sections	'SourceArn' must be a valid ARN. For more information, see <a href="https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html">https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html</a> .
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 <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a>	Geofence collection resources have exceeded the quota per account per region. For more information, see <a href="https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/">https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/</a> .

Código de erro	Exceção	Mensagem de erro antiga	Nova mensagem de erro
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

## Exemplos de código e tutoriais para trabalhar com o Amazon Location Service

Este tópico mostra uma lista de exemplos de códigos, tutoriais e postagens em blogs para ajudar você a aprender sobre o Amazon Location Service. Cada exemplo de código inclui uma descrição de como ele funciona.

Você pode encontrar amostras adicionais na página [AWS Geospatial, na GitHub página](#) de [AWS amostras GitHub da Amazon Location](#) e no [site do AWS blog](#).

### Note

É bom entender a diferença entre a página AWS Geoespacial e a GitHub página de AWS amostras GitHub .

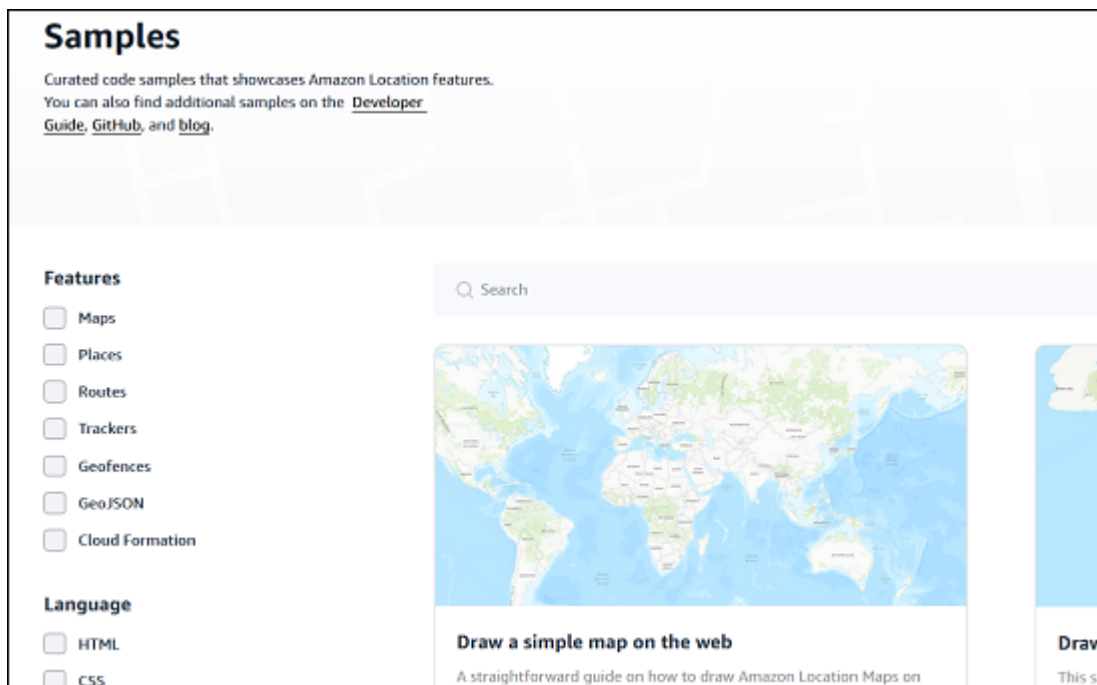
- Geoespacial GitHub — A [GitHub página AWS Geoespacial](#) inclui amostras que são criadas e mantidas pela equipe do Amazon Location Service.
- Amostras GitHub — A [GitHub página de AWS amostras do Amazon Location](#) inclui amostras que foram criadas para o Amazon Location, mas podem ou não ser mantidas ativamente.

O tutorial de [início rápido](#) é um bom ponto de partida a ser considerado antes de usar outras amostras, pois mostra como preencher os pré-requisitos que são úteis para a maioria das amostras.

## Tópicos

- [Site de demonstração do Amazon Location](#)
- [Tutorial: início rápido](#)
- [Tutorial: enriquecimento do banco de dados](#)
- [Exemplo: explore o aplicativo](#)
- [Exemplo: estilize um mapa](#)
- [Exemplo: desenhe marcadores](#)
- [Exemplo: desenhe pontos agrupados](#)
- [Exemplo: desenhe um polígono](#)
- [Exemplo: altere o idioma do mapa](#)
- [Blog: Estimated delivery time notifications \(Notificações de tempo estimado de entrega\)](#)
- [Exemplo: atualizações da posição do fluxo](#)
- [Exemplo: aplicativo móvel de geofencing and tracking](#)

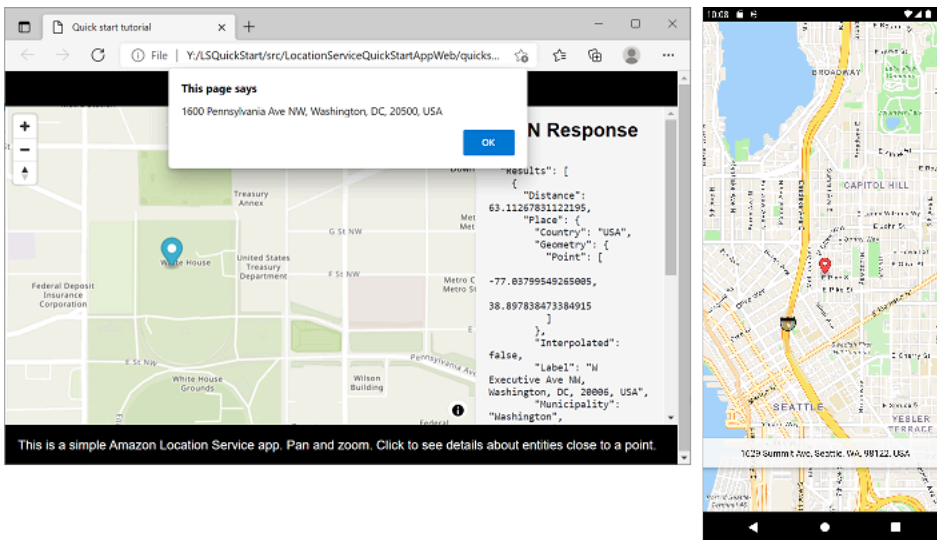
## Site de demonstração do Amazon Location



Você pode ver demonstrações com o código-fonte do Amazon Location Service em ação no [site de demonstração do Amazon Location](#). Esse site inclui uma [demonstração hospedada na Web](#) e também um aplicativo de demonstração para [Android](#).

Você também pode encontrar uma grande variedade de amostras, filtráveis por atributo, linguagem e plataforma na página de [Amostras](#) do site.

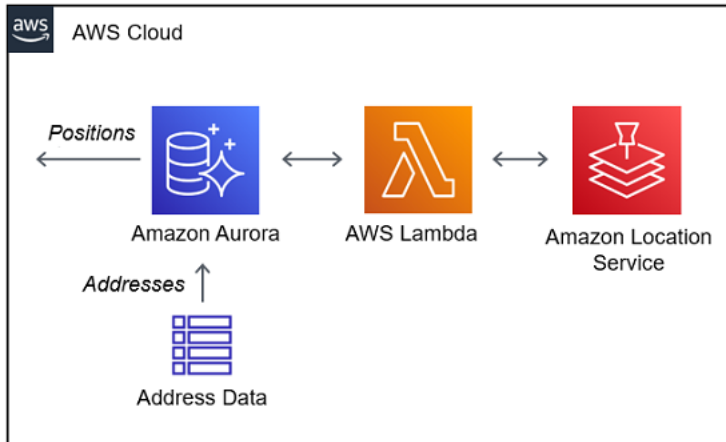
## Tutorial: início rápido



Há tutoriais de início rápido disponíveis para dispositivos web, iOS e Android. Para cada plataforma, o tutorial mostra como adicionar um mapa interativo a um aplicativo e como fazer chamadas para as APIs do Amazon Location Service a partir do seu aplicativo. O tutorial está disponível JavaScript em uma página da web estática, Kotlin para um aplicativo de telefone Android ou Swift para um aplicativo iOS.

- JavaScript para um link estático de documentação da página da web: [Criar um aplicativo Web](#)
- Link de documentação do Kotlin para um aplicativo Android: [Início rápido com o Amazon Location Service](#)
- Link de documentação do Swift para um aplicativo iOS: [Criação de um aplicativo para iOS](#)

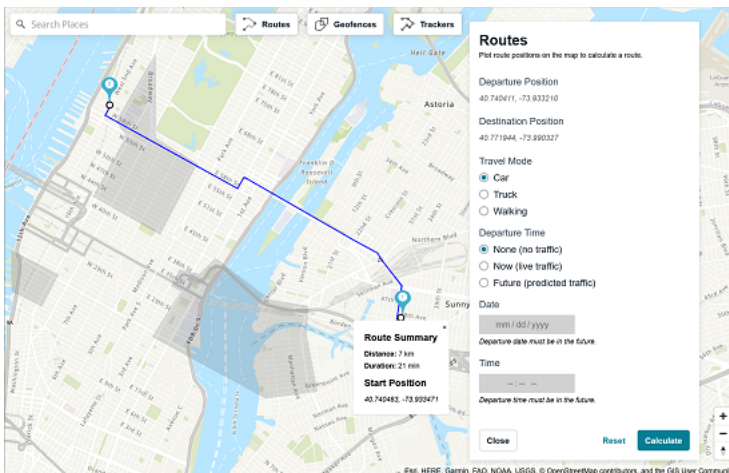
## Tutorial: enriquecimento do banco de dados



Este tutorial mostra como usar o Amazon Location Service, chamado de AWS Lambda para normalizar endereços e adicionar latitude e longitude aos registros em um banco de dados Amazon Aurora. Usa Amazon Aurora e AWS Lambda

Link da documentação: [Amazon Aurora PostgreSQL funções definidas pelo usuário para o Amazon Location Service](#)

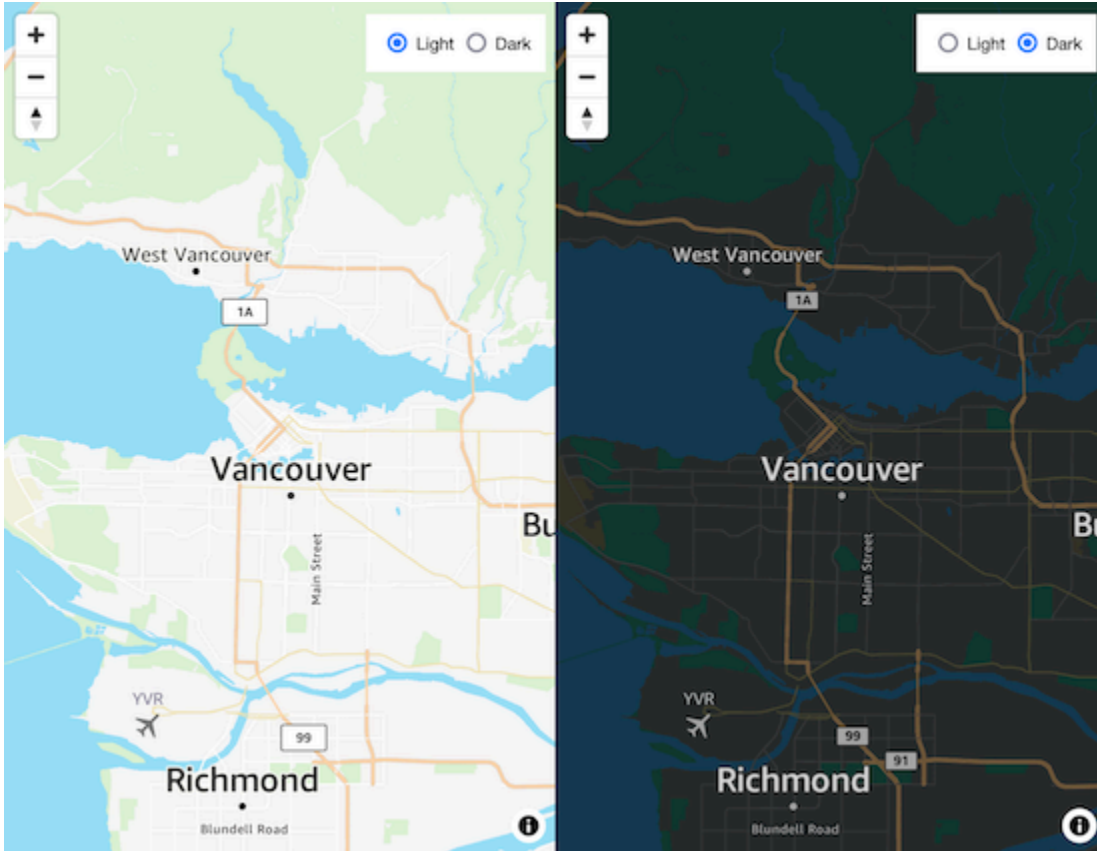
## Exemplo: explore o aplicativo



Uma das melhores maneiras de aprender sobre a funcionalidade do Amazon Location Service é usar a [funcionalidade Explore](#) no console do Amazon Location. Este exemplo completo de aplicativo web imita a funcionalidade de mapas, lugares, rotas, geocercas e rastreadores do console para mostrar como recriar esses atributos em seu próprio aplicativo. Usa Amplify, React e JavaScript

GitHub Link de amostras: [Explore o aplicativo de amostra](#)

## Exemplo: estilize um mapa



Este exemplo de código mostra como alternar entre um mapa de satélite e um roteiro vetorial, usando MapLibre in JavaScript. Usa MapLibre, o assistente de autenticação de localização da Amazon e JavaScript

GitHub Link geoespacial: [mapa interativo com troca de estilo](#)

## Exemplo: desenhe marcadores

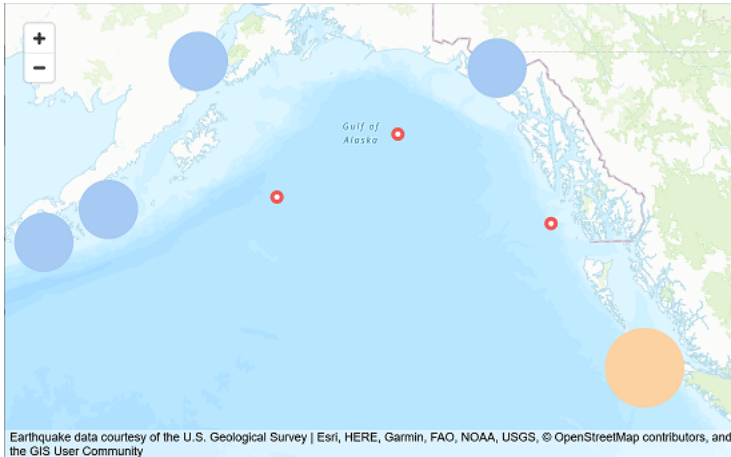




Este exemplo de código mostra os locais do Amazon Locker em Vancouver, BC, Canadá. Mostra como desenhar marcadores nas localizações dos pontos. Usa Node.js MapLibre, React, o auxiliar de autenticação Amazon Location e. JavaScript

GitHub Link geoespacial: [mapa interativo com marcadores](#) em pontos

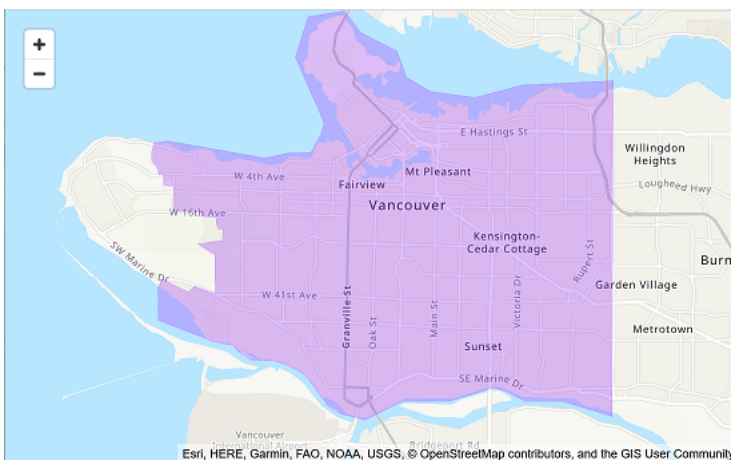
## Exemplo: desenhe pontos agrupados



Usando dados de terremotos do USGS, este exemplo de código mostra como desenhar pontos que cluster quando estão próximos no mapa. Usa MapLibre, Node.js, React, Amplify e. JavaScript

GitHub Link de amostras: [Mapa interativo com grupos de pontos](#)

## Exemplo: desenhe um polígono



Este exemplo de código mostra como desenhar um polígono no mapa. Usa Node.js MapLibre, React, o auxiliar de autenticação Amazon Location e. JavaScript

GitHub Link geoespacial: [mapa interativo com polígonos](#)

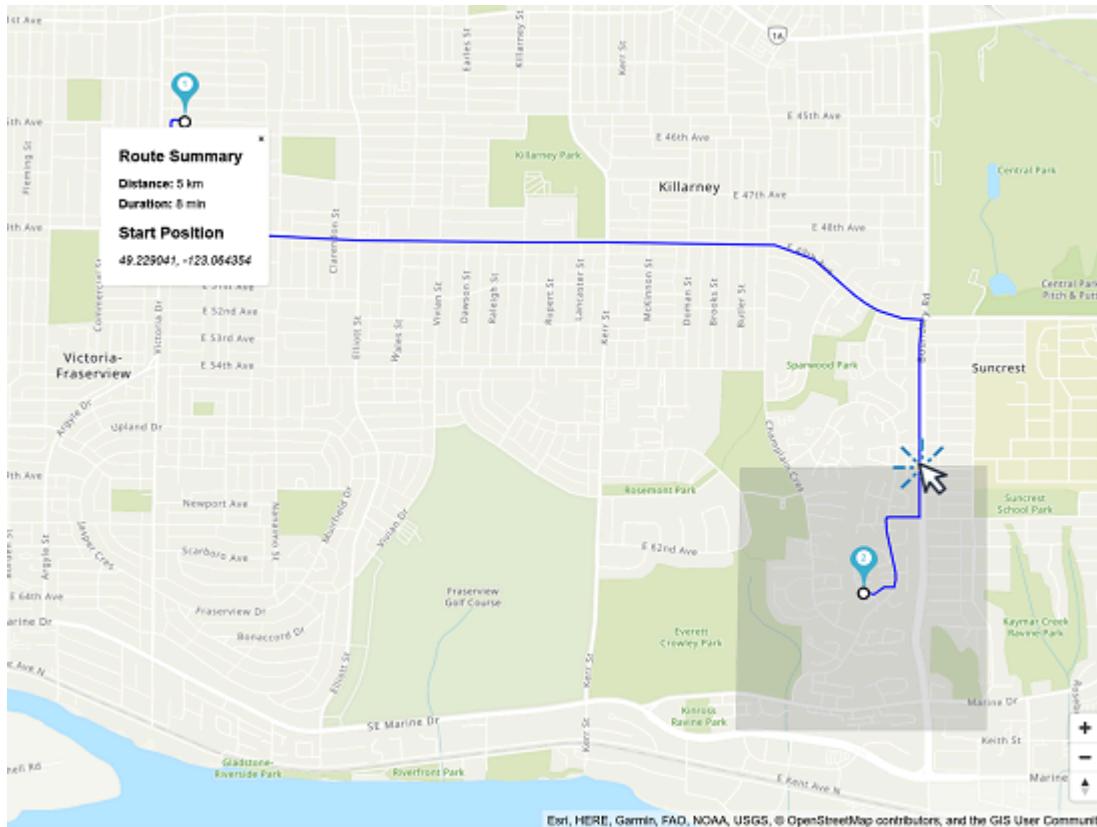
## Exemplo: altere o idioma do mapa



Este exemplo de código mostra como você pode alterar o idioma de exibição dos mapas no Amazon Location. Usa Amplify, React e MapLibre

GitHub Link de amostras: [Alterar amostra de idioma do mapa](#)

## Blog: Estimated delivery time notifications (Notificações de tempo estimado de entrega)



Esta postagem do blog mostra maneiras diferentes de notificar os clientes com prazos de entrega estimados. Ela explica o uso de rotas para mostrar o tempo estimado da viagem e, em seguida, o uso de rastreadores e geocercas para notificar quando um motorista se aproxima do cliente. Usa Amplify, React, Amazon e Amazon EventBridge Simple Notification Service (Amazon SNS).

Link do blog: [Estimated Time of Arrival and Proximity Notifications](#) (Tempo estimado de chegada e notificações de proximidade)

## Exemplo: atualizações da posição do fluxo



Aplicativo Kinesis Stream To Tracker: este exemplo demonstra como usar o Kinesis Data Stream para publicar atualizações do rastreador com o Amazon Location Service. O exemplo é um aplicativo lambda implantável escrito em python que pode ser integrado a um Kinesis Data Stream para consumir os eventos do Kinesis e atualizar as posições do dispositivo em lote.

Link do repositório: [Amazon Location Amazon Kinesis Data Streams Stream To Tracker App](#)

Para obter mais informações sobre rastreamento e cercas geográficas, consulte a documentação de cercas [geográficas e rastreadores](#). [Os desenvolvedores podem implantar o aplicativo seguindo a documentação do Serverless Application Repository da AWS ou diretamente do console Lambda.](#)

Aplicativo de amostra de streaming de posição do dispositivo: esse exemplo de código mostra como transmitir dados de posição do dispositivo para um Kinesis Data Stream e como as notificações de geofence funcionam. Esse aplicativo depende do aplicativo de amostra Kinesis Stream to Tracker, listado acima, para que as posições do rastreador transmitido sejam atualizadas no Amazon Location Service.

Link do repositório: [Amazon Location Device Position Streaming Sample App](#)

## Exemplo: aplicativo móvel de geofencing and tracking

Este exemplo de aplicativo mostra como um rastreador e uma cerca geográfica interagem usando uma combinação dos recursos Lambda AWS IoT e Amazon Location. Há tutoriais disponíveis para iOS e Android.

Link do tutorial: [Exemplo de aplicativo móvel Geofence and Tracker](#)

# Como usar o Amazon Location Service

Você pode usar os recursos do Amazon Location Service para concluir tarefas geográficas e relacionadas à localização. Em seguida, você pode combinar essas tarefas para abordar casos de uso mais complexos, como geomarketing, entrega e rastreamento de ativos.

Quando você estiver pronto para criar recursos de localização em seu aplicativo, use os seguintes métodos para usar a funcionalidade do Amazon Location Service, dependendo de suas metas e inclinações:

- Ferramentas de exploração — Se você quiser experimentar os recursos de localização da Amazon, as seguintes ferramentas são a maneira mais rápida de acessar e testar as APIs:
  - O [console do Amazon Location](#) fornece uma variedade de ferramentas de acesso rápido. Você pode criar e gerenciar seus recursos e testar as APIs usando [a página Explorar](#). O console também é útil para criar recursos (normalmente uma tarefa única) em preparação para o uso de qualquer um dos outros métodos descritos posteriormente.
  - A [Interface de Linha de Comando \(CLI\) da AWS](#) permite criar recursos e acessar as APIs do Amazon Location usando um terminal. A AWS CLI manipula a autenticação quando você a configura com suas credenciais.
  - Você pode ver [exemplos de código e tutoriais](#) que mostram como realizar tarefas usando as APIs do Amazon Location Service. Isso inclui [um exemplo](#) que reproduz grande parte da funcionalidade da página Explorar no console.
- SDKs da plataforma — Se você não estiver visualizando dados em um mapa, poderá usar qualquer uma das [ferramentas padrão da AWS](#) para criar na AWS.
  - Os seguintes SDKs estão disponíveis: C++, Go, Java, .NET JavaScript, Node.js, PHP, Python e Ruby.
- SDKs e bibliotecas de front-end — Se você quiser usar o Amazon Location para criar um aplicativo em uma plataforma móvel ou visualizar dados em um mapa em qualquer plataforma, há as seguintes opções:
  - As AWS Amplify bibliotecas integram o Amazon Location em aplicativos [iOS](#), [Android](#) e [JavaScriptweb](#).
  - As MapLibre bibliotecas permitem renderizar mapas do lado do cliente em aplicativos [iOS](#), [Android](#) e [JavaScriptweb](#) usando o Amazon Location.

- As bibliotecas do Tangram ES permitem que você renderize mapas 2D e 3D a partir de dados vetoriais usando o OpenGL ES em aplicativos da web [iOS](#) e [Android](#). Também existe o Tangram para aplicativos [JavaScript](#)web.
- Envio de solicitações HTTPS diretas — Se você estiver trabalhando com uma linguagem de programação para a qual não há SDK disponível, ou se quiser ter mais controle sobre como enviar uma solicitação para AWS, pode acessar o Amazon Location enviando solicitações HTTPS diretas autenticadas pelo processo de assinatura Signature versão 4. Para ter mais informações, consulte [Processo de assinatura do Signature versão 4](#) no Referência geral da AWS.

Este capítulo descreve muitas das tarefas comuns aos aplicativos que usam dados de localização. A seção de [casos de uso comuns](#) descreve como combiná-los com outros serviços da AWS para obter casos de uso mais complexos.

## Tópicos

- [Pré-requisitos para usar o Amazon Location Service](#)
- [Usando o Amazon Location Maps em seu aplicativo](#)
- [Pesquisando dados de localização e geolocalização usando Amazon Location](#)
- [Calcular rotas usando o Amazon Location Service](#)
- [Colocando geocercas em uma área de interesse usando o Amazon Location](#)
- [Marcar os recursos do Amazon Location Service](#)
- [Conceder acesso ao Amazon Location Service](#)
- [Monitorar o Amazon Location Service](#)
- [Criação de recursos Amazon Location Service com AWS CloudFormation](#)

## Pré-requisitos para usar o Amazon Location Service

Esta seção descreve o que você precisa fazer para usar o Amazon Location Service. Você deve ter uma Conta da AWS e ter configurado o acesso ao Amazon Location para os usuários que desejam usá-lo.

### Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

## Para se inscrever em um Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

## Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

### Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Signing in as the root user](#) (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

### Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.



Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

### Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

### Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

## Conceder acesso ao Amazon Location Service

Por padrão, os usuários não administrativos não têm nenhuma permissão. Antes que eles possam acessar o Amazon Location, você deve conceder permissão anexando uma política do IAM com permissões específicas. Certifique-se de seguir o princípio de privilégio mínimo ao conceder acesso a recursos.

**Note**

Para obter informações sobre como dar aos usuários não autenticados acesso à funcionalidade do Amazon Location Service (por exemplo, em um aplicativo baseado na Web), consulte [Conceder acesso ao Amazon Location Service](#).

O exemplo de política a seguir dá permissão ao usuário para acessar todas as operações do Amazon Location. Para obter mais exemplos, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

Crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center .

- Usuários gerenciados no IAM com provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em [Criar um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do usuário do IAM.

- Usuários do IAM:

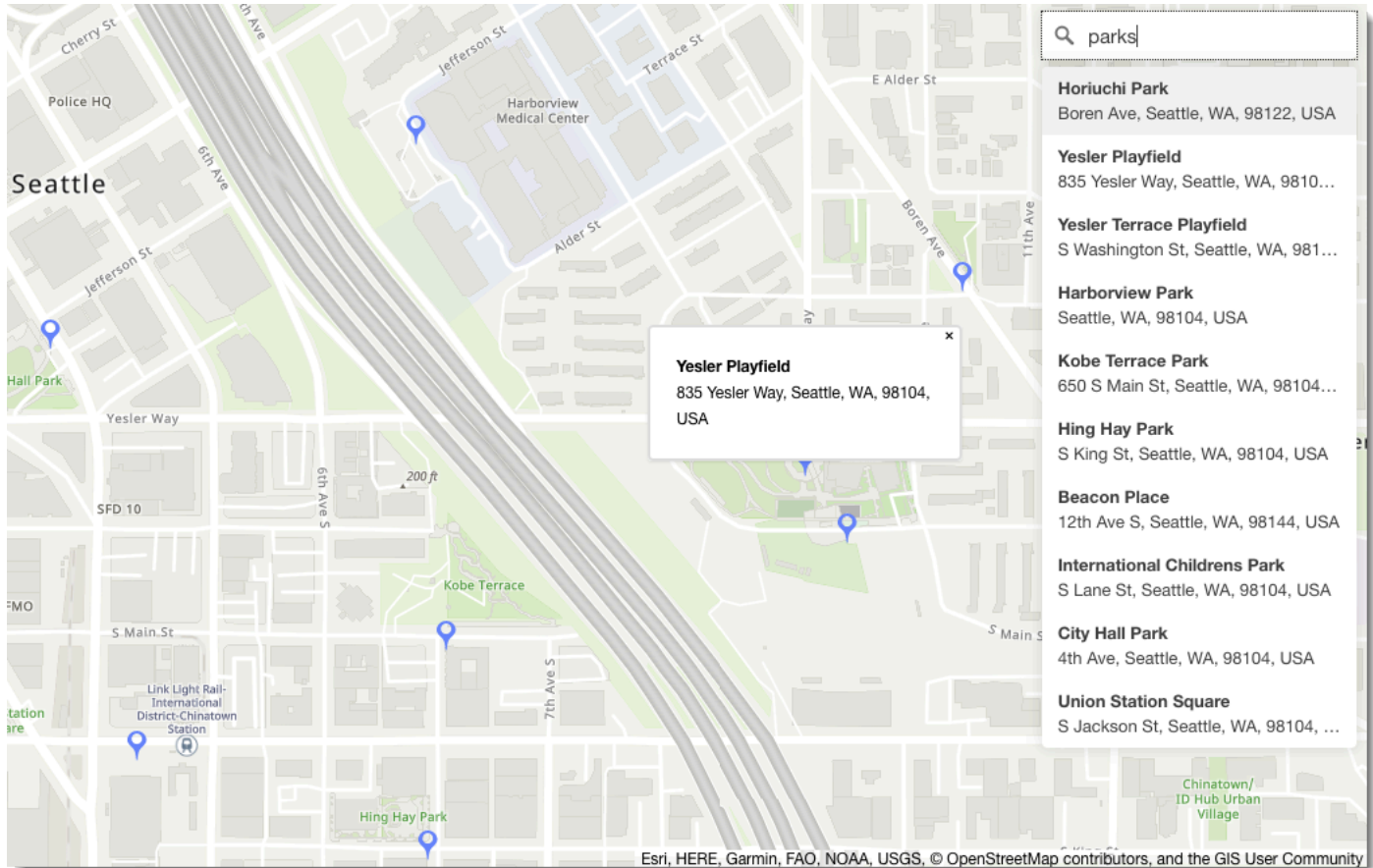
- Crie um perfil que seu usuário possa assumir. Siga as instruções em [Criação de um perfil para um usuário do IAM](#) no Guia do usuário do IAM.

- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adição de permissões a um usuário \(console\)](#) no Guia do usuário do IAM.

Ao criar aplicativos que usam o Amazon Location Service, talvez seja necessário que alguns usuários tenham acesso não autenticado. Para esses casos, consulte [Habilitando o acesso não autenticado usando o Amazon Cognito](#).

## Usando o Amazon Location Maps em seu aplicativo

Os mapas do Amazon Location são econômicos e interativos. Você pode substituir um mapa existente em seu aplicativo para economizar dinheiro, ou adicionar um novo para exibir visualmente dados baseados em localização, como a localização da sua loja.



O Amazon Location Service permite que você escolha um provedor de dados para operações de mapas criando e configurando um recurso de mapa. O recurso de mapa configura o provedor de dados e o estilo usado para renderizar o mapa.

Depois de criar seu recurso, você pode enviar solicitações usando o SDK da AWS diretamente ou usando uma biblioteca criada especificamente para renderizar mapas em seu ambiente.

### Note

Para obter uma visão geral dos conceitos de mapas, consulte [Mapas](#).

## Tópicos

- [Pré-requisitos](#)
- [Exibir um mapa em seu aplicativo](#)
- [Desenhando recursos de dados em um mapa](#)
- [Definindo extensões para um mapa usando MapLibre](#)
- [Gerenciando seus recursos de mapa](#)

## Pré-requisitos

Antes de exibir um mapa em seu aplicativo, siga as etapas de pré-requisito:

### Tópicos

- [Criar um recurso de mapa](#)
- [Autenticando suas solicitações](#)

## Criar um recurso de mapa

Para usar um mapa em seu aplicativo, você deve ter um recurso de mapa, que especifica o estilo do mapa e o provedor de dados a serem usados em seus mapas.

### Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço da AWS](#) para obter mais detalhes.

Você pode criar um recurso de mapa usando o console do Amazon Location Service, a AWS CLI, ou as APIs do Amazon Location.

## Console

Para criar um recurso de mapa usando o console do Amazon Location Service

1. No console do Amazon Location, na página [Mapas](#), selecione Criar mapa para visualizar estilos de mapa.
2. Adicione um nome e uma descrição para o novo recurso de mapa.
3. Escolha um estilo de mapa.

### Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço da AWS](#) para obter mais detalhes.

4. Escolha entre o [Visões políticas](#) para usar.
5. Aceite os Termos e Condições do Amazon Location e selecione Criar mapa. Você pode interagir com o mapa que escolheu: ampliar, reduzir o zoom ou fazer uma panorâmica em qualquer direção.
6. Para permitir que seus usuários alternem estilos (por exemplo, para permitir que eles alternem entre imagens de satélite e estilo vetorial), você deve criar um recurso de mapa para cada estilo.

Você pode excluir recursos com estilos de mapa que não deseja usar na [página inicial do Maps](#) no console.

## API

Para criar um recurso de mapa usando as APIs do Amazon Location

Use a operação [CreateMap](#) das APIs do Amazon Location.

O exemplo a seguir é uma solicitação de API para criar um recurso de mapa chamado *ExampleMap* usando o estilo de *VectorEsriStreets* mapa.

```
POST /maps/v0/maps HTTP/1.1
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

### Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço da AWS](#) para obter mais detalhes.

## AWS CLI

Para criar um recurso de mapa usando comandos do AWS CLI

Use o comando [create-map](#).

O exemplo a seguir cria um recurso de mapa chamado *ExampleMap* usando *VectorEsriStreets* as the map style.

```
aws location \
  create-map \
  --configuration Style="VectorEsriStreets" \
  --map-name "ExampleMap"
```

### Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço da AWS](#) para obter mais detalhes.

## Autenticando suas solicitações

Depois de criar um recurso de mapa e estar pronto para começar a criar recursos de localização em seu aplicativo, você precisa escolher como autenticar suas solicitações.

### Note

A maioria dos aplicativos front-end de mapas exige acesso não autenticado aos mapas ou outros recursos do Amazon Location Service. Dependendo do seu aplicativo, talvez você queira usar o AWS Signature v4 para autenticar solicitações, ou você pode usar as chaves do Amazon Cognito ou da Amazon Location API para uso não autenticado. Para saber mais sobre essas opções, consulte [Conceder acesso ao Amazon Location Service](#).

## Exibir um mapa em seu aplicativo

Esta seção fornece tutoriais sobre como usar as ferramentas de renderização de mapas para exibir um mapa em seu aplicativo móvel ou da web ao usar as APIs do Amazon Location. Conforme mencionado no [Como usar o Amazon Location Service](#) tópico, você tem uma variedade de bibliotecas para usar ao renderizar mapas com o Amazon Location, incluindo Amplify e MapLibre Tangram.

Para exibir um mapa em seu aplicativo, faça o seguinte:

- A maneira mais direta de exibir um mapa em seus aplicativos front-end móveis e web é usar MapLibre. Você pode seguir os [MapLibre tutoriais](#) ou até mesmo o [tutorial de início rápido](#) para aprender a usar. MapLibre
- Se você já é um desenvolvedor AWS Amplify, talvez queira usar o SDK do Amplify Geo. Para saber mais, siga o [tutorial do Amplify](#).
- Se você já é usuário do Tangram e quer continuar a usá-lo para renderizar seu mapa, enquanto migra para o Amazon Location Service, siga o [tutorial do Tangram](#).

### Tópicos

- [Usando a MapLibre biblioteca com o Amazon Location Service](#)
- [Usando a biblioteca Amplify com o Amazon Location Service](#)
- [Usando o Tangram com o Amazon Location Service](#)

## Usando a MapLibre biblioteca com o Amazon Location Service

Os tutoriais a seguir explicam como usar a MapLibre Biblioteca com o Amazon Location.

### Tópicos

- [Usando o MapLibre GL JS com o Amazon Location Service](#)
- [Usando o SDK MapLibre nativo para Android com o Amazon Location Service](#)
- [Usando o SDK MapLibre nativo para iOS com o Amazon Location Service](#)

### Usando o MapLibre GL JS com o Amazon Location Service

Use o [MapLibre GL JS](#) para incorporar mapas do lado do cliente em aplicativos da web.

MapLibre O GL JS é uma JavaScript biblioteca de código aberto compatível com os estilos e blocos fornecidos pela API Amazon Location Service Maps. Você pode integrar o MapLibre GL JS em um HTML ou JavaScript aplicativo básico para incorporar mapas personalizáveis e responsivos do lado do cliente.

Este tutorial descreve como integrar o MapLibre GL JS com o Amazon Location em um HTML e um JavaScript aplicativo básicos. As mesmas bibliotecas e técnicas apresentadas neste tutorial também se aplicam a estruturas, como [React](#) e [Angular](#).

O aplicativo de amostra para este tutorial está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

### Construindo o aplicativo: Estrutura

Este tutorial cria uma aplicação web que usa JavaScript para criar um mapa em uma página HTML.

Comece criando uma página HTML (`index.html`) que inclua o contêiner do mapa:

- Insira um elemento `div` com um `id` de `map` para aplicar as dimensões do mapa à visualização do mapa. As dimensões são herdadas da janela de exibição.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }
    </style>
  </head>
</html>
```



```
    #map {
      height: 100vh; /* 100% of viewport height */
    }
  </style>
</head>
<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>
```

## Construindo o aplicativo: adicionando dependências

Adicione as seguintes dependências ao aplicativo:

- MapLibre GL JS (v3.x) e seu CSS associado.
- O Amazon Location [JavaScript Auxiliar de autenticação](#).

```
<!-- 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>
```

Isso cria uma página vazia com o contêiner do mapa.

## Construindo o aplicativo: configuração

Para configurar seu aplicativo usando JavaScript:

1. Insira os nomes e identificadores dos seus recursos.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
```

```
// Amazon Location Service Map name
const mapName = "ExampleMap";
```

2. Instancie um provedor de credenciais usando o banco de identidade não autenticado que você criou em [Usando mapas – Etapa 2, Configurar a autenticação](#). Vamos colocar isso em uma função chamada `initializeMap`, que também conterá outro código de inicialização do mapa, adicionado na próxima etapa

```
// 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
}
```

## Construindo o aplicativo: inicialização do mapa

Para que o mapa seja exibido após o carregamento da página, você deve inicializar o mapa. Você pode ajustar a localização inicial do mapa, adicionar controles adicionais e sobrepor dados.

```
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

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As sequências de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `sources.grabmaptiles.attribution`. O MapLibre GL JS fornecerá automaticamente a atribuição. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

## Executando o aplicativo

Você pode executar esse aplicativo de amostra usando-o em um servidor da web local ou abrindo-o em um navegador.

Para usar um servidor da web local, você pode usar o `npx`, porque ele é instalado como parte do Node.js. Você pode usar o `npx` `serve` de dentro do mesmo diretório que `index.html`. Isso serve o aplicativo em `localhost:5000`.

### Note

Se a política que você criou para sua função não autenticada do Amazon Cognito incluir uma condição `referer`, você poderá ser impedido de testar com URLs `localhost:.` Nesse caso, você pode testar com um servidor da web que forneça uma URL que esteja na sua política.

Depois de concluir o tutorial, o aplicativo final se parece com o exemplo a seguir.

```
<!-- 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>
```

A execução deste aplicativo exibe um mapa em tela cheia usando o estilo de mapa escolhido. Essa amostra está disponível no repositório de amostras do Amazon Location Service em [GitHub](#).

### Usando o SDK MapLibre nativo para Android com o Amazon Location Service

Use o SDK [MapLibrenativo](#) para incorporar mapas interativos em seus aplicativos Android.

O SDK MapLibre nativo para Android é uma biblioteca baseada [no Mapbox Native](#) e é compatível com os estilos e blocos fornecidos pela API Amazon Location Service Maps. Você pode integrar o SDK MapLibre nativo para Android para incorporar visualizações de mapas interativas com mapas vetoriais escaláveis e personalizáveis em seus aplicativos Android.

Este tutorial descreve como integrar o SDK MapLibre nativo para Android com o Amazon Location. O aplicativo de amostra para este tutorial está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

### Construindo o aplicativo: inicialização

Para inicializar seu aplicativo:

1. Crie um novo projeto do Android Studio a partir do modelo Atividade vazia.
2. Certifique-se de que o Kotlin esteja selecionado para o idioma do projeto.
3. Selecione um SDK mínimo da API 14: Android 4.0 (Ice Cream Sandwich) ou mais recente.
4. Abra a Estrutura do projeto e vá para Arquivo > Estrutura do projeto... para escolher a seção Dependências.
5. Com <All Modules> selecionado, escolha o botão + para adicionar uma nova dependência de biblioteca.
6. Adicione o SDK para Android da AWS versão 2.20.0 ou posterior. Por exemplo:  
`com.amazonaws:aws-android-sdk-core:2.20.0`
7. Adicione o SDK MapLibre nativo para Android versão 9.4.0 ou posterior. Por exemplo:  
`org.maplibre.gl:android-sdk:9.4.0`
8. No nível do projeto do seu arquivo build.gradle, adicione o seguinte repositório maven para acessar os pacotes para Android: MapLibre

```
allprojects {  
    repositories {  
        // Retain your existing repositories
```

```
        google()
        jcenter()

        // Declare the repositories for MapLibre
        mavenCentral()
    }
}
```

## Construindo o aplicativo: configuração

Para configurar seu aplicativo com seus recursos e sua região da AWS:

```
<?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>
```

## Construindo o aplicativo: layout da atividade

Edite `app/src/main/res/layout/activity_main.xml`.

- Adicione um `MapView`, que renderiza o mapa. Isso também definirá o ponto central inicial do mapa.
- Adicione um `TextView`, que exibe a atribuição.

```
<?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

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `source.grabmaptiles.attribution`. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

## Construindo o aplicativo: Solicitar transformação

Crie uma classe nomeada `SigV4Interceptor` para interceptar solicitações da AWS e assiná-las usando o [Signature versão 4](#). Isso será registrado no cliente HTTP usado para buscar recursos do mapa quando a Atividade Principal for criada.

```
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()
    }
}
}
```

## Construindo o aplicativo: atividade principal

A Atividade Principal é responsável por inicializar as visualizações que serão exibidas aos usuários. Isso envolve:

- Instanciando um `CredentialsProvider` do Amazon Cognito.
- Registrando o interceptador `Signature` versão 4.
- Configurar o mapa apontando-o para um descritor de estilo de mapa e exibindo a atribuição apropriada.

O `MainActivity` também é responsável por encaminhar eventos do ciclo de vida para a visualização do mapa, permitindo que ela preserve a janela de visualização ativa entre as invocações.

```
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()  
}  
}
```

A execução deste aplicativo exibe um mapa em tela cheia no estilo de sua escolha. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

## Usando o SDK MapLibre nativo para iOS com o Amazon Location Service

Use o [SDK MapLibre nativo para iOS para](#) incorporar mapas do lado do cliente em aplicativos iOS.

O SDK MapLibre nativo para iOS é uma biblioteca baseada [no Mapbox GL](#) Native e é compatível com os estilos e blocos fornecidos pela API Amazon Location Service Maps. Você pode integrar o SDK MapLibre nativo para iOS para incorporar visualizações de mapa interativas com mapas vetoriais escaláveis e personalizáveis em seus aplicativos iOS.

Este tutorial descreve como integrar o SDK MapLibre nativo para iOS com o Amazon Location. O aplicativo de amostra para este tutorial está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

### Construindo o aplicativo: inicialização

Para inicializar seu aplicativo:

1. Crie um novo projeto Xcode a partir do modelo Aplicativo.
2. Selecione SwiftUI para sua interface.
3. Selecione o aplicativo SwiftUI para seu Ciclo de vida.
4. Selecione Swift como seu idioma.

### Adicionando MapLibre dependências usando pacotes Swift

Para adicionar uma dependência de pacote ao seu projeto do Xcode:

1. Navegue até Arquivo > Pacotes Swift > Adicionar dependência de pacote.
2. Insira a URL do repositório: **<https://github.com/maplibre/maplibre-gl-native-distribution>**

#### Note

Para obter mais informações sobre pacotes Swift, consulte [Adicionar dependências de pacotes ao seu aplicativo](#) em Apple.com

3. No seu terminal, instale CocoaPods:

```
sudo gem install cocoapods
```

4. Navegue até o diretório do projeto do seu aplicativo e inicialize o Podfile com o gerenciador de CocoaPods pacotes:

```
pod init
```

5. Abra o Podfile para adicionar AWSCore como dependência:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
end
```

6. Baixe e instale dependências:

```
pod install --repo-update
```

7. Abra a área de trabalho do Xcode que CocoaPods criou:

```
xed .
```

## Construindo o aplicativo: configuração

Adicione as seguintes chaves e valores ao Info.plist para configurar o aplicativo:

Chave	Valor
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd
MapName	ExampleMap

## Construindo o aplicativo: ContentView layout

Para renderizar o mapa, edite ContentView.swift:

- Adicione um `MapView` que renderiza o mapa.
- Adicione um `TextField` que exiba a atribuição.

Isso também define o ponto central inicial do mapa.

```
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

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e

`source.grabmptiles.attribution`. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

## Construindo o aplicativo: Solicitar transformação

Crie um novo arquivo Swift chamado `AWSSignatureV4Delegate.swift` contendo a seguinte definição de classe para interceptar solicitações da AWS e assiná-las usando o [Signature versão 4](#). Uma instância dessa classe será designada como delegada de armazenamento offline, que também é responsável por reescrever URLs, na visualização do mapa.

```
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
}
```

```
}
```

## Construindo o aplicativo: visualização do mapa

A visualização do Mapa é responsável por inicializar uma instância de `AWSSignatureV4Delegate` e configurar o `MGLMapView` subjacente, que busca recursos e renderiza o mapa. Ele também lida com a propagação de cadeias de atribuição da fonte do descritor de estilo de volta para o `ContentView`.

Crie um novo arquivo Swift chamado `MapView.swift` contendo a seguinte definição de `struct`:

```
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
}
```

```
}  
}
```

A execução deste aplicativo exibe um mapa em tela cheia no estilo de sua escolha. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

## Usando a biblioteca Amplify com o Amazon Location Service

O tutorial a seguir orienta você a usar AWS Amplify com a Amazon Location. Amplify usa MapLibre GL JS para renderizar mapas em seu JavaScript aplicativo baseado.

O Amplify é um conjunto de bibliotecas de cliente de código aberto que fornecem interfaces para diferentes categorias de serviços, incluindo o Amplify Geo, que é desenvolvido pelo Amazon Location Service. [Saiba mais sobre a JavaScript biblioteca AWS Amplify Geo](#).

### Note

Este tutorial pressupõe que você já tenha seguido as etapas em [Usando mapas – Para adicionar um mapa ao seu aplicativo](#).

## Construindo o aplicativo: estrutura

Este tutorial cria uma aplicação web que usa JavaScript para criar um mapa em uma página HTML.

Comece criando uma página HTML (`index.html`) que inclua o contêiner do mapa:

- Insira um elemento `div` com um `id` de `map` para aplicar as dimensões do mapa à visualização do mapa. As dimensões são herdadas da janela de exibição.

```
<html>  
  <head>  
    <style>  
      body { margin: 0; }  
      #map { height: 100vh; } /* 100% of viewport height */  
    </style>  
  </head>  
  
  <body>
```

```

    <!-- map container -->
    <div id="map" />
  </body>
</html>

```

## Construindo o aplicativo: adicionando dependências

Adicione as seguintes dependências ao aplicativo:

- AWS Amplify bibliotecas de mapas e geografias.
- AWS Amplify biblioteca principal.
- AWS Amplify biblioteca de autenticação.
- AWS Amplify folha de estilo.

```

<!-- CSS dependencies -->
  <link href="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-DrPVD9GufrxGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+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-TFMTyWuCbiptXTzvOgzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP" 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>

```

Isso cria uma página vazia com o contêiner do mapa.

Construindo o aplicativo: configuração

Para configurar seu aplicativo usando JavaScript:

1. Insira os identificadores do banco de identidades não autenticado que você criou em [Usando mapas – Etapa 2, Configurar a autenticação](#).

```
// 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. Configure AWS Amplify para usar os recursos que você criou, incluindo o banco de identidades e o recurso Mapa (mostrado aqui com o nome padrão de `explore.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,
    },
  },
});
```

## Construindo o aplicativo: inicialização do mapa

Para que o mapa seja exibido após o carregamento da página, você deve inicializar o mapa. Você pode ajustar a localização inicial do mapa, adicionar controles adicionais e sobrepor dados.

```
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

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `sources.grabmaptiles.attribution`. O Amplify fornecerá automaticamente a atribuição. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

## Executando o aplicativo

Você pode executar esse aplicativo de amostra usando-o em um servidor da web local ou abrindo-o em um navegador.

Para usar um servidor da web local, você pode usar o `npx`, instalado como parte do Node.js, ou qualquer outro servidor da web de sua escolha. Para usar `npx`, digite `npx serve` no mesmo diretório que `index.html`. Isso serve o aplicativo em `localhost:5000`.

**Note**

Se a política que você criou para sua função não autenticada do Amazon Cognito incluir uma condição `referer`, você poderá ser impedido de testar com URLs `localhost:.` Nesse caso, você pode testar com um servidor da web que forneça uma URL que esteja na sua política.

Depois de concluir o tutorial, o aplicativo final se parece com o exemplo a seguir.

```
<html>
  <head>
    <!-- CSS dependencies -->
    <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrxGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+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");
    }
  </script>

```

```
    }  
  
    initializeMap();  
  </script>  
</body>  
</html>
```

A execução deste aplicativo exibe um mapa em tela cheia usando o estilo de mapa escolhido. Esse exemplo também está descrito na guia [Incorporar mapa de qualquer página de recurso de mapa no console do Amazon Location Service](#).

Depois de concluir este tutorial, acesse o tópico [Exibir um mapa](#) na documentação AWS Amplify para saber mais, inclusive como exibir marcadores no mapa.

## Usando o Tangram com o Amazon Location Service

Esta seção fornece os seguintes tutoriais sobre como integrar o Tangram com o Amazon Location.

### Important

Os estilos do Tangram nos tutoriais a seguir são compatíveis somente com os recursos do mapa do Amazon Location configurados com o estilo `VectorHereContrast`.

Veja a seguir um exemplo de um comando da AWS CLI para criar um novo recurso de mapa chamado *TangramExampleMap* usando o *VectorHereContrast* estilo:

```
aws --region us-east-1 \  
  location \  
  create-map \  
  --map-name "TangramExampleMap" \  
  --configuration "Style=VectorHereContrast"
```

### Note

O faturamento é determinado pelo seu uso. Você pode incorrer em taxas pelo uso de outros serviços da AWS. Para obter mais informações, consulte os [preços do Amazon Location Service](#).

## Tópicos

- [Usando o Tangram com o Amazon Location Service](#)
- [Usando o Tangram ES para Android com o Amazon Location Service](#)
- [Usando o Tangram ES para iOS com o Amazon Location Service](#)

## Usando o Tangram com o Amazon Location Service

O [Tangram](#) é um mecanismo de mapeamento flexível, projetado para renderização em tempo real de mapas 2D e 3D a partir de blocos vetoriais. Ele pode ser usado com estilos projetados pelo MapZen e os blocos HERE fornecidos pela API de mapas do Amazon Location Service. Este guia descreve como integrar o Tangram com o Amazon Location em um JavaScript aplicativo HTML/básico, embora as mesmas bibliotecas e técnicas também se apliquem ao usar estruturas como React e Angular.

O Tangram é construído sobre o [Leaflet](#), uma JavaScript biblioteca de código aberto para mapas interativos compatíveis com dispositivos móveis. Isso significa que muitos plug-ins e controles compatíveis com o Leaflet também funcionam com o Tangram.

Os estilos do Tangram criados para funcionar com o [esquema Tilezen](#) são amplamente compatíveis com o Amazon Location ao usar mapas da HERE. Isso inclui:

- [Bubble Wrap](#) — Um estilo de orientação completo com ícones úteis para pontos de interesse
- [Cinnabar](#) — Um visual clássico e ideal para aplicações gerais de mapeamento
- [Refill](#) — Um estilo de mapa minimalista projetado para sobreposições de visualização de dados, inspirado no estilo seminal de Toner da Stamen Design
- [Tron](#) — Uma exploração das transformações de escala na linguagem visual de TRON
- [Walkabout](#) — Um estilo voltado para atividades ao ar livre que é perfeito para caminhar ou passear

[Este guia descreve como integrar o Tangram com o Amazon Location em um JavaScript aplicativo HTML/básico usando o estilo Tangram chamado Bubble Wrap.](#) Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

Embora outros estilos do Tangram sejam melhor acompanhados por blocos Raster, que codificam informações sobre o terreno, esse atributo ainda não é suportado pelo Amazon Location.

**⚠ Important**

Os estilos do Tangram no tutorial a seguir são compatíveis somente com os recursos de mapa do Amazon Location configurados com o estilo `VectorHereContrast`.

### Construindo o aplicativo: estrutura

O aplicativo é uma página HTML JavaScript para criar o mapa em seu aplicativo web. Crie uma página HTML (`index.html`) e crie o contêiner do mapa:

- Insira um elemento `div` com um `id` de mapa para aplicar as dimensões do mapa à visualização do mapa.
- As dimensões são herdadas da janela de exibição.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
</html>
```

### Construindo o aplicativo: adicionando dependências

Adicione a seguinte dependência:

- Folheto e seu CSS associado.
- Tangram.
- SDK da AWS para 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>
```

Isso cria uma página vazia com os pré-requisitos necessários. A próxima etapa orienta você na criação do JavaScript código para seu aplicativo.

## Construindo o aplicativo: configuração

Para configurar seu aplicativo com seus recursos e credenciais:

1. Insira os nomes e identificadores dos seus recursos.

```
// 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. Instancie um provedor de credenciais usando o banco de identidade não autenticado que você criou em [Usando mapas – Etapa 2, Configurar a autenticação](#). Como isso usa credenciais fora do fluxo de trabalho normal do SDK da AWS, as sessões expiram depois de uma hora.

```
// 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,
});
```

- Embora o Tangram permita que você substitua o(s) URL(s) usados para buscar blocos, ele não inclui a capacidade de interceptar solicitações para que elas possam ser assinadas.

Para contornar isso, substitua `sources.mapzen.url` para apontar para o Amazon Location usando um nome de host sintético `amazon.location`, que será gerenciado por um [Service Worker](#). Veja a seguir um exemplo de configuração de cena usando o [Bubble 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,
    },
  },
};
```

### Construindo o aplicativo: Solicitar transformação

Para registrar e inicializar o Service Worker, crie uma função `registerServiceWorker` a ser chamada antes que o mapa seja inicializado. Isso registra o JavaScript código fornecido em um arquivo separado chamado `sw.js` de controle `index.html` do service worker.

As credenciais são carregadas do Amazon Cognito e passadas para o Service Worker junto com a região para fornecer informações para assinar solicitações de blocos com o [Signature versão 4](#).

```
/**
 * 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");
  }
}
```

A implementação do Service Worker `sw.js` escuta os eventos do message para coletar as alterações nas credenciais e na configuração da região. Ele também atua como um servidor proxy, ouvindo eventos `fetch`. Eventos `fetch` direcionados ao nome de host sintético `amazon.location`

serão reescritos para atingir a API do Amazon Location apropriada e assinados usando o Signer do Amplify Core.

```
// 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));
});
```

Para renovar automaticamente as credenciais e enviá-las ao Service Worker antes que elas expirem, use a seguinte função com `index.html`:

```
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());  
}
```

## Construindo o aplicativo: Inicialização do mapa

Para que o mapa seja exibido após o carregamento da página, você deve inicializar o mapa. Você tem a opção de ajustar a localização inicial do mapa, adicionar controles adicionais e sobrepor dados.

### Note

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `source.grabmaptiles.attribution`.

Como o Tangram não solicita esses recursos e só é compatível com mapas da HERE, use “© 2020 HERE”. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

```
/**  
 * 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();
```

## Executando o aplicativo

Para executar esse exemplo, você pode:

- Usar um host que ofereça suporte a HTTPS,
- Use um servidor da web local para cumprir as restrições de segurança do Service Worker.

Para usar um servidor da web local, você pode usar o npx, porque ele é instalado como parte do Node.js. Você pode usar npx `serve` de dentro do mesmo diretório que `index.html` e `sw.js`. Isso serve ao aplicativo em [localhost:5000](http://localhost:5000).

A seguir está o arquivo `index.html`:

```
<!-- 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>
```

A seguir está o arquivo `sw.js`:

```
// 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));
  });
});
```

Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

### Usando o Tangram ES para Android com o Amazon Location Service


O [Tangram ES](#) é uma biblioteca C++ para renderizar mapas 2D e 3D a partir de dados vetoriais usando o OpenGL ES. É a contraparte nativa do [Tangram](#).

Os estilos do Tangram criados para funcionar com o [esquema Tilezen](#) são amplamente compatíveis com o Amazon Location ao usar mapas da HERE. Isso inclui:

- [Bubble Wrap](#) — Um estilo de orientação completo com ícones úteis para pontos de interesse.
- [Cinnabar](#) — Uma aparência clássica e ideal para aplicativos gerais de mapeamento.
- [Refill](#) — Um estilo de mapa minimalista projetado para sobreposições de visualização de dados, inspirado no estilo seminal de Toner da Stamen Design.
- [Tron](#) — Uma exploração das transformações de escala na linguagem visual de TRON.
- [Walkabout](#) — Um estilo voltado para atividades ao ar livre que é perfeito para caminhar ou sair de casa.

Este guia descreve como integrar o Tangram ES para Android com o Amazon Location usando o estilo Tangram chamado Cinnabar. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

Embora outros estilos do Tangram sejam melhor acompanhados por blocos raster, que codificam informações sobre o terreno, esse atributo ainda não é suportado pelo Amazon Location.

 Important

Os estilos do Tangram no tutorial a seguir são compatíveis somente com os recursos de mapa do Amazon Location configurados com o estilo `VectorHereContrast`.

## Construindo o aplicativo: inicialização

Para inicializar seu aplicativo:

1. Crie um novo projeto do Android Studio a partir do modelo Atividade vazia.
2. Certifique-se de que o Kotlin esteja selecionado para o idioma do projeto.
3. Selecione um SDK mínimo de API 16: Android 4.1 (Jelly Bean) ou mais recente.
4. Abra a Estrutura do projeto para selecionar Arquivo, Estrutura do projeto... e escolha a seção Dependências.
5. Com a opção <Todos os módulos> selecionada, escolha o botão + para adicionar uma nova Dependência de biblioteca.
6. Adicione o SDK para Android da AWS versão 2.19.1 ou posterior. Por exemplo:  
`com.amazonaws:aws-android-sdk-core:2.19.1`



7. Adicione o Tangram versão 0.13.0 ou posterior. Por exemplo:  
`com.mapzen.tangram:tangram:0.13.0`.

#### Note

Pesquisar por Tangram: `com.mapzen.tangram:tangram:0.13.0` gerará uma mensagem de que “não foi encontrado”, mas escolher OK permitirá que ele seja adicionado.

## Construindo o aplicativo: configuração

Para configurar seu aplicativo com seus recursos e sua região da AWS:

1. Criar `app/src/main/res/values/configuration.xml`.
2. Insira os nomes e identificadores dos seus recursos e também a região AWS em que foram criados:

```
<?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>
```

## Construindo o aplicativo: layout da atividade

Edite `app/src/main/res/layout/activity_main.xml`.

- Adicione um `MapView`, que renderiza o mapa. Isso também definirá o ponto central inicial do mapa.
- Adicione um `TextView`, que exibe a atribuição.

Isso também definirá o ponto central inicial do mapa.

**Note**

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `source.grabmaptiles.attribution`.

Como o Tangram não solicita esses recursos e só é compatível com mapas da HERE, use “© 2020 HERE”. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

```
<?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>
```

## Construindo o aplicativo: Solicitar transformação

Crie uma classe nomeada `SigV4Interceptor` para interceptar solicitações da AWS e assiná-las usando o [Signature versão 4](#). Isso será registrado no cliente HTTP usado para buscar recursos do mapa quando a Atividade Principal for criada.

```
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()
    }
}
```

## Construindo o aplicativo: atividade principal

A Atividade Principal é responsável por inicializar as visualizações que serão exibidas aos usuários. Isso envolve:

- Instanciando um `CredentialsProvider` do Amazon Cognito.
- Registrando o interceptador `Signature` versão 4.
- Configurando o mapa apontando-o para um estilo de mapa, substituindo URLs de blocos e exibindo a atribuição apropriada.

O `MainActivity` também é responsável por encaminhar os eventos do ciclo de vida para a visualização do mapa.

```
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 getHttpHandler(): HttpHandler {
    val builder = DefaultHttpHandler.getClientBuilder()

    val credentialsProvider = CognitoCachingCredentialsProvider(
        applicationContext,
        getString(R.string.identityPoolId),
        Regions.US_EAST_1
    )

    return DefaultHttpHandler(
        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()  
}  
}
```

A execução deste aplicativo exibe um mapa em tela cheia no estilo de sua escolha. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

## Usando o Tangram ES para iOS com o Amazon Location Service

O [Tangram ES](#) é uma biblioteca C++ para renderizar mapas 2D e 3D a partir de dados vetoriais usando o OpenGL ES. É a contraparte nativa do [Tangram](#).

Os estilos do Tangram criados para funcionar com o [esquema Tilezen](#) são amplamente compatíveis com o Amazon Location ao usar mapas da HERE. Isso inclui:

- [Bubble Wrap](#) — Um estilo de orientação completo com ícones úteis para pontos de interesse
- [Cinnabar](#) — Um visual clássico e ideal para aplicações gerais de mapeamento
- [Refill](#) — Um estilo de mapa minimalista projetado para sobreposições de visualização de dados, inspirado no estilo seminal de Toner da Stamen Design
- [Tron](#) — Uma exploração das transformações de escala na linguagem visual de TRON
- [Walkabout](#) — Um estilo voltado para atividades ao ar livre que é perfeito para caminhar ou passear

Este guia descreve como integrar o Tangram ES para iOS com o Amazon Location usando o estilo Tangram chamado Cinnabar. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

Embora outros estilos do Tangram sejam melhor acompanhados por blocos raster, que codificam informações sobre o terreno, esse atributo ainda não é suportado pelo Amazon Location.

### Important

Os estilos do Tangram no tutorial a seguir são compatíveis somente com os recursos de mapa do Amazon Location configurados com o estilo `VectorHereContrast`.



## Construindo o aplicativo: inicialização

Como inicializar o aplicativo:

1. Crie um novo projeto Xcode a partir do modelo Aplicativo.
2. Selecione SwiftUI para sua interface.
3. Selecione o aplicativo SwiftUI para seu Ciclo de vida.
4. Selecione Swift como seu idioma.

## Construindo o aplicativo: adicionar dependências

Para adicionar dependências, você pode usar um gerenciador de dependências, como: [CocoaPods](#)

1. No seu terminal, instale CocoaPods:

```
sudo gem install cocoapods
```

2. Navegue até o diretório do projeto do seu aplicativo e inicialize o Podfile com o gerenciador de CocoaPods pacotes:

```
pod init
```

3. Abra o Podfile para adicionar AWSCore e Tangram-es como dependências:

```
platform :ios, '12.0'  
  
target 'Amazon Location Service Demo' do  
  use_frameworks!  
  
  pod 'AWSCore'  
  pod 'Tangram-es'  
end
```

4. Baixe e instale dependências:

```
pod install --repo-update
```

5. Abra a área de trabalho do Xcode que CocoaPods criou:

```
xed .
```

## Construindo o aplicativo: configuração

Adicione as seguintes chaves e valores ao Info.plist para configurar o aplicativo e desativar a telemetria:

Chave	Valor
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd
MapName	ExampleMap
SceneURL	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

## Construindo o aplicativo: ContentView layout

Para renderizar o mapa, edite ContentView.swift:

- Adicione um MapView que renderiza o mapa.
- Adicione um TextField que exiba a atribuição.

Isso também define o ponto central inicial do mapa.

### Note

Você deve fornecer uma marca nominativa ou atribuição de texto para cada provedor de dados que você usa, seja em seu aplicativo ou em sua documentação. As strings de atribuição estão incluídas na resposta do descritor de estilo sob as teclas `sources.esri.attribution`, `sources.here.attribution` e `source.grabmaptiles.attribution`. Ao usar os recursos do Amazon Location com [provedores de dados](#), certifique-se de ler os [termos e condições do serviço](#).

```
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()
    }
}
```

## Construindo o aplicativo: Solicitar transformação

Crie um novo arquivo Swift chamado `AWSSignatureV4URLHandler.swift` contendo a seguinte definição de classe para interceptar solicitações AWS e assiná-las usando o [Signature versão 4](#). Isso será registrado como um manipulador de URL dentro do `MapView` do Tangram.

```
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)
}
}
```

## Construindo o aplicativo: visualização do mapa

A visualização do mapa é responsável por inicializar uma instância de `AWSSignatureV4Delegate` e configurar o `MGLMapView` subjacente, que busca recursos e renderiza o mapa. Ele também lida com a propagação de cadeias de atribuição da fonte do descritor de estilo de volta para o `ContentView`.

Crie um novo arquivo Swift chamado `MapView.swift` contendo a seguinte definição de `struct`:

```
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) {
}
}
```

A execução deste aplicativo exibe um mapa em tela cheia no estilo de sua escolha. Essa amostra está disponível como parte do repositório de amostras do Amazon Location Service em [GitHub](#).

## Desenhando recursos de dados em um mapa

Depois de ter um aplicativo que renderiza um mapa, usando o Amplify MapLibre ou o Tangram para renderizar o mapa, a próxima etapa natural é desenhar feições na parte superior do mapa. Por exemplo, convém renderizar a localização de seus clientes como marcadores no mapa.

Em geral, você pode usar [as funções de pesquisa do Places](#) para encontrar localizações a partir dos seus dados e, em seguida, MapLibre usar a funcionalidade do Amplify ou do Tangram para renderizar as localizações.

Para ver exemplos de renderização de diferentes tipos de objetos no mapa, veja os seguintes MapLibre exemplos:

- [Exemplo: desenhe marcadores](#)
- [Exemplo: desenhe pontos agrupados](#)
- [Exemplo: desenhe um polígono](#)

Para obter mais exemplos e tutoriais, consulte [Exemplos de código e tutoriais para trabalhar com o Amazon Location Service](#).

## Definindo extensões para um mapa usando MapLibre

Há momentos em que você não quer que seus usuários consigam deslocar ou ampliar o mundo inteiro. Se você estiver usando o controle MapLibre de mapa, poderá limitar as extensões ou limites do controle de mapa com a `maxBounds` opção e restringir o zoom com `minZoom` e opções. `maxZoom`

O exemplo de código a seguir mostra como inicializar o controle do mapa para restringir a panorâmica a um limite específico (nesse caso, as extensões da fonte de dados Grab).

### Note

Esses exemplos estão JavaScript disponíveis e funcionam dentro do contexto do [Criar um aplicativo Web](#) tutorial.

```
// 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,
  }
);
```

Da mesma forma, você pode definir um nível mínimo e máximo de zoom para o mapa. Os valores para ambos podem estar entre 0 e 24, embora os padrões sejam 0 para o zoom mínimo e 22 para o máximo (os provedores de dados podem não fornecer dados em todos os níveis de zoom. A maioria das bibliotecas de mapas lida com isso automaticamente). O exemplo a seguir inicializa as `maxZoom` opções `minZoom` e no controle MapLibre Mapa.

```
// Set the minimum and maximum zoom levels
var mglMap = new maplibregl.Map(
  {
    container: 'map',
    style: mapName,
    maxZoom: 12,
    minZoom: 5,
    transformRequest,
  }
);
```

#### Tip

O controle MapLibre Map também permite definir essas opções em tempo de execução, em vez de durante a inicialização, com `set . . .` funções `get . . .` e. Por exemplo, use `getMaxBounds` e `setMaxBounds` para alterar os limites do mapa em runtime.

## Gerenciando seus recursos de mapa

Você pode gerenciar seus recursos de mapa usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Listar recursos do mapa

Você pode visualizar uma lista dos recursos do seu mapa usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

#### Console

Para visualizar uma lista de recursos de mapas existentes usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Mapas no painel de navegação à esquerda.



3. Veja uma lista dos recursos do seu mapa em Meus mapas.

## API

Use a operação [ListMaps](#) das APIs de mapas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter uma lista dos recursos do mapa na conta AWS.

```
POST /maps/v0/list-maps
```

Este é um exemplo de resposta para [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

Use o comando [list-map](#).

O exemplo a seguir é um AWS CLI para obter uma lista dos recursos do mapa na conta AWS.

```
aws location list-maps
```

## Obtenha detalhes dos recursos do mapa

Você pode obter detalhes sobre qualquer recurso de mapa na sua conta da AWS usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

## Console

Para visualizar os detalhes de um recurso de mapa usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Mapas no painel de navegação à esquerda.
3. Em Meus mapas, selecione o link do nome do recurso do mapa de destino.

## API

Use a operação [DescribeMap](#) das APIs de mapas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter os detalhes dos recursos do mapa *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

Este é um exemplo de resposta para [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

Use o comando [describe-map](#).

O exemplo a seguir é AWS CLI para obter os detalhes dos recursos do mapa *ExampleMap*.

```
aws location describe-map \
```

```
--map-name "ExampleMap"
```

## Excluir um recurso de mapa

Você pode excluir um recurso de mapa da sua conta AWS usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Warning

Essa operação exclui o recurso permanentemente.

## Console

Para excluir um recurso de mapa existente usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Mapas no painel de navegação à esquerda.
3. Na lista Meus mapas, selecione o mapa de destino.
4. Escolha Excluir mapa.

## API

Use a operação [DeleteMap](#) das APIs de mapas do Amazon Location.

O exemplo a seguir é uma solicitação de API para excluir o recurso do mapa *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

Veja a seguir um exemplo de uma resposta bem-sucedida do [DeleteMap](#):

```
HTTP/1.1 200
```

## CLI

Use o comando [delete-map](#).

O exemplo a seguir é um AWS CLI comando para excluir o recurso de mapa *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

## Pesquisando dados de localização e geolocalização usando Amazon Location

O Amazon Location inclui a capacidade de pesquisar os dados de geolocalização, ou local, do provedor escolhido. Há vários tipos de pesquisa disponíveis.

- **Geocodificação** — A geocodificação é o processo de pesquisar endereços, regiões, nomes comerciais ou outros pontos de interesse, com base na entrada de texto. Ele retorna detalhes e a localização (em latitude e longitude) dos resultados encontrados.
- **Geocodificação reversa** — A geocodificação reversa permite que você encontre lugares próximos a um determinado local.
- **Preenchimento automático** — O preenchimento automático é o processo de fazer sugestões automáticas à medida que o usuário digita uma consulta. Por exemplo, se eles digitarem **Par**, uma sugestão pode ser **Paris, France**.

O Amazon Location permite que você escolha um provedor de dados para operações de busca de lugares criando e configurando um recurso de índice de localização.

Depois de criar seu recurso, você pode enviar solicitações usando o AWS SDK do seu idioma preferido, o Amplify, ou os endpoints da API REST. Você pode usar os dados da resposta para marcar localizações em um mapa, enriquecer os dados de posição e converter posições em texto legível por humanos.

### Note

Para obter uma visão geral dos conceitos de local de pesquisa, consulte [Pesquisa de lugares](#).

### Tópicos

- [Pré-requisitos](#)
- [Geocodificação](#)

- [Geocodificação reversa](#)
- [Autocompletar](#)
- [Usando IDs de locais](#)
- [Categorias de lugares e resultados de filtragem](#)
- [Amazon Aurora PostgreSQL funções definidas pelo usuário para o Amazon Location Service](#)
- [Gerenciar seus recursos do índice de locais](#)

## Pré-requisitos

Antes de começar a geocodificar, reverter a geocodificação ou pesquisar lugares, siga as etapas de pré-requisito:

### Tópicos

- [Criação de um recurso de índice de localização](#)
- [Autenticando suas solicitações](#)

## Criação de um recurso de índice de localização

Comece criando um recurso de índice de lugares em sua AWS conta.

Ao criar um recurso de índice de localização você pode escolher entre os provedores de dados disponíveis para suportar consultas de geocodificação, geocodificação reversa e pesquisas:


1. Esri — para obter mais informações sobre a abrangência da Esri em sua região de interesse, consulte a abrangência de [geocodificação da Esri](#) na documentação da Esri.
2. HERE Technologies — para obter mais informações sobre a abrangência da HERE em sua região de interesse, consulte a [abrangência de geocodificação da HERE](#) na documentação da HERE.
3. Grab — o Grab fornece dados somente do Sudeste Asiático. Para obter mais informações sobre este perfil, consulte [Países/regiões e área de cobertura](#) neste guia.

Você pode fazer isso usando o console do Amazon Location Service AWS CLI, o ou as APIs Amazon Location.

### Console

Para criar um recurso de índice de localização usando o console do Amazon Location Service

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação, escolha Índices de localização.
3. Escolha Criar índice de localização.
4. Preencha as seguintes caixas:
  - Nome – insira um nome para o recurso de índice de localização. Por exemplo, *ExamplePlaceIndex*. Máximo de 100 caracteres. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados.
  - Descrição – insira uma descrição opcional.
5. Em Provedores de dados, escolha um [provedor de dados](#) disponível para usar com seu recurso de índice de localização.

 Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço daAWS](#) para obter mais detalhes.

6. Em Opções de armazenamento de dados, especifique se você pretende armazenar os resultados da pesquisa do seu recurso de índice de localização.
7. (Opcional) Em Tags, insira uma tag Chave e Valor. Isso adiciona uma tag ao seu novo recurso de índice de localização. Para obter mais informações, consulte [Marcar seus recursos](#).
8. Escolha Criar índice de localização.

## API

Para criar um recurso de índice de localização usando as APIs do Amazon Location

Use a operação [CreatePlaceIndex](#) das APIs de localização do Amazon Location.

O exemplo a seguir é uma solicitação de API para criar um recurso de índice de local chamado *ExamplePlaceIndex* usando o provedor de dados *Esri*.

```
POST /places/v0/indexes
```

```
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

## AWS CLI

Para criar um recurso de índice de localização usando comandos AWS CLI

Use o comando [create-place-index](#).

O exemplo a seguir cria um recurso de índice de local chamado *ExamplePlaceIndex* usando a *Esri* como provedor de dados.

```
aws location \
  create-place-index \
  --data-source "Esri" \
  --description "Example place index" \
  --index-name "ExamplePlaceIndex" \
  --tags Tag1=Value1
```

### Note

O faturamento depende do seu uso. Você pode incorrer em taxas pelo uso de outros serviços da AWS. Para obter mais informações, consulte os [preços do Amazon Location Service](#).

## Autenticando suas solicitações

Depois de criar um recurso de índice de localização e estiver pronto para começar a criar recursos de localização em seu aplicativo, escolha como você autenticaria suas solicitações:

- Para explorar as formas para acessar os serviços, consulte [Acessando o Amazon Location Service](#).
- Se você tem um site com usuários anônimos, talvez queira usar chaves de API ou o Amazon Cognito.

## Exemplo

O exemplo a seguir mostra o uso de uma chave de API para autorização, o [AWS JavaScript SDK v3](#) e o Amazon Location. [JavaScript Auxiliar de autenticação](#)

```
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);
```

## Geocodificação

A geocodificação é um processo que converte texto, como endereço, região, nome comercial ou ponto de interesse, em um conjunto de coordenadas geográficas. Você pode usar recursos de



índice de localização para enviar solicitações de geocodificação e incorporar dados recuperados da geocodificação para exibir dados em um mapa para seu aplicativo da web ou móvel.

Esta seção explica como enviar uma solicitação simples de geocodificação e como enviar solicitações de geocodificação com especificações opcionais.

## Geocodificação

Você pode enviar uma solicitação simples para geocodificar usando a operação [SearchPlaceIndexForText](#) para converter um endereço em um conjunto de coordenadas. Uma solicitação simples contém o seguinte parâmetro necessário:

- **Text** — Um endereço, nome, cidade ou região para converter em um conjunto de coordenadas. Por exemplo, a string `Any Town`.

Para especificar um número máximo de resultados por página, use o seguinte parâmetro opcional:

- **MaxResults** — Limita o número máximo de resultados retornados na resposta da consulta.

Você pode usar as APIs de localização da Amazon AWS CLI ou as APIs.

### API

O exemplo a seguir é uma [SearchPlaceIndexForText](#) solicitação para pesquisar no recurso de índice de local *ExamplePlaceIndex*, um endereço, nome, cidade ou região chamado *Qualquer cidade*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "MaxResults": 10
}
```

### AWS CLI

O exemplo a seguir é um [search-place-index-for-text](#) comando para pesquisar no recurso de índice de local *ExamplePlaceIndex*, um endereço, nome, cidade ou região chamado *Qualquer cidade*.

```
aws location \  
  search-place-index-for-text \  
    --index-name ExamplePlaceIndex \  
    --text "Any Town" \  
    --max-results 10
```

## Codifique geograficamente perto de uma posição

Ao geocodificar, você pode geocodificar perto de uma determinada posição com o seguinte parâmetro opcional:

- **BiasPosition** — A posição que você deseja pesquisar nas proximidades. Isso restringe sua pesquisa pesquisando os resultados mais próximos da posição especificada. Definido como [longitude, latitude]

O exemplo a seguir é uma solicitação [SearchPlaceIndexForText](#) para pesquisar no recurso de índice de localização um endereço, nome, cidade ou região chamada *Qualquer cidade* próxima à posição [-123, 4567, 45, 6789].

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text  
Content-type: application/json  
  
{  
  "Text": "Any Town",  
  "BiasPosition": [-123.4567, 45.6789]  
}
```

## Codifique geograficamente dentro de uma caixa delimitadora

Você pode geocodificar dentro de uma caixa delimitadora para restringir seus resultados às coordenadas dentro de um determinado limite usando o seguinte parâmetro opcional:

- **FilterBoundingBox** — Uma caixa delimitadora que você especifica para filtrar seus resultados para coordenadas dentro dos limites da caixa. Definido como [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

**Note**

Uma solicitação não pode conter os parâmetros `FilterBoundingBox` e `BiasPosition`. A especificação dos dois parâmetros na solicitação retorna um erro `ValidationException`.

O exemplo a seguir é uma solicitação [SearchPlaceIndexForText](#) para pesquisar em uma caixa delimitadora um endereço, nome, cidade ou região chamada *Qualquer cidade*. A caixa delimitadora segue o seguinte:

- A longitude do canto sudoeste é *-124,1450*.
- A latitude do canto sudoeste é *41,7045*.
- A longitude do canto nordeste é *-124,1387*.
- A latitude do canto nordeste é *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
  ]
}
```

## Codificar geograficamente dentro de um país

Você pode geocodificar dentro de um ou mais países que especificar usando o seguinte parâmetro opcional:

- `FilterCountries` — O país ou região em que você deseja geocodificar. Você pode definir até 100 países em uma solicitação usando um código de país de três letras [ISO 3166](#). Por exemplo, use AUS para a Austrália.

O exemplo a seguir é uma solicitação [SearchPlaceIndexForText](#) para pesquisar um endereço, nome, cidade ou região chamada *Qualquer cidade* na Alemanha e na França.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "FilterCountries": ["DEU", "FRA"]
}
```

## Filtrar por categoria

Você pode filtrar as categorias que são retornadas em sua solicitação de geocodificação usando o seguinte parâmetro opcional:

- `FilterCategories` — As categorias de resultados que você deseja retornar em sua consulta. É possível especificar até cinco categorias em uma solicitação. Você pode encontrar a lista das categorias do Amazon Location Service na seção [Categorias](#). Por exemplo, é possível especificar `Hotel` somente para os hotéis resultantes da sua consulta.

O exemplo a seguir é uma solicitação [SearchPlaceIndexForText](#) para pesquisar uma cafeteria chamada *Hometown Coffee* nos Estados Unidos.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Hometown Coffee",
  "FilterCategories": ["Coffee Shop"],
  "FilterCountries": ["USA"]
}
```

Para obter mais detalhes sobre a filtragem em categorias, consulte [Categorias de lugares e resultados de filtragem](#)

## Geocodifique em um idioma preferido

Você pode definir uma preferência de idioma para os resultados da sua pesquisa usando o parâmetro `Language` opcional. Por exemplo, uma pesquisa por **100 Main St, Anytown, USA**

pode retornar 100 Main St, Any Town, USA por padrão. Mas se você selecionar `fr` como Language, os resultados poderão retornar 100 Rue Principale, Any Town, États-Unis em vez disso.

- Language — Um código de linguagem a ser usado para renderizar os resultados da sua consulta. O valor deve ser um código de idioma [BCP 47](#) válido. Por exemplo, em para inglês.

#### Note

Se Language não for especificado ou se o idioma especificado não for suportado para um resultado, o idioma padrão do parceiro para esse resultado será usado.

O exemplo a seguir é uma solicitação `SearchPlaceIndexForText` para pesquisar um local chamado **Any Town** com o idioma preferencial especificado como `de`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
{
  "Text": "Any Town",
  "Language": "de"
}
```

## Exemplo de resposta

### Example

Veja a seguir um exemplo de resposta quando você chama a operação [SearchPlaceIndexForText](#) a partir das APIs de localização do Amazon Location. Os resultados incluem [locais](#) relevantes e o [resumo](#) da solicitação. Duas respostas são mostradas, com base na seleção da Esri ou da HERE como parceira.

### 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": "AQAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJJZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
    "Distance":
    523.4619749905755
}
]
```

## Geocodificação reversa

A geocodificação reversa é um processo que converte um conjunto de coordenadas em texto significativo, como endereço, região, nome comercial ou ponto de interesse. Você pode usar recursos de índice de localização para enviar solicitações de geocodificação reversa e incorporar dados recuperados da geocodificação reversa para exibir dados em um mapa para seu aplicativo da web ou móvel.

Esta seção explica como enviar uma solicitação simples de geocodificação reversa.

### Geocodificação reversa

Você pode enviar uma solicitação simples para fazer geocodificação reversa de um conjunto de coordenadas e convertê-las em um endereço significativo, um ponto de interesse ou um local geral sem que um endereço use a operação [SearchPlaceIndexForPosition](#). Uma solicitação simples contém o seguinte parâmetro necessário:

- **Position** — Um conjunto de coordenadas que você deseja converter em um endereço, ponto de interesse ou localização geral. Definido usando o formato `[longitude, latitude]`.

Para especificar um número máximo de resultados por página, adicione o seguinte parâmetro opcional:

- **MaxResults** — Limita o número máximo de resultados retornados na resposta da consulta.

Se você quiser especificar um idioma preferencial para os resultados da sua consulta, use o seguinte parâmetro opcional:



- Language — Um código de linguagem a ser usado para renderizar resultados. O valor deve ser um código de idioma [BCP 47](#) válido. Por exemplo, en para inglês.

### Note

Se Language não for especificado ou se o idioma especificado não for suportado para um resultado, o idioma padrão do parceiro para esse resultado será usado.

Você pode usar as APIs de localização da Amazon AWS CLI ou as APIs.

## API

*O exemplo a seguir é uma [SearchPlaceIndexForPosition](#) solicitação para pesquisar no recurso de índice de local um endereço significativo `ExamplePlaceIndex`, ponto de interesse ou localização geral próxima à posição `[122.3394, 47.6159]`.*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 5,
  "Language": "de"
}
```

## AWS CLI

*O exemplo a seguir é um [search-place-index-for-position](#) comando para pesquisar no recurso de índice de local um endereço significativo `ExamplePlaceIndex`, ponto de interesse ou localização geral perto da posição `[122.3394, 47.6159]`.*

```
aws location \
  search-place-index-for-position \
    --index-name ExamplePlaceIndex \
    --position -122.3394 47.6159 \
    --max-results 5 \
    --language de
```

## Exemplo de resposta

### Example

Veja a seguir um exemplo de resposta ao chamar a operação [SearchPlaceIndexForPosition](#) a partir das APIs de localização do Amazon Location. Os resultados retornam [locais](#) relevantes e o [resumo](#) da solicitação. Duas respostas são mostradas, com base na seleção da Esri ou da Here como parceira.

### 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": "AQAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
    "Distance": 1.868586125090601
  }
]
}
```

## Autocompletar

O preenchimento automático fornece feedback responsivo aos usuários finais enquanto eles digitam sua consulta de pesquisa. Oferece sugestões de endereços e pontos de interesse com base em texto em formato livre parcial ou com erros de ortografia. Você pode usar recursos de índice de localização para solicitar sugestões de preenchimento automático e exibir as sugestões resultantes em seu aplicativo.

O Amazon Location não suporta o armazenamento de sugestões de preenchimento automático. Um erro será retornado se o índice de localização usado para uma chamada de preenchimento automático estiver configurado para uso com códigos geográficos armazenados. Para usar códigos geográficos armazenados e consultar sugestões, crie e configure vários índices de localização.

Esta seção descreve como enviar uma solicitação de preenchimento automático. Ela começa com a forma mais básica da solicitação e, em seguida, mostra parâmetros opcionais que você pode usar para aumentar a relevância dos resultados de pesquisa de preenchimento automático.

### Usando o preenchimento automático

Você pode enviar uma solicitação simples de sugestões de preenchimento automático usando a operação [SearchPlaceIndexForSuggestions](#). A forma mais simples da solicitação tem um único parâmetro obrigatório, a consulta `Text`:

- `Text` — O texto parcial de formato livre a ser usado para gerar sugestões de locais. Por exemplo, a string `eiffel tow`.

Para limitar o número de resultados encontrados, adicione o parâmetro opcional `MaxResults`:

- `MaxResults` — limita o número de resultados encontrados na resposta da consulta.

Você pode usar as APIs do Amazon Location ou a AWS CLI.

## API

O exemplo a seguir é uma [SearchPlaceIndexForSuggestions](#) solicitação para pesquisar no recurso de índice de lugares *ExamplePlaceIndex*,, até 5 sugestões com base no nome parcial do local *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

## AWS CLI

O exemplo a seguir é um [search-place-index-for-suggestions](#) comando para pesquisar no recurso de índice de local *ExamplePlaceIndex*,, até 5 sugestões com base no nome parcial do local *kamp*.

```
aws location \
  search-place-index-for-suggestions \
  --index-name ExamplePlaceIndex \
  --text kamp \
  --max-results 5
```

A chamada para `SearchPlaceIndexForSuggestions` resulta em uma lista de locais com um nome e um ID para cada um. Você pode usar esses resultados para apresentar sugestões sobre o que o usuário pode estar procurando enquanto digita, como fornecer uma lista suspensa de opções abaixo de uma caixa de texto. Por exemplo, aqui estão os resultados das sugestões, com base em um usuário digitando *kamp*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
```

```

    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymyYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
      "PlaceId":
"AQAAAIAAzZfZt3qMrUkG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
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
"
    }
  ]
}

```

A próxima seção explica como usar a `PlaceID` partir desses resultados.

## Usando os resultados do preenchimento automático

A chamada para `SearchPlaceIndexForSuggestions` resulta em uma lista de locais com um nome e um ID para cada um. Você pode usar esses resultados para apresentar sugestões sobre o que o usuário pode estar procurando enquanto digita, como fornecer uma lista suspensa de opções

abaixo de uma caixa de texto. Quando o usuário escolhe um dos resultados, você pode chamar a [GetPlace](#) operação com o ID de sua seleção para retornar os detalhes desse local, incluindo localização, endereço ou outros detalhes.

### Note

O PlaceId é válido somente se todos os itens a seguir forem iguais na solicitação de pesquisa original e na chamada para GetPlace.

- Cliente Conta da AWS
- Região da AWS
- Provedor de dados especificado no recurso de índice de localização

Normalmente, você usa GetPlace com as APIs do Amazon Location. O exemplo a seguir é uma solicitação [GetPlace](#) para encontrar uma das sugestões da seção anterior. Este exemplo é baseado no nome parcial do local *kamp*.

```
POST /places/v0/indexes/ExamplePlaceIndex/  
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKLGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3-  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

## Preenchimento automático perto de uma posição

Ao pesquisar sugestões de lugares com preenchimento automático usando [SearchPlaceIndexForSuggestions](#), você pode obter mais sugestões relevantes localmente adicionando o seguinte parâmetro opcional:

- **BiasPosition** — A posição que você deseja pesquisar nas proximidades. Definido como [longitude, latitude].

*O exemplo a seguir usa uma [SearchPlaceIndexForSuggestions](#) solicitação para pesquisar no recurso ExamplePlaceIndex de índice de lugares sugestões de lugares que correspondam à consulta parcial kamp perto da posição [32.5827, 0.3169].*

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
```

```
Content-type: application/json

{
  "Text": "kamp",
  "BiasPosition": [32.5827,0.3169]
}
```

*As sugestões resultantes para o mesmo Text podem ser diferentes se um BiasPosition diferente for escolhido, como [-96.7977, 32.7776].*

## Preenchimento automático dentro de uma caixa delimitadora

Você pode restringir sua pesquisa de preenchimento automático para receber somente sugestões de lugares localizados dentro de um determinado limite adicionando o seguinte parâmetro opcional:

- **FilterBoundingBox** — Uma caixa delimitadora que você especifica para filtrar seus resultados para coordenadas dentro dos limites da caixa. Definido como [LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]

### Note

Uma solicitação não pode conter os parâmetros **FilterBoundingBox** e **BiasPosition**. A especificação dos dois parâmetros na solicitação retorna um erro **ValidationException**.

O exemplo a seguir usa uma [SearchPlaceIndexForSuggestions](#) solicitação para pesquisar no recurso *ExamplePlaceIndex* de índice de lugares sugestões de lugares que correspondam à consulta parcial *kamp* e que estejam contidas na caixa delimitadora onde:

- A longitude do canto sudoeste da caixa delimitadora é *32,5020*.
- A latitude do canto sudoeste da caixa delimitadora é *0,2678*.
- A longitude do canto nordeste da caixa delimitadora é *32,6129*.
- A latitude do canto nordeste da caixa delimitadora é *0,3502*.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json
```



```
{
  "Text": "kamp",
  "FilterBBox": [
    32.5020, 0.2678,
    32.6129, 0.3502
  ]
}
```

As sugestões retornadas para o mesmo Text são diferentes se uma FilterBBox diferente for escolhida, como [-97.9651, 32.0640, -95.1196, 34.0436].

## Preenchimento automático dentro de um país

Você pode restringir sua pesquisa de preenchimento automático para receber apenas sugestões de lugares localizados em um determinado país ou conjunto de países, adicionando o seguinte parâmetro opcional:

- **FilterCountries** — Os países em que você deseja pesquisar sugestões de lugares. Você pode especificar até 100 países em uma solicitação usando um código de país de três letras [ISO 3166](#). Por exemplo, use AUS para a Austrália.

O exemplo a seguir usa uma [SearchPlaceIndexForSuggestions](#) solicitação para pesquisar no recurso *ExamplePlaceIndex* de índice de lugares sugestões de lugares que correspondam à consulta parcial *kamp* e que estejam contidas em Uganda, Quênia ou Tanzânia:

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}
```

As sugestões retornadas para o mesmo Text são diferentes se uma lista FilterCountries diferente for escolhida, como ["EUA"].

## Exemplo de resposta

Veja a seguir um exemplo de resposta de preenchimentos automáticos sugeridos para a operação [SearchPlaceIndexForSuggestions](#), usando o texto *kamp*.

```

{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIAADsn2T3KdrWearXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
      "PlaceId":
"AQAAAIAAzZfZt3qMruKG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADAj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZ1dvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yY0LhZx4k
",
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
"AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
"
    }
  ]
}

```

## Usando IDs de locais

A pesquisa por lugares retorna uma lista de resultados. A maioria dos resultados inclui um `PlaceId` para esse resultado. Você pode usar o `PlaceId` em uma operação [GetPlace](#) para retornar as informações sobre esse local (incluindo nome, endereço, localização ou outros detalhes).

### Note

O uso [SearchPlaceIndexForSuggestions](#) retornará `PlaceId` resultados para índices de qualquer lugar criados com qualquer fonte de dados. Usando [SearchPlaceIndexForText](#) ou [SearchPlaceIndexForPosition](#) retornará um `PlaceId` somente se a fonte de dados usada for AQUI.

Cada `PlaceId` define de forma exclusiva o local ao qual se refere, mas um único local pode ter mais de um `PlaceId` ao longo do tempo e com base no contexto. As regras a seguir descrevem a exclusividade e a longevidade de um `PlaceId`.

- As chamadas `PlaceId` retornadas que você faz são específicas para você Conta da AWS, para a AWS região e para o provedor de dados em seu `PlaceIndex` recurso. `GetPlace` encontrará resultados somente quando esses três atributos corresponderem à chamada original que criou `PlaceId` o.
- O `PlaceId` para um local mudará quando os dados sobre esse local mudarem. Por exemplo, quando a empresa a que se refere muda de local ou muda de nome.
- O resultado `PlaceId` de uma chamada de pesquisa repetida pode mudar quando o serviço de back-end faz uma atualização. Os mais antigos `PlaceId` continuarão sendo encontrados, mas novas chamadas para pesquisar poderão resultar em uma ID diferente.

O `PlaceId` é uma string. Não há limite específico para o comprimento de um `PlaceId`. Veja a seguir um exemplo de um `PlaceId` válido.

```
AQAAAIAADsn2T3KdIrWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Chamar `GetPlace` com um `PlaceId` para um local cujos dados foram alterados (por exemplo, um local comercial que fechou) resultará em um erro 404, `ResourceNotFound`. Chamar `GetPlace`

com um PlaceId que não é válido ou fora do contexto, como de outro Conta da AWS, retornará um `400 ValidationException` erro.

Embora você possa usar PlaceID em solicitações subsequentes, PlaceID não pretende ser um identificador permanente e o ID pode mudar entre chamadas consecutivas de API. Veja o seguinte comportamento de placeID para cada provedor de dados:

- Esri: Os IDs de local serão alterados a cada trimestre, no mínimo. O período típico para essas mudanças seria março, junho, setembro e dezembro. Os IDs de local também podem mudar entre as alterações trimestrais típicas, mas isso será muito menos frequente.
- AQUI: Recomendamos que você armazene os dados em cache por no máximo uma semana para mantê-los atualizados. Você pode presumir que menos de 1% de mudanças de ID serão liberadas em relação ao lançamento, o que ocorre aproximadamente 1 a 2 vezes por semana.
- Grab: Os IDs de lugar podem expirar ou se tornar inválidos nas seguintes situações.
  - Operações de dados: O POI pode ser removido do banco de dados do Grab POI pelo Grab Map Ops com base na verdade fundamental, como estar fechado no mundo real, ser detectado como um POI duplicado ou ter informações incorretas. O Grab sincronizará os dados com o ambiente do Waypoint semanalmente.
  - POI interpolado: o POI interpolado é um POI temporário gerado em tempo real ao atender a uma solicitação e será marcado como derivado no campo da resposta. `place.result_type` As informações dos POIs interpolados serão retidas por pelo menos 30 dias, o que significa que, em 30 dias, você poderá obter detalhes do POI por ID do Place na API Place Details. Após 30 dias, os POIs interpolados (tanto o ID do local quanto os detalhes) podem expirar e ficar inacessíveis a partir da API Places Details.

## Categorias de lugares e resultados de filtragem

Os lugares são categorizados. Se você pesquisar uma empresa, a empresa pode ser um `Restaurant`, por exemplo. Até mesmo os resultados de uma busca por um endereço podem ser categorizados de acordo com o endereço, rua ou cruzamento.

Em termos gerais, o Amazon Location Service categoriza os locais em Tipos de locais. Os pontos de interesse são ainda categorizados em Tipos de pontos de interesse.

### Note

Nem todos os resultados terão categorias.

Você pode usar as categorias para filtrar suas pesquisas de geocodificação.

## Filtrar resultados

Ao usar `SearchPlaceIndexForText`, você pode filtrar os resultados que são resultantes das categorias que você deseja usar. Por exemplo: .

- Se você quiser pesquisar um lugar chamado “Hometown Coffee” e os resultados categorizados forem apenas cafeterias, você pode fazer isso chamando `SearchPlaceIndexForText` e incluindo a Categoria do ponto de interesse, `Coffee Shop` no parâmetro `FilterCategories`.
- Ao pesquisar “123 Main St, Anytown, WA, 98123, USA”, você pode filtrar o resultado apenas para endereços, para não obter correspondências, por exemplo, o código postal. Filtre apenas para endereços incluindo o Tipo de local, `AddressType` no parâmetro `FilterCategories`.

### Note

Nem todos os provedores de dados oferecem suporte à filtragem ou da mesma forma. Para ter mais informações, consulte [Limitações de filtragem por provedor de dados](#).

A próxima seção lista as categorias que você pode filtrar.

## Categorias

As listas a seguir mostram as categorias que o Amazon Location Service usa para categorizar e filtrar. Essas categorias são usadas em todos os idiomas, independentemente do parâmetro de idioma estar definido para um idioma diferente.

### Note

O Amazon Location Service mapeia categorias de provedores de dados para esse conjunto de categorias. Se um provedor de dados colocar um local em uma categoria que não faz parte da lista de categorias do Amazon Location Service, a categoria do provedor será incluída nos resultados como uma categoria suplementar.

Tipos de local — Esses tipos são usados para indicar o tipo de correspondência que foi usado para encontrar o resultado.

- `AddressType` — Retornado quando o resultado correspondia a um endereço.
- `StreetType` — Retornado quando o resultado foi comparado a uma rua.
- `IntersectionType` — Retornado quando o resultado foi comparado ao cruzamento de duas ruas.
- `PointOfInterestType` — Retornado quando o resultado correspondia a um ponto de interesse, como uma empresa ou um local cívico.
- `CountryType` — Retornado quando o resultado correspondia a um país ou região principal.
- `RegionType` — Retornado quando o resultado correspondia a uma região dentro de um país, como um estado ou província.
- `SubRegionType` — Retornado quando o resultado correspondia a uma sub-região dentro de um país, como um condado ou área metropolitana.
- `MunicipalityType` — Retornado quando o resultado correspondia a uma cidade ou vila.
- `NeighborhoodType` — Retornado quando o resultado correspondia a um bairro ou área dentro de uma cidade.
- `PostalCodeType` — Retornado quando o resultado correspondia a um código postal.

Categorias de pontos de interesse — Essas categorias são usadas para indicar o tipo de empresa ou localização para obter resultados de pontos de interesse.

- `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

## Limitações de filtragem por provedor de dados

Nem todos os provedores têm a mesma funcionalidade de filtragem. A tabela a seguir descreve as diferenças entre os níveis.

Provedor	APIs com suporte ao filtro	Categorias suportadas para filtragem	Valores de retorno
Esri	SearchPlaceIndexForText , SearchPlaceIndexForSuggestions	Filtrar por Tipos de locais e Categorias de pontos de interesse.	As categorias resultantes são SearchPlaceIndexForText , SearchPlaceIndexFor



Provedor	APIs com suporte ao filtro	Categorias suportadas para filtragem	Valores de retorno
			<code>rPosition</code> , e <code>GetPlace</code>
Here	<code>SearchPlaceIndexForText</code> , <code>SearchPlaceIndexForSuggestions</code>	Filtrar somente por Tipos de local.	As categorias resultantes são <code>SearchPlaceIndexForText</code> e <code>SearchPlaceIndexForSuggestions</code> , <code>SearchPlaceIndexForPosition</code> e <code>GetPlace</code>
Grab	Sem suporte	Sem suporte	Sem suporte
Open Data	n/a (não há suporte para pesquisar locais)	n/a	n/a

## Amazon Aurora PostgreSQL funções definidas pelo usuário para o Amazon Location Service

Você pode usar o Amazon Location Service para trabalhar com coordenadas e endereços armazenados em tabelas de banco de dados para limpar e enriquecer seus dados geoespaciais.

Por exemplo: .

- Você pode usar a geocodificação para converter endereços em coordenadas para normalizar e preencher lacunas nos dados dos endereços armazenados em uma tabela de banco de dados.
- Você pode geocodificar endereços para obter suas posições e usar as coordenadas com funções espaciais do banco de dados, como uma função que mostra linhas em uma área especificada.
- Você pode usar dados enriquecidos para gerar relatórios automatizados, como gerar um relatório automatizado que ilustra todos os dispositivos em uma determinada área ou um relatório

automatizado para machine learning que ilustra áreas com maiores taxas de falha ao enviar atualizações de localização.

Este tutorial mostra como formatar e enriquecer endereços armazenados em uma Amazon Aurora PostgreSQL tabela de banco de dados usando o Amazon Location Service.

- Amazon Aurora PostgreSQL – Um mecanismo de banco de dados relacional totalmente gerenciado, compatível com o MySQL e o PostgreSQL, que gera até cinco vezes o throughput do MySQL e até três vezes o throughput do PostgreSQL, sem alterar a maior parte do aplicativo existente. Para obter mais informações, consulte [O que é o Amazon Aurora?](#) no Guia do usuário do Amazon Aurora.

#### Important

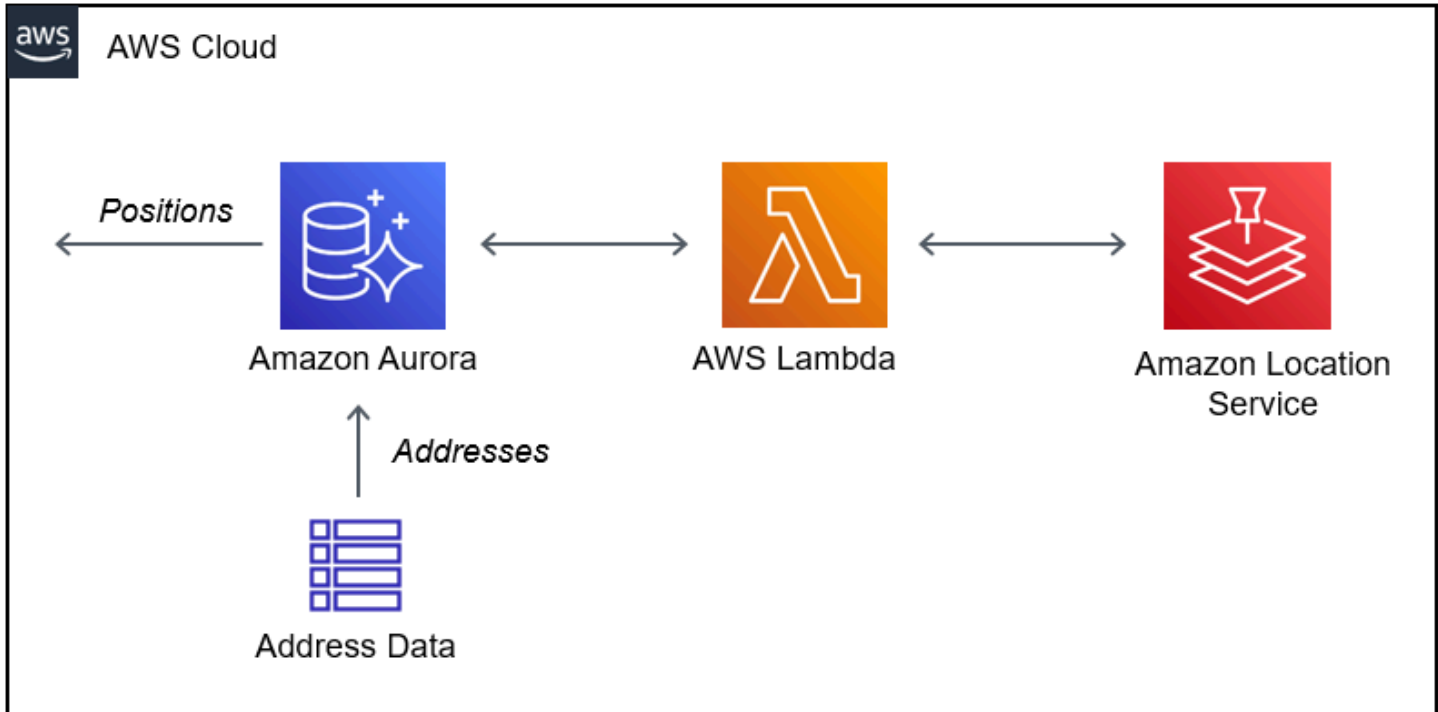
O aplicativo resultante deste tutorial usa um índice de local que armazena os resultados da geocodificação. Para obter informações sobre as cobranças aplicáveis para armazenar resultados de geocodificação, consulte os [preços do Amazon Location Service](#).

O código de amostra está disponível no repositório de amostras do Amazon Location Service em [GitHub](#), que inclui [um AWS CloudFormation modelo](#).

#### Tópicos

- [Visão geral](#)
- [Pré-requisitos](#)
- [Início rápido](#)
- [Criar um recurso de índice de localização](#)
- [Crie uma AWS Lambda função para geocodificação](#)
- [Conceda Amazon Aurora PostgreSQL acesso a AWS Lambda](#)
- [Invocar a função do AWS Lambda](#)
- [Enriquecendo um banco de dados contendo dados de endereço](#)
- [Próximas etapas](#)

## Visão geral



A arquitetura envolve as seguintes integrações:

- Essa solução usa um recurso de índice de local do Amazon Location para fornecer suporte a consultas de geocodificação usando a operação `SearchPlaceIndexForText`.
- AWS Lambda usa um Python Lambda que geocodifica endereços quando uma política do IAM dá permissão para deixar que AWS Lambda chame a operação de geocodificação do Amazon Location, `SearchPlaceIndexForText`.
- Conceda permissão para Amazon Aurora PostgreSQL invocar a função do Lambda de geocodificação usando uma função SQL definida pelo usuário.

## Pré-requisitos

Antes de começar, você precisa dos seguintes pré-requisitos:

- Um cluster Amazon Aurora PostgreSQL. Para obter mais informações sobre [Criar um cluster do banco de dados do Amazon Aurora](#), consulte o Guia do usuário do Amazon Aurora.

**Note**

Se o cluster do Amazon Aurora não estiver disponível publicamente, você também deverá configurar o Amazon Aurora para se conectar AWS Lambda a uma nuvem privada virtual (VPC) na sua conta AWS. Para ter mais informações, consulte [Conceda Amazon Aurora PostgreSQL acesso a AWS Lambda](#).

- Uma ferramenta de desenvolvedor de SQL para se conectar ao cluster Amazon Aurora PostgreSQL.

## Início rápido

Como alternativa às etapas deste tutorial, você pode iniciar uma pilha rápida para implantar uma AWS Lambda função que ofereça suporte à operação [SearchPlaceIndexForText](#) do Amazon Location. Isso configura automaticamente sua conta AWS para permitir que o Amazon Aurora faça chamadas AWS Lambda.

Depois de configurar sua conta AWS, você precisará:

- Adicionar o atributo do Lambda ao Amazon Aurora. Consulte Adicionar o perfil do IAM a um cluster do banco de dados do Amazon Aurora em [Conceda Amazon Aurora PostgreSQL acesso a AWS Lambda](#).
- Carregar a função definida pelo usuário em seu banco de dados. Consulte [Invocar a função do AWS Lambda](#).

A button with a yellow-to-orange gradient background, rounded corners, and a blue play button icon on the right. The text "Launch Stack" is written in white on the left side.

## Criar um recurso de índice de localização

Comece criando um recurso de índice de local para dar suporte às consultas de geocodificação.

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação, escolha Índices de localização.
3. Preencha as seguintes caixas:

- Nome – insira um nome para o recurso de índice de localização. Por exemplo, *AuroraPlaceIndex*. Máximo de 100 caracteres. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados.
  - Descrição – insira uma descrição opcional. Por exemplo, *Índice de local do Amazon Aurora*.
4. Em Provedores de dados, escolha um [provedor de dados](#) disponível para usar com seu recurso de índice de localização. Se você não tiver nenhuma preferência, recomendamos começar com o *Esri*.
  5. Em Opções de armazenamento de dados, especifique Sim, os resultados serão armazenados. Isso indica que você pretende salvar os resultados da geocodificação em um banco de dados.
  6. (Opcional) Em Tags, insira uma tag Chave e Valor. Isso adiciona uma tag ao seu novo recurso de índice de localização. Para obter mais informações, consulte [Marcar seus recursos](#).
  7. Escolha Criar índice de localização.

## Crie uma AWS Lambda função para geocodificação

Para criar uma conexão entre Amazon Aurora PostgreSQL e o Amazon Location Service, você precisa de uma função AWS Lambda para lidar com solicitações do mecanismo de banco de dados. Essa função traduz o evento de função definida pelo usuário do Lambda e chama a operação `SearchPlaceIndexForText` do Amazon Location.

Você pode criar a função usando o console AWS Lambda, o AWS Command Line Interface ou as APIs AWS Lambda.

Para criar uma função do Lambda definida pelo usuário usando o console

1. Abra o console AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, selecione Funções.
3. Selecione Criar função e certifique-se de que Autor do zero esteja selecionado.
4. Preencha as seguintes caixas:
  - Nome da função – insira um nome para a sua função. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados, sem espaços. Por exemplo, *AuroraGeocoder*.
  - Runtime – selecione *Python 3.8*.
5. Selecione Criar função.

6. Selecione a guia Código para abrir o editor.
7. Substitua o código do espaço reservado em `lambda_function.py` pelo seguinte:

```
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. Se você nomeou seu índice de local de forma diferente *AuroraPlaceIndex*, crie uma variável de ambiente chamada `PLACE_INDEX_NAME` para atribuir o nome do recurso a:
  - Selecione a guia Configuração, selecione Variáveis de ambiente.

- Selecione Editare, em seguida, selecione Adicionar variável de ambiente.
  - Em Chave: insira PLACE\_INDEX\_NAME.
  - Em Valor: insira o nome do seu recurso de índice de local.
9. Selecione Implantar para salvar a função atualizada.
  10. Selecione Testar no menu suspenso, selecione Configurar eventos de teste.
  11. Selecione Criar evento de teste.
  12. Insira o seguinte evento de teste:

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Selecione Testar para testar a função do Lambda.
14. Escolha a guia Configuração.
15. Em Configuração geral: selecione Permissões.
16. Em Função de execução: selecione o Nome da função com hiperlink para conceder permissões do Amazon Location Service à sua função do Lambda.
17. Na guia Permissões: selecione Adicionar permissões no menu suspenso e, em seguida, selecione Criar política em linha.
18. Selecione a guia JSON.
19. Adicione a política do IAM a seguir:
  - A política a seguir dá permissão para enviar SearchPlaceIndexForText para o recurso de índice de locais *AuroraPlaceIndex*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:SearchPlaceIndexForText",
      "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
    }
  ]
}
```

20. Escolha Revisar política.
21. Insira um nome de política. Por exemplo, *AuroraPlaceIndexReadOnly*.
22. Escolha Criar política.

## Conceda Amazon Aurora PostgreSQL acesso a AWS Lambda

Antes de Amazon Aurora PostgreSQL poder invocar uma função AWS Lambda, você deve conceder permissão de acesso.

Se seu cluster Amazon Aurora PostgreSQL não estiver acessível publicamente, você precisará, primeiro, criar um endpoint da VPC para AWS Lambda para que o Amazon Aurora chame sua função do Lambda.

Crie um endpoint da VPC para AWS Lambda

### Note

Essa etapa só é necessária se seu cluster Amazon Aurora PostgreSQL não estiver acessível publicamente.

1. Abra a [Amazon Virtual Private Cloud Console](#).
2. No painel de navegação à esquerda, selecione Endpoints.
3. Selecione Criar endpoint.
4. No filtro Nome de serviço, insira “lambda” e, em seguida, selecione `com.amazonaws.<region>.lambda`.
5. Escolha a VPC que contém seu cluster do Aurora.
6. Selecione uma sub-rede para cada zona de disponibilidade.
7. No filtro do Grupo de segurança, insira “default” ou o nome do grupo de segurança do qual seu cluster do Aurora é membro e selecione o grupo de segurança.
8. Selecione Criar endpoint.

Crie uma política do IAM para conceder permissão para invocar sua função AWS Lambda

1. Abra o [console do IAM](#).
2. No painel de navegação à esquerda, expanda Gerenciamento de acesso e selecione Políticas.



3. Escolha Criar política.
4. Na guia JSON, insira a seguinte política:
  - Veja a seguir um exemplo de uma política do IAM que concede Amazon Aurora PostgreSQL permissão para invocar a função `AuroraGeocoder` AWS Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Selecione Próximo: Tags para adicionar tags opcionais.
6. Escolha Próximo: Revisar.
7. Revise sua política e insira os seguintes detalhes para a política:
  - Nome – Use caracteres alfanuméricos e '+=,.@-\_' . Máximo de 128 caracteres. Por exemplo, *AuroraGeocoderInvoke*.
  - Descrição – insira uma descrição opcional. Use caracteres alfanuméricos e '+=,.@-\_' . Máximo de 1000 caracteres.
8. Escolha Criar política. Observe o ARN dessa política, que você usa para vincular a política a um perfil do IAM.

Crie um perfil do IAM para dar permissão ao Amazon Relational Database Service (Amazon RDS)

Ao criar um perfil do IAM, Amazon Aurora PostgreSQL você pode assumir a função em seu nome para acessar sua função do Lambda. Para ter mais informações, consulte [Criar um perfil para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.

O exemplo a seguir é um AWS CLI comando que cria uma função chamada *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"
    }
  ]
}'
```

Anexe a política do IAM ao perfil do IAM

Quando você tiver um perfil do IAM, anexe a política do IAM que você criou.

O exemplo a seguir é um AWS CLI comando que anexa a política *AuroraGeocoderInvoke* à função *AuroraGeocoderInvokeRole*.

```
aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-
name AuroraGeocoderInvokeRole
```

Adicione o perfil do IAM a um cluster do banco de dados do Amazon Aurora

O exemplo a seguir é um AWS CLI comando para adicionar uma função do IAM a um cluster de Amazon Aurora PostgreSQL banco de dados chamado *MyAuroraCluster*.

```
aws rds add-role-to-db-cluster \
--db-cluster-identifier MyAuroraCluster \
--feature-name Lambda \
--role-arn AuroraGeocoderInvokeRole \
--region your-region
```

## Invocar a função do AWS Lambda

Depois de conceder permissão para Amazon Aurora PostgreSQL invocar sua função do Lambda de geocodificação, você pode criar uma Amazon Aurora PostgreSQL função definida pelo usuário para invocar a função AWS Lambda de geocodificação. Para obter mais informações, consulte [Invocando uma AWS Lambda função a partir de um Amazon Aurora PostgreSQL cluster de banco de dados](#) no Guia do usuário do Amazon Aurora.

## Instale as extensões necessárias do PostgreSQL

Para instalar as extensões `aws_lambda` e `aws_commons` necessárias do PostgreSQL, consulte [Visão geral do uso de uma função do Lambda](#) no Guia do usuário do Amazon Aurora.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

## Instale as extensões necessárias do PostGIS

PostGIS é uma extensão do PostgreSQL para armazenar e gerenciar informações espaciais. Para obter mais informações, consulte [Trabalhando com e extensão PostGIS](#) no Guia do usuário do Amazon Relational Database Service.

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

## Crie uma função SQL definida pelo usuário que invoque a função do Lambda

Em um editor SQL, crie uma nova função definida pelo usuário `f_SearchPlaceIndexForText` para invocar a função: *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;
$function$;

```

Chame a função SQL para geocodificar a partir do Aurora

A execução da instrução SQL invoca a função Lambda *AuroraGeocoder*, que pega registros de endereço da tabela do banco de dados no banco de dados e os codifica geograficamente usando um recurso de índice de local. Amazon Aurora PostgreSQL

### Note

Amazon Aurora PostgreSQL invoca a função do Lambda para cada chamada para a função SQL definida pelo usuário.

Se você estiver geocodificando 50 linhas, Amazon Aurora PostgreSQL invoca a função do Lambda 50 vezes. Uma invocação para cada linha.

A função `f_SearchPlaceIndexForText` SQL a seguir faz solicitações à [SearchPlaceIndexForText](#) API do Amazon Location por meio da função *AuroraGeocoder* Lambda. A função retorna uma geom coluna que é uma geometria PostGIS, `ST_AsText(geom)` que é convertida em texto.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

Por padrão, o retorno conterá uma linha. Para solicitar linhas adicionais, até o `MaxResults` limite, execute a instrução do SQL a seguir, fornecendo um `BiasPosition` e limitando os resultados ao Canadá.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Para filtrar os resultados usando uma caixa delimitadora, passe um [Box2D](#) como `filter_bbox`:

- [FilterBBox](#) – Filtra os resultados retornando lugares dentro de uma caixa delimitadora. Esse parâmetro é opcional.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)::box2d, '{"CAN"}', 5);
```

Para obter mais informações sobre os tipos e funções do PostGIS, consulte a [Referência do PostGIS](#).

## Enriquecendo um banco de dados contendo dados de endereço

Você pode estruturar um endereço formatado e, simultaneamente, normalizar e geocodificar usando a operação `SearchPlaceIndexForText` do Amazon Location, dada uma tabela de banco de dados com os seguintes dados divididos nas seguintes colunas:

- `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;
```

O exemplo a seguir ilustra uma linha de dados resultante na tabela:

```
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)
```

Atualize a tabela do banco de dados e preencha as colunas

O exemplo a seguir atualiza a tabela e preenche as colunas com os resultados das SearchPlaceIndexForText consultas:

```
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;
```

## Próximas etapas

O código de amostra está disponível no repositório de amostras do Amazon Location Service em [GitHub](#), que inclui [um AWS CloudFormation modelo](#).

## Gerenciar seus recursos do índice de locais

Você pode gerenciar seus recursos do índice de locais usando o console de Amazon Location AWS CLI ou as APIs do Amazon Location.

### Liste os recursos do seu índice de locais

Você pode visualizar a lista de recursos do seu índice de locais usando o console do Amazon Location AWS CLI ou as APIs do Amazon Location:

#### Console

Como visualizar uma lista de recursos do índice de locais usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Índices de locais no painel de navegação à esquerda.
3. Veja uma lista de recursos do seu índice de locais em Meus índices de locais.

#### API

Use a operação [ListPlaceIndexes](#) das APIs de localização do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter uma lista de recursos do índice de locais na conta AWS.

```
POST /places/v0/list-indexes
```

Este é um exemplo de resposta para [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

Use o comando [list-place-indexes](#).

O exemplo a seguir é um AWS CLI para obter uma lista de recursos do índice de locais na conta AWS.

```
aws location list-place-indexes
```

## Veja detalhes do recurso do índice de locais

Você pode obter detalhes sobre qualquer recurso de índice de locais em sua conta AWS usando o console de localização do Amazon Location, o AWS CLI ou as APIs de do Amazon Location:

### Console

Como visualizar os detalhes de um recurso do índice de locais usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Índices de locais no painel de navegação à esquerda.
3. Em Meus índices de locais, selecione o link do nome do recurso no índice de locais de destino.

### API

Use a operação [DescribePlaceIndex](#) nas APIs do Amazon Location Place.

O exemplo a seguir é uma solicitação de API para obter os detalhes do recurso place index para *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

Este é um exemplo de resposta para [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

Use o comando [describe-place-index](#).

O exemplo a seguir é AWS CLI para obter os detalhes do recurso de índice de locais *ExamplePlaceIndex*.

```
aws location describe-place-index \
  --index-name "ExamplePlaceIndex"
```

## Excluir um recurso do índice de locais

Você pode excluir um recurso do índice de locais da sua conta AWS usando o console de Amazon Location, o AWS CLI ou as APIs do Amazon Location:

### Console

Como excluir um recurso do índice de locais usando o console do Amazon Location

#### Warning

Essa operação exclui o recurso permanentemente.

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Índices de locais no painel de navegação à esquerda.
3. Em Meus índices de locais, selecione o recurso do índice de locais de destino.
4. Selecione Excluir índice de locais.

## API

Use a operação [DeletePlaceIndex](#) das APIs de localização do Amazon Location.

O exemplo a seguir é uma solicitação de API para excluir o recurso place index

*ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

Veja a seguir um exemplo de uma resposta bem-sucedida do [DeletePlaceIndex](#):

```
HTTP/1.1 200
```

## CLI

Use o comando [delete-place-index](#).

O exemplo a seguir é um AWS CLI comando para excluir o recurso place index

*ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```


## Calcular rotas usando o Amazon Location Service

O Amazon Location permite que você selecione um provedor de dados para calcular uma rota criando e configurando um recurso de calculadora de rotas.

Você pode usar o recurso de calculadora de rotas para [calcular uma rota](#) com parâmetros específicos usando o SDK AWS ou os endpoints da API REST. Use esse recurso de calculadora de

rotas para calcular rotas entre uma origem, um destino e até 23 pontos de referência para diferentes modos de transporte, evasões e condições de tráfego.

Você também pode usar o recurso de calculadora de rotas para criar entradas para seus algoritmos ou produtos de planejamento de rotas [calculando uma matriz de rotas](#). Calcule o tempo de viagem e a distância de viagem entre um conjunto de posições de partida e um conjunto de posições de destino. O software de planejamento de rotas pode usar esses dados de tempo e distância para otimizar uma rota ou um conjunto de rotas; por exemplo, se você estiver planejando várias rotas de entrega e quiser encontrar a melhor rota e horário para cada parada. Você pode calcular uma matriz de rotas para diferentes modos de transporte, evasões e condições de tráfego.

 Note

Para obter uma visão geral dos conceitos de roteamento, consulte [Rotas](#).

## Tópicos

- [Pré-requisitos](#)
- [Calcular uma rota](#)
- [Planejamento de rotas com uma matriz de rotas](#)
- [Posições não localizadas em uma estrada](#)
- [Horário de partida](#)
- [Modo de viagem](#)
- [Gerenciando seus recursos da calculadora de rotas](#)

## Pré-requisitos

Antes de começar a calcular as rotas, siga as etapas de pré-requisito:

### Tópicos

- [Criar um recurso de calculadora de rotas](#)
- [Autenticando suas solicitações](#)

## Criar um recurso de calculadora de rotas

Antes de calcular uma rota, crie um recurso de calculadora de rotas em sua conta AWS.

Ao criar um recurso de calculadora de rotas, você pode escolher entre os provedores de dados disponíveis:

1. Esri – para obter mais informações sobre a cobertura da Esri na região de seu interesse, consulte [Detalhes da Esri sobre redes de ruas e cobertura de trânsito](#).
2. HERE Technologies – para obter informações adicionais sobre a cobertura da HERE em sua região de interesse, consulte [Cobertura de roteamento de carros da HERE](#) e [Cobertura de roteamento de caminhões da HERE](#).
3. Grab — Para obter mais informações sobre a cobertura do Grab, consulte [Países/regiões e área de cobertura](#).

#### Note

Se a aplicação estiver rastreando ou roteando os ativos que você usa em sua empresa, como veículos de entrega ou funcionários, você não deverá usar a Esri como provedor de geolocalização. Consulte a seção 82 dos [Termos de serviço da AWS](#) para obter mais detalhes.

Isso pode ser feito usando o console do Amazon Location Service, a AWS CLI, ou as APIs do Amazon Location.

#### Console

Para criar um recurso de calculadora de rotas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Calculadoras de rotas.
3. Escolha Criar calculadora de rotas.
4. Preencha as seguintes caixas:
  - Nome – insira um nome para o recurso da calculadora de rotas. Por exemplo, *ExampleCalculator*. Máximo de 100 caracteres. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados.
  - Descrição – insira uma descrição opcional.
5. Para Provedores de dados, escolha um [Provedor de dados](#) para usar como calculadora de rotas.

- (Opcional) Em Tags, insira uma tag Chave e Valor. Isso adiciona uma tag ao seu novo recurso de calculadora de rotas. Para obter mais informações, consulte [Marcar seus recursos](#).
- Escolha Criar calculadora de rotas.

## API

Para criar um recurso de calculadora de rotas usando as APIs do Amazon Location

Use a operação [CreateRouteCalculator](#) das APIs de localização do Amazon Location.

O exemplo a seguir é uma solicitação de API para criar um recurso de calculadora de rotas chamado *ExampleCalculator* usando o provedor de dados *Esri*.

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

## AWS CLI

Para criar um recurso de calculadora de rotas usando comandos do AWS CLI

Use o comando `create-route-calculator`.

O exemplo a seguir cria um recurso de calculadora de rotas chamado *ExampleCalculator* usando a *Esri* como provedor de dados.

```
aws location \
  create-route-calculator \
  --calculator-name "ExampleCalculator" \
  --data-source "Esri" \
  --tags Tag1=Value1
```

**Note**

O faturamento depende do seu uso. Você pode incorrer em taxas pelo uso de outros serviços da AWS. Para obter mais informações, consulte os [preços do Amazon Location Service](#).

## Autenticando suas solicitações

Depois de criar um recurso de calculadora de rotas e estiver pronto para começar a criar recursos de localização em seu aplicativo, escolha como você autenticaria suas solicitações:

- Para explorar as formas para acessar os serviços, consulte [Acessando o Amazon Location Service](#).
- Se você tem um site com usuários anônimos, talvez queira usar chaves de API ou o Amazon Cognito.

### Exemplo

O exemplo a seguir mostra o uso de uma chave de API para autorização, o [AWS JavaScript SDK v3](#) e o Amazon Location. [JavaScript Auxiliar de autenticação](#)

```
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);
```

## Calcular uma rota

Você pode usar o Amazon Location Service para calcular rotas entre uma origem e um destino, com até 23 pontos de referência ao longo da rota, para diferentes modos de transporte, evasões e condições de tráfego.

### Note

Primeiro, você deve criar um recurso de calculadora de rotas e configurar a autenticação para suas solicitações para o Amazon Location. Para ter mais informações, consulte [Pré-requisitos](#).

## Começar a calcular rotas

Envie uma solicitação simples usando a operação [CalculateRoute](#). Uma solicitação simples contém os seguintes campos obrigatórios:

- `DeparturePosition` — A posição inicial a partir da qual calcular a rota. Definido como `[longitude, latitude]`
- `DestinationPosition` — A posição final para a qual calcular a rota. Definido como `[longitude, latitude]`.

### Note

Se você especificar uma posição de partida ou destino que não esteja localizada em uma estrada, o Amazon Location [moverá a posição para a estrada mais próxima](#).

Opcionalmente, você pode especificar [pontos de referência](#), um [horário de partida](#) e um [modo de viagem](#) em sua solicitação.

Você pode usar a AWS CLI ou as APIs do Amazon Location.



## API

O exemplo a seguir é uma `CalculateRoute` solicitação usando o recurso de calculadora de rotas *ExampleCalculator*. A solicitação especifica o cálculo de uma rota de uma posição de partida `[-122, 7565, 49, 0021]` até uma posição de destino `[-122, 3394, 47, 6159]`.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565, 49.0021],
  "DestinationPosition": [-122.3394, 47.6159]
}
```

## AWS CLI

O exemplo a seguir é um `calculate-route` comando usando o recurso de calculadora de rotas *ExampleCalculator*. A solicitação especifica o cálculo de uma rota de uma posição de partida `[-122, 7565, 49, 0021]` até uma posição de destino `[-122, 3394, 47, 6159]`.

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

Por padrão, a resposta resulta na `Distance` em quilômetros. Você pode alterar a unidade de medida para milhas usando o seguinte parâmetro opcional:

- `DistanceUnit` — Especifica o sistema de unidades a ser usado para os resultados da distância.

## Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565, 49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

## Definindo pontos de referência

Ao calcular uma rota, você pode especificar até 23 pontos de parada intermediários entre a posição de partida e a posição de destino usando as posições do ponto de referência.

- `WaypointPositions` — Especifica uma lista ordenada de posições intermediárias a serem incluídas ao longo de uma rota entre a posição de partida e a posição de destino.

### Note

Se você especificar uma posição de ponto de referência que não esteja localizada em uma estrada, o Amazon Location moverá a posição para a estrada mais próxima.

## Example

A solicitação [CalculateRoute](#) a seguir calcula uma rota com 2 pontos de referência:

- A posição de partida é `[-122,7565, 49,0021]` e a posição de destino é `[-122,3394, 47,6159]`.
- Para o parâmetro de solicitação `WaypointPositions`:
  - A primeira posição de parada é `[-122,1884, 48,0936]`.
  - A segunda posição de parada é `[-122,3493, 47,6205]`.
- Para incluir a geometria da linestring do trecho entre esses dois pontos de referência, defina o seguinte parâmetro opcional como *verdadeiro*:
  - `IncludeLegGeometry` — Inclui a geometria de cada trecho entre um par de posições na resposta.

```
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
}
```

## Exemplo de resposta

Veja a seguir um exemplo de solicitação com a resposta correspondente ao chamar a operação [CalculateRoute](#) da API de rotas do Amazon Location com o `IncludeLegGeometry` definido como *verdadeiro*, que inclui a geometria da linestring de cada trecho entre um par de posições na resposta.

### 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
  ]
}
```

## Planejamento de rotas com uma matriz de rotas

Você pode usar o Amazon Location Service para criar entradas para seu software de planejamento e otimização de rotas. Você pode criar resultados de rotas, incluindo tempo e distância de viagem, para rotas entre um conjunto de posições de partida e um conjunto de posições de destino.

Por exemplo, considerando as posições de partida A e B e as posições de destino X e Y, o Amazon Location Service retornará o tempo e a distância da viagem para rotas de A a X, A a Y, B a X e B a Y.

Você pode calcular as rotas com diferentes modos de transporte, evasões e condições de tráfego. Por exemplo, você pode especificar que o veículo é um caminhão com 11 metros de comprimento, e a rota calculada usará essas restrições para determinar o tempo e a distância da viagem.

O número de resultados (e rotas calculadas) é o número de posições de partida multiplicado pelo número de posições de destino. Você é cobrado por cada rota calculada, não por cada solicitação ao serviço. Assim sendo, uma matriz de rotas com 10 partidas e 10 destinos será cobrada como 100 rotas.

### Cálculo de uma matriz de rotas

Você pode calcular uma matriz de rotas entre um conjunto de posições de partida e um conjunto de posições de destino. Os resultados da rota incluirão o tempo de viagem e a distância da viagem.

#### Pré-requisito

- Primeiro, você deve criar um recurso de calculadora de rotas e configurar a autenticação para suas solicitações para o Amazon Location. Para ter mais informações, consulte [Pré-requisitos](#).

Envie uma solicitação usando a operação [CalculateRouteMatrix](#). Uma solicitação mínima contém os seguintes campos obrigatórios:

- **DeparturePositions** — O conjunto de posições iniciais para as quais calcular as rotas. Definido como uma matriz de [longitude, latitude]
- **DestinationPositions** — O conjunto de posições finais para as quais calcular as rotas. Definido como uma matriz de [longitude, latitude].

#### Note

Se você especificar uma posição de partida ou destino que não esteja localizada em uma estrada, o Amazon Location [moverá a posição para a estrada mais próxima](#).

Opcionalmente, você pode especificar um [horário de partida](#) e um [modo de viagem](#) em sua solicitação.

Você pode usar a AWS CLI ou as APIs do Amazon Location.

#### API

O exemplo a seguir é uma `CalculateRouteMatrix` solicitação usando o recurso de calculadora de rotas *ExampleCalculator*. A solicitação especifica o cálculo da matriz de rotas das posições de partida `[-122,7565,49,0021]` e `[-122,2014,47,6101]` até as posições de destino `[-122,3394,47,6159]` e `[-122,4813,48,7511]`.

```
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

O exemplo a seguir é um `calculate-route-matrix` comando usando o recurso de calculadora de rotas *ExampleCalculator*. A solicitação especifica o cálculo da matriz de rotas das posições de partida `[-122, 7565, 49, 0021]` e `[-122, 2014, 47, 6101]` até as posições de destino `[-122, 3394, 47, 6159]` e `[-122, 4813, 48, 7511]`.

```
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]]"
```

Por padrão, a resposta resulta na Distance em quilômetros. Você pode alterar a unidade de medida para milhas usando o seguinte parâmetro opcional:

- `DistanceUnit` — Especifica o sistema de unidades a ser usado para os resultados da distância.

### 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"
}
```

### Restrições nas posições de partida e destino

Ao calcular uma matriz de rotas, há restrições nas posições de partida e destino. Essas restrições variam de acordo com o provedor usado pelo recurso `RouteCalculator`.

Limitação	Esri	Grab	HERE
Número de posições	Até 10 posições de partida e 10 posições de destino.	Até 350 posições de partida e 350 posições de destino.	Até 350 posições de partida e 350 posições de destino.  Para rotas mais longas, restrições adicionais se aplicam. Consulte a <a href="#">seção</a> .
Distância entre posições	Qualquer par de posições de partida e destino deve estar a 400 km uma da outra (40 km para rotas de pedestres).		Todas as posições de partida e destino devem estar dentro de um círculo de 180 km de diâmetro.  Para rotas mais longas, restrições adicionais se aplicam. Consulte a <a href="#">seção</a> .
Comprimento da rota	As rotas não serão concluídas se o tempo total de viagem da rota for superior a 400 minutos.		As rotas que se desviam mais de 10 km fora de um círculo ao redor dos pontos de partida e destino não serão calculadas.  Para rotas mais longas, restrições adicionais se aplicam. Consulte a <a href="#">seção</a> .
Regiões	O cálculo de uma matriz de rotas não é suportado na Coreia.	Disponível no sudeste da Ásia. Para obter uma lista de países/	Sem restrições adicionais.

Limitação	Esri	Grab	HERE
		regiões compatíveis e obter mais informações, consulte <a href="#">Países/regiões e área de cobertura</a> .	

## Planejamento de rotas mais longas

O cálculo de uma matriz de resultados de rotas é útil para um planejamento eficiente de rotas, mas o cálculo pode levar algum tempo. Todos os provedores de dados do Amazon Location Service limitam o número de rotas ou a distância das rotas que podem ser calculadas. Por exemplo, o HERE permite criar rotas entre 350 posições de partida e destino, mas essas posições devem estar dentro de um círculo de 180 km. E se você quiser planejar com rotas mais longas?

Você pode calcular uma matriz de rotas com comprimentos irrestritos para um número menor de rotas, usando a `RouteCalculator` com o HERE como provedor de dados. Isso não muda a forma como você chama a API [CalculateRouteMatrix](#). O Amazon Location simplesmente permite rotas mais longas quando você atende aos requisitos.

Os requisitos para cálculos de rotas de maior extensão são:

- A `RouteCalculator` deve usar o provedor de dados HERE.
- O número de posições de partida não deve ser maior que 15.
- O número total de rotas a serem calculadas não deve ser maior que 100.
- O roteamento de longa distância não é permitido para roteamento de caminhões sem pedágio quando as rotas são maiores que 1.000 km. Essa combinação é mais lenta de calcular e pode fazer com que a chamada atinja o tempo limite. Você pode calcular essas rotas individualmente com a [CalculateRoute](#) operação.

Se sua chamada não atender a esses requisitos (por exemplo, você está solicitando 150 cálculos de rota em uma única chamada), a `CalculateRouteMatrix` voltará a permitir apenas as regras de rota mais curtas. Você pode então calcular as rotas, desde que as posições estejam dentro de um círculo de 180 km.

Ao calcular rotas mais longas, tenha em mente estes pontos:



- Rotas mais longas podem levar mais tempo para serem calculadas, até mais do que o tempo máximo das APIs do Amazon Location. Se você tiver tempos limite frequentes com rotas específicas, poderá tentar um número menor de rotas em cada chamada para a `CalculateRouteMatrix`.
- Se você adicionar mais posições de destino ou de partida à sua solicitação `CalculateRouteMatrix`, a operação poderá mudar para o modo mais restrito e você poderá receber um erro em uma rota que pode ser calculada sem problemas quando há menos rotas para criar. Nesse caso, reduza o número de posições de destino ou de partida e faça várias solicitações para obter o conjunto completo de cálculos de rota de que você precisa.

## Exemplo de resposta

Veja a seguir um exemplo de solicitação com a resposta correspondente ao chamar a operação [CalculateRouteMatrix](#) a partir da API de rotas do Amazon Location.

### 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"
  }
}

```

## Posições não localizadas em uma estrada

Ao usar a `CalculateRoute` ou a `CalculateRouteMatrix`, se você especificar uma posição de partida, destino ou ponto de passagem que não esteja localizado em uma estrada, o Amazon Location moverá a posição para uma estrada próxima.

A solicitação [CalculateRoute](#) a seguir especifica uma posição de partida e uma posição de destino que não estão localizadas em uma estrada:

```

POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-123.128014, 49.298472],
  "DestinationPosition": [-123.134701, 49.294315]
}

```

A resposta resultante retorna uma posição que foi ajustada para uma estrada próxima:

```
{
  "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"
}
}

```

## Horário de partida

Por padrão, quando você chama a `CalculateRoute` ou a `CalculateRouteMatrix`, se não fornecer um horário de partida na solicitação, as rotas calculadas refletem as condições ideais de tráfego.

Você pode definir um horário de partida específico para usar as condições de tráfego ao vivo e preditivas do provedor de dados escolhido, usando uma das seguintes opções:

- `DepartNow` — Quando definido como *verdadeiro*, ele usa condições de tráfego ao vivo para calcular o caminho de viagem mais rápido.

- **DepartureTime** — Quando fornecido, ele usa condições de tráfego preditivas e conhecidas para o horário solicitado. Definido no seguinte [formato](#): YYYY-MM-DDThh:mm:ss.sssZ.

## Example

A solicitação [CalculateRoute](#) a seguir define o horário de partida para 2 de julho de 2024, às 12:15:20 UTC.

```
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,
}
```

## Modo de viagem

Você pode definir um modo de viagem ao usar a `CalculateRoute` ou a `CalculateRouteMatrix`. O modo de viagem afeta a velocidade de viagem e a compatibilidade rodoviária. Embora o modo de viagem padrão seja de carro, você pode especificar qual modo de viagem está usando ao viajar ao longo de uma rota com o seguinte parâmetro opcional:

- **TravelMode** — Especifica o modo de transporte ao calcular uma rota, como: *Bicycle*, *Car*, *Motorcycle*, *Truck* ou *Walking*.

## Limitações

- Se você especificar o modo de viagem `Walking` e seu provedor de dados for a Esri, o início e o destino devem estar dentro de 40 km.
- `Bicycle` ou `Motorcycle` estão disponíveis somente ao usar o Grab como provedor de dados.
- O Grab fornece apenas rotas de `Bicycle` e `Walking` em determinadas cidades. Para ter mais informações, consulte [Países/regiões e área de cobertura](#).

- Truck não está disponível ao usar o Grab como provedor de dados.

### Preferências adicionais

Se você especificar um `TravelMode` de *Car*, poderá especificar preferências de rota adicionais com o seguinte parâmetro opcional:

- `CarModeOptions` — Especifica as preferências de rota ao viajar em um carro, como *AvoidFerries* ou *AvoidTolls*.

Se você especificar um `TravelMode` de *Truck*, poderá especificar preferências de rota adicionais com o seguinte parâmetro opcional:

- `TruckModeOptions` — Especifica as preferências de rota ao viajar em um caminhão como *AvoidFerries* ou *AvoidTolls*, além de especificar rotas que podem acomodar *TruckDimensions* e *TruckWeight*.

### Example

A solicitação [CalculateRoute](#) a seguir especifica *Truck* como o modo de viagem. Restrições adicionais de rota incluem: evitar rotas que usam balsas e evitar estradas que não possam acomodar as dimensões e o peso do caminhão.

```
{
  "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"
    }
  }
}
```

```
}  
  }  
}
```

## Gerenciando seus recursos da calculadora de rotas

Você pode gerenciar seus recursos da calculadora de rotas usando o console Amazon Location, o AWS CLI ou as APIs do Amazon Location.

### Faça uma lista dos seus recursos da calculadora de rotas

Você pode visualizar sua lista de calculadoras de rotas usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location:

#### Console

Para ver uma lista de calculadoras de rotas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Calculadoras de rotas no painel de navegação à esquerda.
3. Veja os detalhes da calculadora de rotas em Minhas calculadoras de rotas.

#### API

Use a operação [ListRouteCalculators](#) das APIs do Amazon Location Routes.

O exemplo a seguir é uma solicitação de API para obter uma lista de calculadoras de rotas na conta AWS.

```
POST /routes/v0/list-calculators
```

Este é um exemplo de resposta para [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

Use o comando `list-route-calculators`.

O exemplo a seguir é um AWS CLI para obter uma lista de calculadoras de rotas na conta AWS.

```
aws location list-route-calculators
```

## Obtenha os detalhes da calculadora de rotas

Você pode obter detalhes sobre qualquer recurso das calculadoras de rotas em sua conta AWS usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location:

### Console

Para ver os detalhes de uma calculadora de rotas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Calculadoras de rotas no painel de navegação à esquerda.
3. Em Minhas calculadoras de rotas, selecione o link do nome da calculadora de rotas de destino.

### API

Use a operação [DescribeRouteCalculator](#) das APIs do Amazon Location Routes.

O exemplo a seguir é uma solicitação de API para obter os detalhes da calculadora de rotas *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```

Este é um exemplo de resposta para [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

Use o comando `describe-route-calculator`.

O exemplo a seguir é AWS CLI para obter os detalhes da calculadora de rotas *ExampleCalculator*.

```
aws location describe-route-calculator \
  --calculator-name "ExampleCalculator"
```

## Exclua uma calculadora de rotas

Você pode excluir uma calculadora de rotas da sua conta AWS usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location:

### Console

Para excluir uma calculadora de rotas usando o console do Amazon Location

#### Warning

Essa operação exclui o recurso permanentemente.

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Calculadoras de rotas no painel de navegação à esquerda.
3. Em Minhas calculadoras de rotas, selecione a calculadora da rotas de destino.

#### 4. Selecione Excluir calculadora de rotas.

### API

Use a operação [DeleteRouteCalculator](#) das APIs do Amazon Location Routes.

O exemplo a seguir é uma solicitação de API para excluir a coleção de cercas geográficas.

*ExampleCalculator*

```
DELETE /routes/v0/calculators/ExampleCalculator
```

Este é um exemplo de resposta para [DeleteRouteCalculator](#):

```
HTTP/1.1 200
```

### CLI

Use o comando `delete-route-calculator`.

O exemplo a seguir é um AWS CLI comando para excluir a coleção de cercas geográficas.

*ExampleCalculator*

```
aws location delete-route-calculator \  
    --calculator-name "ExampleCalculator"
```

## Colocando geocercas em uma área de interesse usando o Amazon Location

Um aplicativo para geocercamento avalia a posição de um dispositivo rastreado em relação às áreas de interesse registradas anteriormente. Isso permite que ações sejam tomadas com base nas atualizações de posição. Por exemplo, você pode iniciar um evento que solicita uma notificação quando um cliente que pediu um café em seu aplicativo móvel estiver perto de uma loja.

### Note

Para obter uma visão geral dos conceitos de geocercamento e rastreador, consulte [Geocercas e rastreadores](#).

Esta seção do guia fornece step-by-step instruções para criar um aplicativo de delimitação geográfica usando o Amazon Location Service.

### Visão geral das etapas

1. Adicione geocercas em torno de áreas de interesse e armazene-as em um recurso de coleta de geocercas.
2. Comece a rastrear seus dispositivos-alvo e armazene o histórico de localização do dispositivo em um recurso de rastreador.
3. Vincule seu recurso de rastreador ao seu recurso de coleta de geocercas para que as atualizações de localização do dispositivo sejam avaliadas automaticamente em relação a todas as suas geocercas.
4. Você pode avaliar as posições dos dispositivos diretamente em relação aos seus recursos de coleta de geocercas se não quiser usar o Amazon Location Trackers para manter o histórico de localização de seus dispositivos.

Depois de implementar sua solução para geocercas, seu recurso de coleta de geocercas emite os seguintes eventos:

- ENTER: um dispositivo rastreado entra em uma geocerca que está dentro de uma coleção de geocercas.
- EXIT: um dispositivo rastreado sai de uma geocerca que está dentro de uma coleção de geocercas.

Você pode usar EventBridge a Amazon para reagir a eventos roteando-os para outro lugar.

Como alternativa ao envio de atualizações por meio do Amazon Location Service APIs de cada dispositivo, você pode usar MQTT para enviar atualizações do dispositivo.

Os tópicos a seguir descrevem essas etapas e alternativas em detalhes.

### Tópicos

- [Adicione geocercas](#)
- [Inicie o rastreamento](#)
- [Vincule um rastreador a uma coleção de geocercas](#)
- [Avalie as posições dos dispositivos em relação às geocercas](#)

- [Verifique as posições do dispositivo](#)
- [Reagindo aos eventos do Amazon Location Service com a Amazon EventBridge](#)
- [Rastreamento usando AWS IoT e MQTT com o Amazon Location Service](#)
- [Gerenciando seus recursos de coleta de geocercas](#)
- [Gerenciando seus recursos de rastreador](#)
- [Exemplo de aplicativo móvel de geofencing and tracking](#)

## Adicione geocercas

As geocercas contêm pontos e vértices que formam um limite fechado, que define uma área de interesse. As coleções de geocercas armazenam e gerenciam uma ou várias geocercas.

[As coleções de cercas geográficas de localização da Amazon armazenam cercas geográficas definidas usando um formato de dados geoespaciais padrão chamado Geo \(7946\). JSON RFC](#) Você pode usar ferramentas, como [geojson.io](#), gratuitamente para desenhar suas cercas geográficas graficamente e salvar o arquivo Geo de saída. JSON

### Note

O Amazon Location não oferece suporte para polígonos com furos, multipolígonos, polígonos no sentido horário e geocercas que cruzam o antimeridiano.

## Criar uma coleção de geofences

Crie uma coleção de cercas geográficas para armazenar e gerenciar cercas geográficas usando o console de localização da Amazon, o AWS CLI ou o Amazon Location. APIs

### Console

Para criar uma coleção de geocercas usando o console Amazon Location

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Coleções de geocercas.
3. Selecione Criar coleção de geocercas.
4. Preencha as seguintes caixas:

- Nome: insira um nome exclusivo. Por exemplo, *ExampleGeofenceCollection*. Máximo de 100 caracteres. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados.
  - Descrição: insira uma descrição opcional para diferenciar seus recursos.
5. Em EventBridge regra com CloudWatch como destino, você pode criar uma EventBridge regra opcional para começar a [reagir aos eventos de delimitação geográfica](#). Isso permite que o Amazon Location publique eventos no Amazon CloudWatch Logs.
  6. (Opcional) Em Tags, insira uma tag Chave e Valor. Isso adiciona uma tag à sua nova coleção de geocercas. Para obter mais informações, consulte [Marcar os recursos do Amazon Location Service](#).
  7. (Opcional) Em Criptografia de chave gerenciada pelo cliente, você pode selecionar Adicionar uma chave gerenciada pelo cliente. Isso adiciona uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia por meio da criptografia padrão de AWS propriedade. Para obter mais informações, consulte [Criptografia de dados em repouso](#).
  8. Selecione Criar coleção de geocercas.

## API

Para criar uma coleção de cercas geográficas usando a localização da Amazon APIs

Use a [CreateGeofenceCollection](#) operação da Amazon Location Geofences APIs.

O exemplo a seguir usa uma API solicitação para criar uma coleção de geofence chamada *ExampleGeofenceCollection*. A coleção de cercas geográficas está associada a uma [AWS KMS chave gerenciada pelo cliente para criptografar os dados](#) do cliente.

```
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

Para criar uma coleção de geocercas usando comandos AWS CLI

Use o comando [create-geofence-collection](#).

O exemplo a seguir usa um AWS CLI para criar uma coleção de geofence chamada *ExampleGeofenceCollection*. A coleção de cercas geográficas está associada a uma [AWS KMS chave gerenciada pelo cliente para criptografar os dados](#) do cliente.

```
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

O faturamento depende do seu uso. Você pode incorrer em taxas pelo uso de outros serviços da AWS. Para obter mais informações, consulte os [preços do Amazon Location Service](#).

## Desenhe geocercas

Agora que você criou sua coleção de geocercas, você pode definir suas geocercas. As geocercas são definidas como um polígono ou como um círculo. [Para desenhar uma cerca geográfica poligonal, você pode usar uma ferramenta de JSON edição geográfica, como geojson.io.](#)

Para criar uma geocerca circular, você deve definir o ponto central do círculo e o raio. Por exemplo, se você quiser criar uma geocerca para ser notificado sempre que um dispositivo chegar a 50 metros de um local específico, use a latitude e a longitude desse local e especifique o raio como 50 metros.

Usando o Amazon Location Service APIs, você também pode adicionar metadados à sua cerca geográfica, na forma de pares de valores-chave. Eles podem ser úteis para armazenar informações sobre a geocerca, como seu tipo ou outras informações específicas do seu aplicativo. Você pode usar esses metadados quando [Reagindo aos eventos do Amazon Location Service com a Amazon EventBridge](#).

## Adicionando geocercas poligonais

Esta seção descreve a criação de geocercas poligonais

Desenhe cercas geográficas usando uma ferramenta Geo JSON

[Agora que você criou sua coleção de cercas geográficas, você pode definir suas cercas geográficas usando uma ferramenta de JSON edição geográfica, como \[geojson.io\]\(https://geojson.io\).](#)

Para criar um arquivo Geo JSON

1. Abra uma ferramenta de JSON edição geográfica. Por exemplo, o [geojson.io](https://geojson.io).
2. Selecione o ícone Desenhar um polígono e desenhe sua área de interesse.
3. Escolha Salvar e, em seguida, escolha Geo no JSON menu suspenso.

Coloque Geo JSON geofences em uma coleção de geofence

Você pode usar o JSON arquivo Geo resultante para carregar suas cercas geográficas usando o console do Amazon Location Service, o AWS CLI, ou o Amazon Location: APIs

### Console

Para adicionar uma geocerca a uma coleção de geocercas usando o console do Amazon Location Service

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Coleções de geocercas.
3. Na lista de Coleções de geocercas, selecione o link do nome da coleção de geocercas de destino.
4. Em Geocercas, selecione Criar geocercas.
5. Na janela Adicionar cercas geográficas, arraste e solte seu Geo JSON na janela.
6. Selecione Adicionar geocercas.

### API

Para adicionar cercas geográficas usando a localização da Amazon APIs

Use a [PutGeofence](#) operação da Amazon Location Geofences APIs.

O exemplo a seguir usa uma API solicitação para adicionar uma cerca geográfica com o ID *GEOFENCE-EXAMPLE1* para uma coleção de geofence chamada *ExampleGeofenceCollection*. Ela também especifica uma única propriedade de metadados de delimitação geográfica com a chave e o valor. 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]
      ]
    ]
  }
}
```

Como alternativa, você pode adicionar mais de uma geocerca usando a operação [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]
          ]
        ]
      }
    }
  ]
}
```





Use a [PutGeofence](#) operação da Amazon Location Geofences APIs.

O exemplo a seguir usa uma API solicitação para adicionar uma cerca geográfica com o ID *GEOFENCE-EXAMPLE2* para uma coleção de geofence chamada *ExampleGeofenceCollection*:

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}
```

## AWS CLI

Para adicionar uma cerca geográfica circular a uma coleção de cercas geográficas usando comandos AWS CLI

Use o comando [put-geofence](#).

O exemplo a seguir usa um AWS CLI para adicionar uma cerca geográfica a uma coleção de cerca geográfica chamada *ExampleGeofenceCollection*.

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'
```

### Note

Você também pode inserir JSON geometria complexa em seu próprio arquivo, como no exemplo a seguir.

```
$ aws location \
  put-geofence \
```

```
--collection-name ExampleGeofenceCollection \  
--geofence-id ExampleGeofenceCircle \  
--geometry file:circle.json
```

No exemplo, o arquivo circle.json inclui JSON a geometria do círculo.

```
{  
  "Circle": {  
    "Center": [-74.006975, 40.717127],  
    "Radius": 287.7897969218057  
  }  
}
```

## Inicie o rastreamento

Esta seção orienta você na criação de um aplicativo de rastreamento que captura a localização dos dispositivos.

### Criar um rastreador

Crie um recurso de rastreador para armazenar e processar as atualizações de posição de seus dispositivos. Você pode usar o console do Amazon Location Service AWS CLI, o ou o Amazon LocationAPIs.

Cada atualização de posição armazenada em seus recursos de rastreador pode incluir uma medida da precisão da posição e até três campos de metadados sobre a posição ou o dispositivo que você deseja armazenar. Os metadados são armazenados como pares chave-valor e podem armazenar informações como velocidade, direção, pressão dos pneus ou temperatura do motor.

Os rastreadores filtram as atualizações de posição à medida que elas são recebidas. Isso reduz o ruído visual nos caminhos do dispositivo (chamado de instabilidade) e reduz o número de eventos falsos de entrada e saída da geocerca. Isso também ajuda a gerenciar custos, reduzindo o número de avaliações iniciadas de geocercas.

Os rastreadores oferecem três opções de filtragem de posições para ajudar a gerenciar custos e reduzir a instabilidade nas atualizações de localização.

- Baseado na precisão: use com qualquer dispositivo que forneça uma medição de precisão. A maioria dos dispositivos móveis fornece essas informações. A precisão de cada medição de

posição é afetada por muitos fatores ambientais, incluindo recepção de GPS satélite, paisagem e proximidade de dispositivos Wi-Fi e Bluetooth. A maioria dos dispositivos, incluindo a maioria dos dispositivos móveis, pode fornecer uma estimativa da precisão da medição junto com a medição. Com a filtragem `AccuracyBased`, o Amazon Location ignora atualizações de localização se o dispositivo tiver se movido menos em relação à precisão medida. Por exemplo, se duas atualizações consecutivas de um dispositivo tiverem uma faixa de precisão de 5 m e 10 m, o Amazon Location ignora a segunda atualização se o dispositivo tiver se movido menos de 15 m. O Amazon Location não avalia nem armazena as atualizações ignoradas em relação a geocercas.

Quando a precisão não é fornecida, ela é tratada como zero, a medição é considerada perfeitamente precisa.

#### Note

Você também pode usar a filtragem baseada em precisão para remover toda a filtragem. Se você selecionar a filtragem com base na precisão, mas substituir todos os dados de precisão para zero ou omitir totalmente a precisão, o Amazon Location não filtrará nenhuma atualização.

- Baseada na distância: use quando seus dispositivos não fornecem uma medição precisa, mas você quer aproveitar a filtragem para reduzir a instabilidade e gerenciar os custos mesmo assim. A filtragem `DistanceBased` ignora as atualizações de localização nas quais os dispositivos se moveram menos de 30 m (98,4 pés). Quando você usa a filtragem de posições `DistanceBased`, o Amazon Location não avalia nem armazena essas atualizações ignoradas em relação a geocercas.

A precisão da maioria dos dispositivos móveis, incluindo a precisão média dos dispositivos iOS e Android, é de cerca de 15 m. Na maioria dos aplicativos, a filtragem `DistanceBased` pode reduzir o efeito das imprecisões de localização ao exibir a trajetória do dispositivo em um mapa e o efeito de vai e vem de vários eventos consecutivos de entrada e saída quando os dispositivos estão próximos à borda de uma geocerca. Ela também pode ajudar a reduzir o custo do seu aplicativo, fazendo menos chamadas para avaliar as geocercas vinculadas ou recuperar as posições dos dispositivos.

- Baseada no tempo: (padrão) use quando seus dispositivos enviam atualizações de posição com muita frequência (mais de uma vez a cada 30 segundos) e você deseja obter as avaliações da geocerca em tempo quase real sem armazenar todas as atualizações. Na filtragem `TimeBased`, cada atualização de localização é avaliada em relação a coleções de geocercas vinculadas, mas

nem todas as atualizações de localização são armazenadas. Se a frequência de atualização for superior a 30 segundos, apenas uma atualização a cada 30 segundos será armazenada para cada ID de dispositivo exclusivo.

### Note

Esteja atento aos custos do seu aplicativo de rastreamento ao decidir seu método de filtragem e a frequência das atualizações de posição. Você é cobrado por cada atualização de localização e uma vez para avaliar a atualização de posição em relação a cada coleção de geocercas vinculada. Por exemplo, ao usar a filtragem baseada no tempo, se seu rastreador estiver vinculado a duas coleções de geocercas, cada atualização de posição contará como uma solicitação de atualização de localização e duas avaliações da coleção de geocercas. Se você estiver registrando atualizações de posição a cada cinco segundos para seus dispositivos e usando a filtragem com base no tempo, você será cobrado por 720 atualizações de localização e 1.440 avaliações da geocerca por hora para cada dispositivo. Sua fatura não é afetada pelo número de geocercas em cada coleção. Como cada coleção de geocercas pode conter até 50 mil geocercas, convém combinar suas geocercas em menos coleções, sempre que possível, para reduzir o custo das avaliações das geocercas.

Por padrão, você receberá EventBridge eventos sempre que um dispositivo rastreado entrar ou sair de uma cerca geográfica vinculada. Para obter mais informações, consulte [Vincule um rastreador a uma coleção de geocercas](#).

Você pode ativar eventos para todas as atualizações de posição filtradas para um recurso de rastreador. Para obter mais informações, consulte [Ativar eventos de atualização para um rastreador](#).

### Note

Se você deseja criptografar seus dados usando sua própria chave gerenciada pelo AWS KMS cliente, o recurso Bounding Polygon Queries será desativado por padrão. Isso ocorre porque, ao usar esse recurso Bounding Polygon Queries, uma representação das posições do seu dispositivo não será criptografada usando a chave gerenciada. AWS KMS No entanto, a posição exata do dispositivo é criptografada mesmo assim usando sua chave gerenciada. Você pode optar por ativar o atributo Bounding Polygon Queries definindo o parâmetro `KmsKeyEnableGeospatialQueries` como verdadeiro ao criar ou atualizar um rastreador.

## Console

Para criar um rastreador usando o console do Amazon Location

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Rastreadores.
3. Selecione Criar rastreador.
4. Preencha os seguintes campos:
  - Nome: insira um nome exclusivo. Por exemplo, *ExampleTracker*. Máximo de 100 caracteres. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados.
  - Descrição: Insira uma descrição opcional.
5. Em Filtragem de posições, escolha a opção que melhor se adequa à forma como você pretende usar seu recurso de rastreador. Se você não definir Filtragem de posições, a configuração padrão será TimeBased. Para obter mais informações, consulte [Rastreadores](#) este guia e a Referência [PositionFiltering](#) de rastreadores API do Amazon Location Service.
6. (Opcional) Em Tags, insira uma tag Chave e Valor. Isso adiciona uma tag à sua nova coleção de geocercas. Para obter mais informações, consulte [Marcar seus recursos](#).
7. (Opcional) Em Criptografia de chave gerenciada pelo cliente, você pode selecionar Adicionar uma chave gerenciada pelo cliente. Isso adiciona uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia por meio da criptografia padrão de AWS propriedade. Para obter mais informações, consulte [Criptografia de dados em repouso](#).
8. (Opcional) Em KmsKeyEnableGeospatialQueries, você pode optar por ativar as consultas geoespaciais. Isso permite que você use o recurso Bounding Polygon Queries e, ao mesmo tempo, criptografe seus dados usando uma chave gerenciada pelo cliente. AWS KMS

### Note

Quando você usa o atributo Bounding Polygon Queries, a representação das suas posições dos dispositivos não é criptografada usando a chave gerenciada AWS KMS. No entanto, a posição exata do dispositivo é criptografada mesmo assim usando sua chave gerenciada.

- (Opcional) Em EventBridge configuração, você pode optar por ativar EventBridge eventos para atualizações de posição filtradas. Isso enviará um evento sempre que uma atualização de posição de um dispositivo nesse rastreador atender à avaliação de filtragem de posições.
- Selecione Criar rastreador.

## API

Para criar um rastreador usando a localização da Amazon APIs

Use a [CreateTracker](#) operação dos Amazon Location Trackers APIs.

O exemplo a seguir usa uma API solicitação para criar um rastreador chamado *ExampleTracker*. O recurso rastreador está associado a uma [AWS KMS chave gerenciada pelo cliente para criptografar os dados do cliente](#) e não [permite atualizações de posição](#) em 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"
  }
}
```

Crie um rastreador com **KmsKeyEnableGeospatialQueries** ativado

O exemplo a seguir tem o parâmetro `KmsKeyEnableGeospatialQueries` definido como verdadeiro. Isso permite que você use o recurso Bounding Polygon Queries e, ao mesmo tempo, criptografe seus dados usando uma chave gerenciada pelo cliente. AWS KMS

Para obter informações sobre como usar o atributo Bounding Polygon Queries, consulte [???](#)

**Note**

Quando você usa o recurso Bounding Polygon Queries, uma representação das posições do seu dispositivo não é criptografada usando a chave gerenciada. AWS KMS No entanto, a posição exata do dispositivo é criptografada mesmo assim usando sua chave gerenciada.

```
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

Para criar um rastreador usando comandos AWS CLI

Use o comando [create-tracker](#).

O exemplo a seguir usa o AWS CLI para criar um rastreador chamado *ExampleTracker*. O recurso rastreador está associado a uma [AWS KMS chave gerenciada pelo cliente para criptografar os dados do cliente](#) e não [permite atualizações de posição](#) em. 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"
```



## Crie um rastreador com **KmsKeyEnableGeospatialQueries** ativado

O exemplo a seguir tem o parâmetro `KmsKeyEnableGeospatialQueries` definido como verdadeiro. Isso permite que você use o recurso `Bounding Polygon Queries` e, ao mesmo tempo, criptografe seus dados usando uma chave gerenciada pelo cliente. AWS KMS

Para obter informações sobre como usar o atributo `Bounding Polygon Queries`, consulte [???](#)

### Note

Quando você usa o recurso `Bounding Polygon Queries`, uma representação das posições do seu dispositivo não é criptografada usando a chave gerenciada. AWS KMS No entanto, a posição exata do dispositivo é criptografada mesmo assim usando sua chave gerenciada.

```
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

O faturamento depende do seu uso. Você pode incorrer em taxas pelo uso de outros serviços da AWS . Para obter mais informações, consulte os [preços do Amazon Location Service](#).

Você pode editar a Descrição, a filtragem de posição e a EventBridge configuração após a criação do rastreador, escolhendo Editar rastreador.

## Autenticando suas solicitações

Depois de criar um recurso de rastreador e estiver pronto para começar a avaliar as posições dos dispositivos em relação às geocercas, escolha como autenticar suas solicitações:

- Para explorar as formas para acessar os serviços, consulte [Acessando o Amazon Location Service](#).
- Se você quiser publicar as posições de dispositivos com solicitações não autenticadas, talvez você deva usar o Amazon Cognito.

## Exemplo

O exemplo a seguir mostra o uso de um pool de identidades do Amazon Cognito para autorização, usando a [AWS JavaScript SDKv3](#) e o Amazon Location. [JavaScript Auxiliar de autenticação](#)

```
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);
```

## Atualize seu rastreador com a posição de um dispositivo

Para rastrear seus dispositivos, você pode publicar atualizações das posições dos dispositivos em seu rastreador. Posteriormente, você pode recuperar essas posições dos dispositivos ou o histórico das posições do dispositivo a partir do seu recurso de rastreador.

Cada atualização de posição deve incluir a ID do dispositivo, um carimbo de data e hora e uma posição. Opcionalmente, você pode incluir outros metadados, incluindo a precisão e até três pares chave-valor para seu próprio uso.

Se o seu rastreador estiver vinculado a uma ou mais coleções de geocercas, as atualizações serão avaliadas em relação a essas geocercas (seguindo as regras de filtragem que você especificou para o rastreador). Se um dispositivo violar uma área cercada geograficamente (movendo-se de dentro para fora da área ou vice-versa), você receberá eventos em. EventBridge Esses ENTER ou EXIT eventos incluem os detalhes da atualização da posição, incluindo a ID do dispositivo, o timestamp e quaisquer metadados associados.

### Note

Para obter mais informações sobre a filtragem de posições, consulte [Criar um rastreador](#). Para obter mais informações sobre os eventos das geocercas, consulte [Reagindo aos eventos do Amazon Location Service com a Amazon EventBridge](#).

Use um desses métodos para enviar atualizações do dispositivo:

- [Envie MQTT atualizações](#) para um recurso AWS do IoT Core e vincule-o ao seu recurso rastreador.
- Envie atualizações de localização usando os Amazon Location TrackersAPI, usando o AWS CLI, ou o Amazon Location. APIs Você pode usar o [AWS SDKs](#) para ligar para o a APIs partir do seu aplicativo iOS ou Android.

## API

Para enviar uma atualização de posição usando a localização da Amazon APIs

Use a [BatchUpdateDevicePosition](#) operação dos Amazon Location Trackers APIs.

O exemplo a seguir usa uma API solicitação para publicar uma atualização da posição do dispositivo para *ExampleDevice* para um rastreador *ExampleTracker*.

```
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

Para enviar uma atualização de posição usando comandos AWS CLI

Use o comando [batch-update-device-position](#).

O exemplo a seguir usa um AWS CLI para publicar uma atualização da posição do dispositivo para *ExampleDevice-1* e *ExampleDevice-2* para um rastreador *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"}}]'

```

## Obtenha o histórico de localização de um dispositivo a partir de um rastreador

Seu recurso de rastreador do Amazon Location guarda o histórico de localização de todos os seus dispositivos rastreados por um período de 30 dias. Você pode recuperar o histórico de localização do

dispositivo, incluindo todos os metadados associados, do seu recurso de rastreador. Os exemplos a seguir usam o AWS CLI, ou o Amazon Location APIs.

## API

Para obter o histórico de localização do dispositivo de um rastreador usando o Amazon Location APIs

Use a [GetDevicePositionHistory](#) operação dos Amazon Location Trackers APIs.

O exemplo a seguir usa uma API URI solicitação para obter o histórico de localização do dispositivo de *ExampleDevice* de um rastreador chamado *ExampleTracker* começando de 19:05:07 (inclusive) e termina em 19:20:07 (exclusivo) em 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

Para obter o histórico de localização do dispositivo de um rastreador usando comandos AWS CLI

Use o comando [get-device-position-history](#).

O exemplo a seguir usa um AWS CLI para obter o histórico de localização do dispositivo de *ExampleDevice* de um rastreador chamado *ExampleTracker* começando de 19:05:07 (inclusive) e termina em 19:20:07 (exclusivo) em 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"
```

## Liste suas posições dos dispositivos

Você pode ver uma lista de posições de dispositivos para um rastreador usando o AWS CLI, ou o Amazon Location APIs, com o `ListDevicePositions` API. Quando você liga para o `ListDevicePositions` API, uma lista das posições mais recentes de todos os dispositivos associados a um determinado rastreador é retornada. Por padrão, isso API retorna 100 das posições mais recentes do dispositivo por página de resultados para um determinado rastreador. Para retornar somente os dispositivos que estão dentro de uma região específica, use o parâmetro `FilterGeometry` para criar uma Bounding Polygon Query (solicitação do polígono limítrofe). Dessa forma, quando você ligar `ListDevicePositions`, somente os dispositivos dentro do polígono serão retornados.

### Note

Se você deseja criptografar seus dados usando sua própria chave gerenciada pelo AWS KMS cliente, o recurso Bounding Polygon Queries será desativado por padrão. Isso ocorre porque, ao usar esse recurso, uma representação das posições do seu dispositivo não será criptografada usando a chave AWS KMS gerenciada. A posição exata do dispositivo é criptografada mesmo assim usando sua chave gerenciada.

Você pode optar por ativar o atributo Bounding Polygon Queries. Isso é feito definindo o parâmetro `KmsKeyEnableGeospatialQueries` como verdadeiro ao criar ou atualizar um rastreador.

## API

Use a [ListDevicePositions](#) operação dos Amazon Location Trackers APIs.

O exemplo a seguir é uma API solicitação para obter uma lista das posições do dispositivo na área poligonal, usando o parâmetro opcional. [FilterGeometry](#) O exemplo retorna três localizações de dispositivos presentes na área definida pelo array `Polygon`.

```
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"  
}
```

Este é um exemplo de resposta para [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

Use o comando [list-trackers](#).

O exemplo a seguir é AWS CLI para obter uma lista de dispositivos em uma área poligonal.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

## Vincule um rastreador a uma coleção de geocercas

Agora que você tem uma coleção de geocercas e um rastreador, você pode vinculá-los para que as atualizações de localização sejam avaliadas automaticamente em relação a todas as suas geocercas. Se você não quiser avaliar todas as atualizações de localização ou se você não estiver armazenando algumas de suas localizações em um recurso de rastreador, você pode [avaliar as posições dos dispositivos em relação às geocercas](#) quando precisar.

Quando as posições dos dispositivos são avaliadas em relação às geocercas, os eventos são gerados. Você pode definir uma ação para esses eventos. Para obter mais informações sobre ações que você pode definir para eventos de delimitação geográfica, consulte [Reagir aos eventos do Amazon Location Service com](#) a Amazon. EventBridge

Um evento do Amazon Location inclui os atributos da atualização das posições dos dispositivos que o geram e alguns atributos da geocerca onde se entrou ou saiu. Para obter mais informações sobre os dados incluídos em um evento de geocercas, consulte [Exemplos de EventBridge eventos da Amazon para o Amazon Location Service](#).

Os exemplos a seguir vinculam um recurso de rastreamento a uma coleção de cercas geográficas usando o console, o ou o Amazon AWS CLI Location. APIs

## Console

Para vincular um recurso de rastreador a uma coleção de geocercas usando o console do Amazon Location Service

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Rastreadores.
3. Em Rastreadores de dispositivos, selecione o link do nome do rastreador de destino.
4. Em Coleções de geocercas vinculadas, selecione Vincular coleção de geocercas.
5. Na janela Coleção de geocercas vinculadas, selecione uma coleção de geocercas no menu suspenso.
6. Escolha Vincular.

Depois de vincular o recurso de rastreador, ele receberá um status de Ativo.

## API

Para vincular um recurso rastreador a uma coleção de cercas geográficas usando a localização da Amazon APIs

Use a [AssociateTrackerConsumer](#) operação dos Amazon Location Trackers APIs.

O exemplo a seguir usa uma API solicitação que associa *ExampleTracker* com uma coleção de cercas geográficas usando seu [Amazon Resource Name](#) ()ARN.

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
```

```
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

## AWS CLI

Para vincular um recurso rastreador a uma coleção de cercas geográficas usando comandos AWS CLI

Use o comando [associate-tracker-consumer](#).

O exemplo a seguir usa um AWS CLI para criar uma coleção de geofence chamada *ExampleGeofenceCollection*.

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
    --tracker-name "ExampleTracker"
```

## Avalie as posições dos dispositivos em relação às geocercas

Há duas maneiras de avaliar as posições em relação às geocercas para gerar os eventos das geocercas:

- Você pode vincular os rastreadores e coleções de geocercas. Para obter mais informações, consulte a seção: [Vincule um rastreador a uma coleção de geocercas](#).
- Você pode fazer uma solicitação direta ao recurso de coleta de cerca geográfica para avaliar uma ou mais posições, usando o [BatchEvaluateGeofencesAPI](#)

Além disso, você pode prever eventos de entrada de uma cerca geográfica para um dispositivo entrando, saindo ou permanecendo ocioso dentro de uma cerca geográfica. Use o [ForecastGeofenceEventsAPI](#) para prever eventos.

Se você também quiser rastrear seu histórico de localização do dispositivo ou exibir as localizações em um mapa, vincule o rastreador a uma coleção de geocercas. Por outro lado, talvez você não queira avaliar todas as atualizações de localização ou não pretenda armazenar dados de localização

em um recurso de rastreador. Se algum desses for o caso, você pode fazer uma solicitação direta à coleção de geocercas e avaliar uma ou mais posições dos dispositivos em relação às suas geocercas.

A avaliação das posições dos dispositivos em relação às geocercas gera eventos. Você pode reagir a esses eventos e encaminhá-los para outros AWS serviços. Para obter mais informações sobre as ações que você pode realizar ao receber eventos de delimitação geográfica, consulte [Reagir aos eventos do Amazon Location Service com a Amazon EventBridge](#).

Um evento do Amazon Location inclui os atributos da atualização das posições dos dispositivos que o geram, incluindo o horário, a posição, a precisão e os metadados de chave-valor, além de alguns atributos da geocerca onde se entrou ou saiu. Para obter mais informações sobre os dados incluídos em um evento de geocercas, consulte [Exemplos de EventBridge eventos da Amazon para o Amazon Location Service](#).

Os exemplos a seguir usam o AWS CLI, ou o Amazon Location APIs.

## API

Para avaliar as posições do dispositivo em relação à posição das cercas geográficas usando a localização da Amazon APIs

Use a [BatchEvaluateGeofences](#) operação da Amazon Location Geofences APIs.

O exemplo a seguir usa uma API solicitação para avaliar a posição do dispositivo *ExampleDevice* para uma coleção de cercas geográficas associada *ExampleGeofenceCollection*. Substitua esses valores por sua própria cerca geográfica e dispositivo. 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

Para avaliar as posições do dispositivo em relação à posição das cercas geográficas usando comandos AWS CLI

Use o comando [batch-evaluate-geofences](#).

O exemplo a seguir usa um AWS CLI para avaliar a posição de *ExampleDevice* contra uma coleção de cercas geográficas associada *ExampleGeofenceCollection*. Substitua esses valores por sua própria cerca geográfica e dispositivo. 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"}} ]'
```

A avaliação das posições dos dispositivos em relação às geocercas gera eventos. Tradicionalmente, você pode reagir aos eventos usando [Amazon EventBridge](#), mas esse processo só permite que você reaja aos eventos depois que eles acontecerem. Se você precisar prever quando um dispositivo entra ou sai de uma cerca geográfica, por exemplo, se um dispositivo está cruzando uma fronteira e, como consequência, estará sujeito a uma regulamentação diferente, você poderá usar o [ForecastGeofenceEventsAPI](#) para prever eventos futuros de delimitação geográfica.

[ForecastGeofenceEventsAPI](#) Ele usa critérios como o dispositivo time-to-breach, proximidade, velocidade e posição para prever eventos. API retornará um `ForecastedBreachTime`, que sinaliza o tempo estimado em que o evento de geofence ocorrerá.

O exemplo a seguir usa a localização da Amazon APIs.

## API

Para prever eventos de delimitação geográfica usando a localização da Amazon APIs

Use a [ForecastGeofenceEvents](#) operação da Amazon Location Geofences APIs.

O exemplo a seguir usa uma API solicitação para prever eventos de delimitação geográfica para um *ExampleDevice* em relação a um *ExampleGeofence*. Substitua esses valores por sua própria cerca geográfica e dispositivo. 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
}
```

## Verifique as posições do dispositivo

Para verificar a integridade da posição de um dispositivo, use [VerifyDevicePosition](#) API. Isso API retorna informações sobre a integridade da posição do dispositivo, avaliando propriedades como o sinal da célula do dispositivo, o ponto de acesso Wi-Fi, o endereço Ipv4 e se um proxy está em uso.

### Pré-requisitos

Antes de poder usar a lista APIs para verificação do dispositivo, verifique se você tem os seguintes pré-requisitos:

- Você criou um rastreador para o dispositivo ou dispositivos que deseja verificar. Para obter mais informações, consulte [Inicie o rastreamento](#).

O exemplo a seguir mostra uma solicitação para a localização da Amazon [VerifyDevicePosition](#) API.

### API

Para verificar as posições do dispositivo usando a localização da Amazon APIs

Use a [VerifyDevicePosition](#) operação do Amazon Location Tracking APIs.

O exemplo a seguir mostra uma API solicitação para avaliar a integridade da posição de um dispositivo. Substitua esses valores pelo seu próprio dispositivosIDs.

```
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

O Integrity SDK fornece recursos aprimorados relacionados à verificação de dispositivos e está disponível para uso mediante solicitação. Para ter acesso ao SDK, entre em contato com o [Sales Support](#).

## Reagindo aos eventos do Amazon Location Service com a Amazon EventBridge

EventBridge O Amazon é um barramento de eventos sem servidor que conecta aplicativos de forma eficiente usando dados de AWS serviços como o Amazon Location. EventBridge recebe eventos da Amazon Location e encaminha esses dados para alvos como AWS Lambda. É possível configurar regras de roteamento que determinam o destino dos dados para criar arquiteturas de aplicativos que reagem em tempo real.

Somente eventos de delimitação geográfica (ENTER e EXIT eventos, à medida que os dispositivos entram ou saem das áreas cercadas geograficamente) são enviados por padrão. EventBridge Você também pode ativar todos os eventos de atualização de posição filtrados para um recurso de rastreador. Para ter mais informações, consulte [Ativar eventos de atualização para um rastreador](#).

Para obter mais informações, consulte [Eventos e padrões de eventos](#) no Guia do EventBridge usuário da Amazon.

### Tópicos

- [Ativar eventos de atualização para um rastreador](#)
- [Crie regras de eventos para o Amazon Location](#)
- [Exemplos de EventBridge eventos da Amazon para o Amazon Location Service](#)



## Ativar eventos de atualização para um rastreador

Por padrão, a Amazon Location envia somente eventos de ENTER EXIT delimitação geográfica para. EventBridge Você pode ativar todos os UPDATE eventos de posição filtrados para o qual um rastreador seja enviado. EventBridge Você pode fazer isso ao [criar](#) ou [atualizar](#) um rastreador.

Por exemplo, para atualizar um rastreador existente usando oAWS CLI, você pode usar o comando a seguir (use o nome do seu recurso de rastreador no lugar de *MyTracker*).

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Para desativar os eventos de posição de um rastreador, você deve usar a API ou o console do Amazon Location Service.

## Crie regras de eventos para o Amazon Location

Você pode criar [até 300 regras por barramento de eventos](#) EventBridge para configurar ações tomadas em resposta a um evento de localização da Amazon.

Por exemplo, você pode criar uma regra para eventos de geocercas para os quais uma notificação push será enviada quando um telefone for detectado dentro de um limite de geocerca.

Para criar uma regra para eventos do Amazon Location

Usando os valores a seguir, [crie uma EventBridge regra](#) com base nos eventos de localização da Amazon:

- Em Tipo de regra, escolha Regra com um padrão de evento.
- Na caixa Padrão de evento, adicione o seguinte padrão:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
}
```

Para criar uma regra para atualizações da posição do rastreador, use o padrão a seguir:

```
{
  "source": ["aws.geo"],
```

```
"detail-type": ["Location Device Position Event"]
}
```

Você pode optar por especificar somente os eventos ENTER ou EXIT adicionando uma tag `detail` (se sua regra for para atualizações de posição do rastreador, há apenas uma `EventType`, então não há necessidade de filtrá-la):

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

Você também pode optar por filtrar as propriedades da posição ou da geocerca:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

- Em **Selecionar destinos**, escolha a ação alvo a ser tomada quando um evento for recebido a partir do Amazon Location Service.

Por exemplo, use um tópico do Amazon Simple Notification Service (SNS) para enviar um e-mail ou mensagem de texto quando ocorrer um evento. Para fazer isso, você precisa criar um tópico do Amazon SNS usando o console do Amazon SNS. Para obter mais informações, consulte [Usando o Amazon SNS para notificações ao usuário](#).

**⚠ Warning**

É uma prática recomendada confirmar se a regra do evento foi aplicada com sucesso ou se sua ação automatizada pode não ser iniciada conforme o esperado. Para verificar sua regra de evento, inicie as condições para a regra de evento. Por exemplo, simule um dispositivo entrando em uma área geocercada.

Você também pode capturar todos os eventos do Amazon Location excluindo a seção `detail-type`. Por exemplo: .

```
{
  "source": [
    "aws.geo"
  ]
}
```

**ℹ Note**

O mesmo evento pode ser entregue mais de uma vez. Você pode usar a ID do evento para eliminar a duplicação dos eventos que você recebe.

## Exemplos de EventBridge eventos da Amazon para o Amazon Location Service

Veja a seguir um exemplo de um evento para entrar em uma geocerca, iniciado chamando-se `BatchUpdateDevicePosition`.

```
{
  "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"
  }
}
}
}

```

Veja a seguir um exemplo de um evento para sair de uma geocerca, iniciado chamando-se `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"
}
}
}

```

Veja a seguir um exemplo de um evento para uma atualização de posição, iniciado chamando-se `BatchUpdateDevicePosition`.

```

{
  "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"
    }
  }
}
```

## Rastreamento usando AWS IoT e MQTT com o Amazon Location Service

[MQTT](#) é um protocolo de mensagens leve e amplamente adotado, projetado para dispositivos restritos. AWS IoT Core suporta conexões de dispositivos que usam o MQTT protocolo e MQTT o protocolo WebSocket Secure (WSS).

O [AWS IoT Core](#) conecta dispositivos à AWS e permite que você envie e receba mensagens entre eles. O mecanismo de AWS IoT Core regras armazena consultas sobre os tópicos de mensagens de seus dispositivos e permite que você defina ações para enviar mensagens para outros AWS serviços, como o Amazon Location Service. Dispositivos que estão cientes de sua localização como coordenadas podem ter suas localizações encaminhadas para a Amazon Location por meio do mecanismo de regras.

### Note

Os dispositivos podem conhecer sua própria posição, por exemplo, via embutido GPS. AWS IoT também tem suporte para rastreamento de localização de dispositivos de terceiros. Para obter mais informações, consulte [Local do dispositivo AWS IoT Core](#) no Guia do desenvolvedor do AWS IoT Core.

O passo a passo a seguir descreve o rastreamento usando AWS IoT Core regras. Você também pode enviar as informações do dispositivo para sua própria AWS Lambda função, se precisar processá-las antes de enviar para a Amazon Location. Para obter mais detalhes sobre o uso do Lambda para processar as localizações de seus dispositivos, consulte [Usando AWS Lambda com MQTT](#).

### Tópicos

- [Pré-requisito](#)
- [Crie uma AWS IoT Core regra](#)
- [Teste sua AWS IoT Core regra no console](#)
- [Usando AWS Lambda com MQTT](#)

## Pré-requisito

Antes de começar a rastrear, conclua os seguintes pré-requisitos:

- [Crie um recurso rastreador](#) para o qual você enviará os dados de localização do dispositivo.
- [Crie uma IAM função](#) para conceder AWS IoT Core acesso ao seu rastreador.

Ao seguir essas etapas, use a política a seguir para dar acesso ao seu rastreador:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

## Crie uma AWS IoT Core regra

Em seguida, crie uma AWS IoT Core regra para encaminhar a telemetria posicional de seus dispositivos para o Amazon Location Service. Para obter mais informações sobre como criar regras, consulte os seguintes tópicos no Guia do desenvolvedor do AWS IoT Core :

- [Criação de uma AWS IoT regra](#) para obter informações sobre a criação de uma nova regra.
- [Ação de localização](#) para obter informações específicas sobre a criação de uma regra para publicação no Amazon Location

## Teste sua AWS IoT Core regra no console

Se nenhum dispositivo estiver publicando telemetria que inclua localização no momento, você poderá testar sua regra usando o AWS IoT Core console. O console tem um cliente de teste no qual você pode publicar uma mensagem de amostra para verificar os resultados da solução.

1. Faça login no AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. No painel de navegação à esquerda, expanda Teste e escolha cliente MQTT de teste.
3. Em Publicar em um tópico, defina o nome do tópico como *iot/topic* (ou o nome do tópico que você configurou em sua AWS IoT Core regra, se diferente) e forneça o seguinte para o payload da Mensagem.

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Escolha Publicar no tópico para enviar a mensagem de teste.
5. Para validar se a mensagem foi recebida pelo Amazon Location Service, use o comando AWS CLI a seguir. Se você o modificou durante a configuração, substitua o nome do rastreador pelo que você usou.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

## Usando AWS Lambda com MQTT

Embora o uso não AWS Lambda seja mais necessário ao enviar dados de localização do dispositivo para o Amazon Location para rastreamento, talvez você ainda queira usar o Lambda em alguns casos. Por exemplo, se você mesmo quiser processar os dados de localização do seu dispositivo, antes de enviá-los para o Amazon Location. Os tópicos a seguir descrevem como usar o Lambda para processar as mensagens antes de enviá-las ao rastreador. Para obter mais informações sobre esse padrão, consulte a [arquitetura de referência](#).



## Tópicos

- [Pré-requisito](#)
- [Criar uma função do Lambda](#)
- [Crie uma AWS IoT Core regra](#)
- [Teste sua AWS IoT Core regra no console](#)

### Pré-requisito

Antes de começar a rastrear, você deve [criar um recurso de rastreador](#). Para criar um recurso de rastreamento, você pode usar o console de localização da Amazon AWS CLI, o ou o Amazon LocationAPIs.

O exemplo a seguir usa o console do Amazon Location Service para criar o recurso rastreador:

1. Abra o console do Amazon Location Service em <https://console.aws.amazon.com/location/>.
2. No painel de navegação à esquerda, selecione Rastreadores.
3. Selecione Criar rastreador.
4. Preencha as seguintes caixas:
  - Nome: insira um nome exclusivo com no máximo 100 caracteres. As entradas válidas incluem caracteres alfanuméricos, hífens e sublinhados. Por exemplo, *MyTracker*.
  - Descrição: insira uma descrição opcional. Por exemplo, *Tracker for storing AWS IoT Core device positions*.
  - Filtragem de posição: selecione a filtragem que você deseja usar para atualizações de posição. Por exemplo, Filtragem baseada em precisão.
5. Selecione Criar rastreador.

### Criar uma função do Lambda

Para criar uma conexão entre AWS IoT Core o Amazon Location Service, você precisa de uma AWS Lambda função para processar mensagens encaminhadas por AWS IoT Core. Essa função extrairá todos os dados posicionais, os formatará para o Amazon Location Service e os enviará por meio do Amazon Location API Tracker. Você pode criar essa função por meio do AWS Lambda console ou usar o AWS Command Line Interface (AWS CLI) ou AWS Lambda APIs o.

Para criar uma função do Lambda que publique atualizações de posição no Amazon Location usando o console:

1. Abra o AWS Lambda console em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, selecione Funções.
3. Selecione Criar função e confirme se Autor do zero está selecionado.
4. Preencha as seguintes caixas:
  - Nome da função: insira um nome para a sua função. Entradas válidas incluem caracteres alfanuméricos, hifens, pontos e sublinhados, sem espaços. Por exemplo, *MyLambda*.
  - Tempo de execução — Escolha *Python 3.8*.
5. Selecione Criar função.
6. Selecione a guia Código para abrir o editor.
7. Substitua o código do espaço reservado em `lambda_function.py` pelo seguinte, substituindo o valor atribuído a `TRACKER_NAME` pelo nome do rastreador que você criou como [pré-requisito](#).

```
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. Selecione Implantar para salvar a função atualizada.
9. Escolha a guia Configuração.
10. Na seção Permissões, escolha o nome da Função com hiperlink para conceder permissões do Amazon Location Service à sua função do Lambda.

11. Na página de Resumo do seu perfil, escolha Adicionar permissões e, na lista suspensa, selecione Criar política embutida.
12. Escolha a JSONguia e substitua a política pelo documento a seguir. Isso permite que sua função do Lambda atualize as posições do dispositivo gerenciadas por todos os recursos do rastreador em todas as Regiões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

13. Escolha Revisar política.
14. Insira um nome de política. Por exemplo, *AmazonLocationTrackerWriteOnly*.
15. Escolha Criar política.

Você pode modificar esse código de função, conforme o necessário, para se adaptar ao esquema de mensagens do seu próprio dispositivo.

Crie uma AWS IoT Core regra

Em seguida, crie uma AWS IoT Core regra para encaminhar a telemetria posicional de seus dispositivos para a AWS Lambda função para transformação e publicação no Amazon Location Service. O exemplo de regra fornecido pressupõe que qualquer transformação necessária das cargas úteis do dispositivo seja tratada pela sua função do Lambda. Você pode criar essa regra por meio do AWS IoT Core console, do AWS Command Line Interface (AWS CLI) ou do AWS IoT Core APIs.

**Note**

Embora o AWS IoT console gerencie a permissão necessária AWS IoT Core para permitir a invocação da função Lambda, se você estiver criando sua regra AWS CLI a partir do SDK ou, [deverá configurar uma política para a qual conceder](#) permissão. AWS IoT

Para criar um AWS IoT Core usando o console

1. Faça login no AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. No painel de navegação à esquerda, expanda Agir e Regras.
3. Escolha Criar uma regra para iniciar o novo assistente de regras.
4. Insira um nome e uma descrição para a regra.
5. Para a Declaração de regra de consulta, atualize o atributo FROM para se referir a um tópico em que pelo menos um dispositivo esteja publicando telemetria que inclua a localização. Se você estiver testando a solução, nenhuma modificação será necessária.

```
SELECT * FROM 'iot/topic'
```

6. Em Definir uma ou mais ações, selecione Adicionar ação.
7. Selecione Enviar uma mensagem para uma função do Lambda.
8. Escolha Configurar ação.
9. Encontre e selecione sua função do Lambda na lista.
10. Selecione Adicionar ação.
11. Escolha Criar regra.

Teste sua AWS IoT Core regra no console

Se nenhum dispositivo estiver publicando telemetria que inclua localização no momento, você poderá testar sua regra e essa solução usando o AWS IoT Core console. O console tem um cliente de teste no qual você pode publicar uma mensagem de amostra para verificar os resultados da solução.

1. Faça login no AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. No painel de navegação à esquerda, expanda Teste e escolha cliente MQTT de teste.

3. Em Publicar em um tópico, defina o nome do tópico como *iot/topic* (ou o nome do tópico que você configurou em sua AWS IoT Core regra, se diferente) e forneça o seguinte para o payload da Mensagem. Substitua o carimbo de data/hora *1604940328* com um carimbo de data/hora válido nos últimos 30 dias (qualquer carimbo de data/hora anterior a 30 dias é ignorado).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Escolha Publicar no tópico para enviar a mensagem de teste.
5. Para validar se a mensagem foi recebida pelo Amazon Location Service, use o comando AWS CLI a seguir. Se você o modificou durante a configuração, substitua o nome do rastreador e o ID do dispositivo pelos que você usou.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

## Gerenciando seus recursos de coleta de geocercas

Gerencie suas coleções de geocercas usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Liste seus recursos de coleta de geocercas

Você pode visualizar sua lista de coleta de geocercas usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location:

#### Console

Para ver uma lista de coleções de geocercas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.

3. Veja uma lista de suas coleções de geocercas em Minhas coleções de geocercas.

## API

Use a operação [ListGeofenceCollections](#) das APIs de geocercas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter uma lista de coleções de geocercas na conta AWS.

```
POST /geofencing/v0/list-collections
```

Este é um exemplo de resposta para `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

Use o comando [list-geofence-collections](#).

O exemplo a seguir é uma AWS CLI para obter uma lista de coleções de geocercas na conta da AWS.

```
aws location list-geofence-collections
```

## Obtenha detalhes da coleção de geocercas

Você pode obter detalhes sobre qualquer recurso de coleta de geocercas em sua conta AWS usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location:

### Console

Para visualizar os detalhes de uma coleção de geocercas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.
3. Em Minhas coleções de geocercas, selecione o link do nome da coleção de geocercas de destino.

## API

Use a operação [DescribeGeofenceCollection](#) das APIs de geocercas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter os detalhes da coleção de cercas geográficas para. *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection
```

Este é um exemplo de resposta para 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

Use o comando [describe-geofence-collection](#).

O exemplo a seguir é AWS CLI para obter os detalhes da coleção de cercas geográficas. *ExampleCollection*

```
aws location describe-geofence-collection \
  --collection-name "ExampleCollection"
```



## Excluir uma coleção de geocercas

Você pode excluir uma coleção de geocercas da sua conta AWS usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Console

Para excluir uma coleção de geocercas usando o console do Amazon Location

#### Warning

Essa operação exclui o recurso permanentemente.

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.
3. Em Minha coleção de geocercas, selecione a coleção de geocercas de destino.
4. Escolha Excluir coleção de geocercas.

### API

Use a operação [DeleteGeofenceCollection](#) das APIs do Amazon Location.

O exemplo a seguir é uma solicitação de API para excluir a coleção de cercas geográficas.

*ExampleCollection*

```
DELETE /geofencing/v0/collections/ExampleCollection
```

Este é um exemplo de resposta para DeleteGeofenceCollection:

```
HTTP/1.1 200
```

### CLI

Use o comando [delete-geofence-collection](#).

O exemplo a seguir é um AWS CLI comando para excluir a coleção de cercas geográficas.

*ExampleCollection*

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

## Listar geocercas armazenadas

Você pode listar geocercas armazenadas em uma coleção de geocercas especificada usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Console

Para ver uma lista de geocercas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.
3. Em Minha coleção de geocercas, selecione o link do nome da coleção de geocercas de destino.
4. Veja geocercas na coleção de geocercas em Geocercas

### API

Use a operação [ListGeofences](#) das APIs de geocercas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter uma lista de cercas geográficas armazenadas na coleção de cercas geográficas. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

Este é um exemplo de resposta para 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

Use o comando [list-geofences](#).

O exemplo a seguir é AWS CLI para obter uma lista de cercas geográficas armazenadas na coleção de cercas geográficas. *ExampleCollection*

```
aws location list-geofences \
  --collection-name "ExampleCollection"
```

## Obter detalhes de uma geocerca

Você pode obter os detalhes de uma geocerca específica, como hora de criação, hora de atualização, geometria e status, de uma coleção de geocercas usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

## Console

Para visualizar o status de uma geocerca usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.
3. Em Minha coleção de geocercas, selecione o link do nome da coleção de geocercas de destino.
4. Em Geocercas, você poderá ver o status de suas geocercas.

## API

Use a operação [GetGeofence](#) das APIs de geocercas do Amazon Location.

O exemplo a seguir é uma solicitação de API para obter os detalhes da cerca geográfica de uma coleção de cercas geográficas. *ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

Este é um exemplo de resposta para 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

Use o comando [get-geofence](#).

O exemplo a seguir é AWS CLI para obter os detalhes da coleção de cercas geográficas. *ExampleCollection*

```
aws location get-geofence \
  --collection-name "ExampleCollection" \
  --geofence-id "ExampleGeofence1"
```

## Excluir geocercas

Você pode excluir geocercas de uma coleção de geocercas usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location.

### Console

Para excluir uma geocerca usando o console do Amazon Location

**⚠ Warning**

Essa operação exclui o recurso permanentemente.

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Escolha Coleções de geocercas no painel de navegação à esquerda.
3. Em Minha coleção de geocercas, selecione o link do nome da coleção de geocercas de destino.
4. Em Geocercas, selecione a geocerca de destino.
5. Escolha Excluir geocerca.

**API**

Use a operação [BatchDeleteGeofence](#) das APIs de geocercas do Amazon Location.

O exemplo a seguir é uma solicitação de API para excluir cercas geográficas da coleção de cercas geográficas. *ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

Veja a seguir um exemplo de uma resposta bem-sucedida do [BatchDeleteGeofence](#):

```
HTTP/1.1 200
```

**CLI**

Use o comando [batch-delete-geofence](#).

O exemplo a seguir é um AWS CLI comando para excluir cercas geográficas da coleção de cercas geográficas. *ExampleCollection*

```
aws location batch-delete-geofence \
```

```
--collection-name "ExampleCollection" \  
--geofence-ids "ExampleGeofence11"
```

## Gerenciando seus recursos de rastreador

Você pode gerenciar seus rastreadores usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location.

### Faça uma lista dos seus rastreadores

Você pode visualizar sua lista de rastreadores usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location:

#### Console

Para ver uma lista de rastreadores existentes usando o console Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Rastreadores na barra de navegação à esquerda.
3. Veja uma lista dos seus recursos de rastreador em Meus rastreadores.

#### API

Use a operação [ListTrackers](#) das APIs do Amazon Location Trackers.

O exemplo a seguir é uma solicitação de API para obter uma lista de rastreadores em sua conta AWS.

```
POST /tracking/v0/list-trackers
```

Este é um exemplo de resposta para [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

Use o comando [list-trackers](#).

O exemplo a seguir é um AWS CLI para obter uma lista de rastreadores em sua conta AWS.

```
aws location list-trackers
```

## Desconectando um rastreador de uma coleção de geocercas

Você pode desconectar um rastreador de uma coleção de geocercas usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location:

### Console

Para desassociar um rastreador de uma coleção associada de geocercas usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Rastreadores no painel de navegação à esquerda.
3. Em Meus rastreadores, selecione o link do nome do rastreador de destino.
4. Em Coleções de geocercas vinculadas, selecione uma coleção de geocercas com status Vinculada.
5. Selecione Desvincular.

### API

Use a operação [DisassociateTrackerConsumer](#) das APIs do Amazon Location Trackers.

O exemplo a seguir é uma solicitação de API para desassociar um rastreador de uma coleção de geocercas associada.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

Este é um exemplo de resposta para [DisassociateTrackerConsumer](#):

```
HTTP/1.1 200
```

## CLI

Use o comando [disassociate-tracker-consumer](#).

O exemplo a seguir é um comando AWS CLI para desassociar um rastreador de uma coleção de geocercas associada.

```
aws location disassociate-tracker-consumer \
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/
ExampleCollection" \
  --tracker-name "ExampleTracker"
```

## Obtenha detalhes do rastreador

Você pode obter detalhes sobre qualquer rastreador em sua conta AWS usando o console do Amazon Location, o AWS CLI ou as APIs do Amazon Location.

### Console

Para visualizar os detalhes do rastreador usando o console do Amazon Location

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Rastreadores na barra de navegação à esquerda.
3. Em Meus rastreadores, selecione o link do nome do rastreador de destino.
4. Veja os detalhes do rastreador em Informações.

### API

Use a operação [DescribeTracker](#) das APIs do Amazon Location Tracker.

O exemplo a seguir é uma solicitação de API para obter os detalhes do rastreador.

*ExampleTracker*

```
GET /tracking/v0/trackers/ExampleTracker
```



Este é um exemplo de resposta para [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

Use o comando [describe-tracker](#).

O exemplo a seguir é um AWS CLI comando para obter detalhes do rastreador.

*ExampleTracker*

```
aws location describe-tracker \
  --tracker-name "ExampleTracker"
```

## Exclua um rastreador

Você pode excluir um rastreador da sua conta AWS usando o console de localização da AmazonAWS CLI, o ou as APIs de localização da Amazon:

### Console

Para excluir um recurso de mapa existente usando o console do Amazon Location

#### Warning

Essa operação exclui o recurso permanentemente. Se o recurso do rastreador estiver em uso, você poderá encontrar um erro. Certifique-se de que o recurso de destino não seja uma dependência para seus aplicativos.

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. Selecione Rastreadores no painel de navegação à esquerda.
3. Em Meus rastreadores, selecione o rastreador de destino.
4. Selecione Excluir rastreador.

## API

Use a operação [DeleteTracker](#) das APIs do Amazon Location Tracker.

O exemplo a seguir é uma solicitação de API para excluir o rastreador *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

Este é um exemplo de resposta para [DeleteTracker](#):

```
HTTP/1.1 200
```

## CLI

Use o comando [delete-tracker](#).

O exemplo a seguir é um AWS CLI comando para excluir o rastreador *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

## Exemplo de aplicativo móvel de geofencing and tracking

Este tópico aborda tutoriais criados para demonstrar os principais recursos do uso das cercas geográficas e rastreadores de localização da Amazon em um aplicativo móvel. Os aplicativos demonstram como um rastreador e uma cerca geográfica interagem usando uma combinação dos recursos Lambda AWS IoT e Amazon Location. Há dois tutoriais disponíveis.

- [Exemplo de aplicativo de rastreamento e geofencing para Android, e você pode clonar os arquivos do projeto em: /tree/main/. GitHub \[https://github.com/aws-geospatial/ amazon-location-samples-android\]\(https://github.com/aws-geospatial/amazon-location-samples-android\) \[tracking-with-geofence-notifications\]\(#\)](#)

- [Exemplo de aplicativo de rastreamento e geofencing para iOS, e você pode clonar os arquivos do projeto em: /tree/main/. GitHub <https://github.com/aws-geospatial/amazon-location-samples-ios-tracking-with-geofence-notifications>](https://github.com/aws-geospatial/amazon-location-samples-ios-tracking-with-geofence-notifications)

## Aplicativo de rastreamento e geofence de amostras para Android

Este tópico aborda o tutorial do Android desenvolvido para demonstrar os principais recursos do uso das cercas geográficas e rastreadores de localização da Amazon em um aplicativo móvel. Os aplicativos demonstram como um rastreador e um geofence interagem usando uma combinação dos recursos Lambda AWS IoT e Amazon Location.

### Tópicos

- [Crie recursos de localização da Amazon para seu aplicativo](#)
- [Crie uma coleção Geofence](#)
- [Vincule um rastreador a uma coleção de geocercas](#)
- [Usando o AWS Lambda com o MQTT](#)
- [Configurar o código do aplicativo de amostra](#)
- [Usando o aplicativo de amostra](#)

### Crie recursos de localização da Amazon para seu aplicativo

Para começar, você precisará criar os recursos de localização da Amazon necessários. Esses recursos serão essenciais para a funcionalidade do aplicativo e para a execução dos trechos de código fornecidos.

#### Note

Se você não criou uma AWS conta, siga as instruções no guia do usuário de [gerenciamento de AWS contas](#).

Para começar, você precisará criar um ID de pool de identidade do Amazon Cognito, use o seguinte procedimento:

1. Abra o [console do Amazon Cognito](#) e selecione grupos de identidades no menu do lado esquerdo e, em seguida, selecione Criar grupo de identidades.

2. Verifique se a opção Acesso de convidado está marcada e pressione Avançar para continuar.
3. Em seguida, crie uma nova função do IAM ou use uma função existente do IAM.
4. Insira o nome do Identity Pool e certifique-se de que o Identity Pool tenha acesso aos (geo) recursos de localização da Amazon para o mapa e o rastreador que você criará no próximo procedimento.

Em seguida, você precisa criar e estilizar um mapa no console de localização da AWS Amazon, usando o seguinte procedimento:

1. Navegue até a [seção Mapas](#) do console de localização da Amazon e selecione Criar mapa.
2. Dê um nome e uma descrição ao novo mapa. Registre o nome que você atribuir, conforme ele será usado posteriormente no tutorial.
3. Ao escolher um estilo de mapa, considere o provedor de dados do mapa. Consulte a seção 82 dos [termos de AWS serviço](#) para obter mais detalhes.
4. Aceite os [Termos e Condições de Localização da Amazon](#) e selecione Criar Mapa para concluir o processo de criação do mapa.

Em seguida, você precisa criar um rastreador no console de localização da Amazon, use o seguinte procedimento:

1. Abra a [seção Mapas](#) no console de localização da Amazon.
2. Selecione Criar rastreador.
3. Preencha os campos obrigatórios. Anote o nome do rastreador, pois ele será referenciado ao longo deste tutorial.
4. No campo Filtragem de posição, escolha a opção que melhor se adequa à forma como você pretende usar seu recurso de rastreador. Se você não definir a filtragem de posição, a configuração padrão será. TimeBased Para obter mais informações, consulte [Rastreadores](#) e [PositionFiltering](#) na Referência da API de localização da Amazon.
5. Escolha Criar rastreador para concluir a criação do rastreador.

## Crie uma coleção Geofence

Agora você criará uma coleção de geofence. Você pode usar o console, a API ou a CLI. Os procedimentos a seguir orientam você em cada opção.

- Crie uma coleção de cercas geográficas usando o console de localização da Amazon:
  1. Abra a seção [Geofence Collections](#) do console de localização da Amazon.
  2. Selecione Criar coleção de geocercas.
  3. Forneça um nome e uma descrição para a coleção.
  4. De acordo com a EventBridge regra com Amazon CloudWatch como destino, você pode criar uma EventBridge regra opcional para começar a reagir aos eventos de delimitação geográfica. Isso permite que a Amazon Location publique eventos no Amazon CloudWatch Logs.
  5. Pressione Criar coleção de cercas geográficas para concluir a criação da coleção.
- Crie uma coleção de cercas geográficas usando a API de localização da Amazon:

Use a [CreateGeofenceCollection](#) operação das APIs Amazon Location Geofences. O exemplo a seguir usa uma solicitação de API para criar uma coleção de geofence chamada.

***GEOCOLLECTION\_NAME***

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Crie uma coleção de geofence usando comandos CLI: AWS

Use o comando `create-geofence-collection`. O exemplo a seguir usa uma AWS CLI para criar uma coleção de geofence chamada. *GEOCOLLECTION\_NAME* Para obter mais informações sobre como usar a AWS CLI, consulte a documentação da [interface de linha de AWS comando](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --tags Tag1=Value1
```

## Vincule um rastreador a uma coleção de geocercas

Para vincular um rastreador a uma coleção de cercas geográficas, você pode usar o console, a API ou a CLI. Os procedimentos a seguir orientam você em cada opção.

Vincule um recurso de rastreamento a uma coleção de cercas geográficas usando o console do Amazon Location Service:

1. Abra o console de localização da Amazon.
2. No painel de navegação à esquerda, selecione Rastreadores.
3. Em Rastreadores de dispositivos, selecione o link do nome do rastreador de destino.
4. Em Coleções de geocercas vinculadas, selecione Vincular coleção de geocercas.
5. Na janela Coleção de geocercas vinculadas, selecione uma coleção de geocercas no menu suspenso.
6. Escolha Vincular.
7. Depois de vincular o recurso de rastreador, ele receberá um status de Ativo.

Vincule um recurso rastreador a uma coleção de cercas geográficas usando as APIs de localização da Amazon:

Use a operação `AssociateTrackerConsumer` das APIs do Amazon Location Trackers. O exemplo a seguir usa uma solicitação de API que associa an a uma `ExampleTracker` coleção de geofence usando seu Amazon Resource Name (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
}
```

Vincule um recurso rastreador a uma coleção de cercas geográficas usando comandos CLI AWS :

Use o comando `associate-tracker-consumer` . O exemplo a seguir usa uma AWS CLI para criar uma coleção de geofence chamada. *GOECOLLECTION\_NAME*

```
aws location \  
associate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GOECOLLECTION_NAME" \  
  --tracker-name "ExampleTracker"
```

## Usando o AWS Lambda com o MQTT

Para criar uma conexão entre AWS IoT a Amazon Location, você precisa de uma função Lambda para processar mensagens encaminhadas por eventos. EventBridge CloudWatch Essa função extrairá todos os dados posicionais, os formatará para o Amazon Location e os enviará por meio da API Amazon Location Tracker.

O procedimento a seguir mostra como criar essa função por meio do console Lambda:

1. Abra o [console de](#).
2. No painel de navegação à esquerda, selecione Funções.
3. Em seguida, escolha Criar função e certifique-se de que a opção Autor do zero esteja selecionada.
4. forneça um nome de função e, para a opção Runtime, escolha Node.js 16.x.
5. Escolha a opção Criar função.
6. Abra a guia Código para acessar o editor.
7. Substitua o código do espaço reservado no `index.js` arquivo com o seguinte:

```
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. Pressione Deploy para salvar a função atualizada.
9. Em seguida, abra a guia Configuração.
10. Na seção Acionadores, pressione o botão Adicionar gatilho.
11. Selecione EventBridge (CloudWatch Eventos) no campo Fonte.
12. Selecione a opção Regras existentes.



13. Insira o nome da regra, por exemplo `AmazonLocationMonitor-GEOFENCECOLLECTION_NAME`.
14. Pressione o botão Adicionar.
15. Isso também anexará declarações de políticas baseadas em recursos na guia de permissões

Agora você configurará o cliente de teste MQTT usando AWS IoT, use o seguinte procedimento:

1. Abra o <https://console.aws.amazon.com/iot/>.
2. No painel de navegação esquerdo, selecione o cliente de teste MQTT.
3. Você verá uma seção intitulada Cliente de teste MQTT, na qual você pode configurar sua conexão MQTT.
4. Depois de definir as configurações necessárias, clique no botão Connect para estabelecer uma conexão com o broker MQTT usando os parâmetros fornecidos.
5. Registre o endpoint, conforme usado posteriormente no tutorial.

Uma vez conectado ao cliente de teste, você pode se inscrever nos tópicos do MQTT ou publicar mensagens nos tópicos usando os respectivos campos de entrada fornecidos na interface do cliente de teste do MQTT. Em seguida, você criará uma AWS IoT política.

6. No menu do lado esquerdo, em Gerenciar, expanda a opção Segurança e clique em Políticas.
7. Clique no botão Criar política.
8. Insira um nome de política.
9. Em Documento de política, selecione a guia JSON.
10. Copie e cole a política mostrada abaixo, mas certifique-se de atualizar todos os elementos com seu *REGION* e *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. Selecione o botão Criar para finalizar.

Depois de concluir o procedimento anterior, agora você atualizará as permissões para a função de convidado da seguinte forma:

1. Navegue até o Amazon Cognito e abra seu pool de identidades. Em seguida, vá para o acesso do usuário e selecione a função de convidado.
2. Clique nas políticas de permissão para ativar a edição.

```

{
  '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. Com as mudanças de política acima, todos os AWS recursos necessários agora estão configurados adequadamente para o aplicativo.

### Configurar o código do aplicativo de amostra

1. Clone esse repositório: <https://github.com/aws-geospatial/amazon-location-samples-android/tree/main/tracking-with-geofence-notifications> em sua máquina local.

2. Abra o AmazonSampleSDKApp projeto no Android Studio.
3. Crie e execute o aplicativo em seu dispositivo ou emulador Android.

Usando o aplicativo de amostra

Para usar a amostra, siga estes procedimentos:

- Crie um **custom.properties**:

Para configurar seu custom.properties arquivo, siga estas etapas:

1. Abra seu editor de texto ou IDE preferido.
2. Criar um novo arquivo.
3. Salve o arquivo com o nome custom.properties.
4. Atualize o custom.properties com o exemplo de código a seguir e substitua o MQTT\_END\_POINTPOLICY\_NAME, GEOFENCE\_COLLECTION\_NAME, e TOPIC\_TRACKER pelos nomes dos seus recursos:

```
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. Limpe e reconstrua o projeto. Depois disso, você pode executar o projeto.
- Faça login:

Para fazer login no aplicativo, siga as etapas abaixo:

1. Pressione o botão Entrar.
2. Forneça um ID do grupo de identidades, um nome do rastreador e um nome do mapa.
3. Pressione Entrar novamente para concluir.

- Gerenciar filtros:

Abra a tela de configuração e faça o seguinte:

1. Ative ou desative os filtros usando a interface do switch.
2. Atualize os filtros de tempo e distância quando necessário.

- Operações de rastreamento:

Abra a tela de rastreamento e faça o seguinte:

- Você pode iniciar e interromper o rastreamento em primeiro plano, segundo plano ou no modo de economia de bateria pressionando os respectivos botões.

## Exemplo de aplicativo de rastreamento e geofencing para iOS

Este tópico aborda o tutorial do iOS desenvolvido para demonstrar os principais recursos do uso das cercas geográficas e rastreadores de localização da Amazon em um aplicativo móvel. Os aplicativos demonstram como um rastreador e um geofence interagem usando uma combinação dos recursos Lambda AWS IoT e Amazon Location.

### Tópicos

- [Crie recursos de localização da Amazon para seu aplicativo](#)
- [Crie uma coleção Geofence](#)
- [Vincule um rastreador a uma coleção de geocercas](#)
- [Usando o AWS Lambda com o MQTT](#)
- [Configurando um exemplo de código de aplicativo](#)
- [Usando o aplicativo de amostra](#)

### Crie recursos de localização da Amazon para seu aplicativo

Para começar, você precisará criar os recursos de localização da Amazon necessários. Esses recursos serão essenciais para a funcionalidade do aplicativo e para a execução dos trechos de código fornecidos.

#### Note

Se você não criou uma AWS conta, siga as instruções no guia do usuário de [gerenciamento de AWS contas](#).

Para começar, você precisará criar um ID de pool de identidade do Amazon Cognito, use o seguinte procedimento:

1. Abra o [console do Amazon Cognito](#) e selecione grupos de identidades no menu do lado esquerdo e, em seguida, selecione Criar grupo de identidades.
2. Verifique se a opção Acesso de convidado está marcada e pressione Avançar para continuar.
3. Em seguida, crie uma nova função do IAM ou use uma função existente do IAM.
4. Insira o nome do Identity Pool e certifique-se de que o Identity Pool tenha acesso aos (geo) recursos de localização da Amazon para o mapa e o rastreador que você criará no próximo procedimento.

Em seguida, você precisa criar e estilizar um mapa no console de localização da AWS Amazon, usando o seguinte procedimento:

1. Navegue até a [seção Mapas](#) do console de localização da Amazon e selecione Criar mapa.
2. Dê um nome e uma descrição ao novo mapa. Registre o nome que você atribuir, conforme ele será usado posteriormente no tutorial.
3. Ao escolher um estilo de mapa, considere o provedor de dados do mapa. Consulte a seção 82 dos [termos de AWS serviço](#) para obter mais detalhes.
4. Aceite os [Termos e Condições de Localização da Amazon](#) e selecione Criar Mapa para concluir o processo de criação do mapa.

Em seguida, você precisa criar um rastreador no console de localização da Amazon, use o seguinte procedimento:

1. Abra a [seção Mapas](#) no console de localização da Amazon.
2. Selecione Criar rastreador.
3. Preencha os campos obrigatórios. Anote o nome do rastreador, pois ele será referenciado ao longo deste tutorial.
4. No campo Filtragem de posição, escolha a opção que melhor se adequa à forma como você pretende usar seu recurso de rastreador. Se você não definir a filtragem de posição, a configuração padrão será. TimeBased Para obter mais informações, consulte [Comece a rastrear](#) e [PositionFiltering](#) na Referência da API de localização da Amazon.
5. Escolha Criar rastreador para concluir a criação do rastreador.

## Crie uma coleção Geofence

Agora você criará uma coleção de geofence. Você pode usar o console, a API ou a CLI. Os procedimentos a seguir orientam você em cada opção.

- Crie uma coleção de cercas geográficas usando o console de localização da Amazon:
  1. Abra a seção [Geofence Collections](#) do console de localização da Amazon.
  2. Selecione Criar coleção de geocercas.
  3. Forneça um nome e uma descrição para a coleção.
  4. De acordo com a EventBridge regra com Amazon CloudWatch como destino, você pode criar uma EventBridge regra opcional para começar a reagir aos eventos de delimitação geográfica. Isso permite que a Amazon Location publique eventos no Amazon CloudWatch Logs.
  5. Pressione Criar coleção de cercas geográficas para concluir a criação da coleção.
- Crie uma coleção de cercas geográficas usando a API de localização da Amazon:

Use a [CreateGeofenceCollection](#) operação das APIs Amazon Location Geofences. O exemplo a seguir usa uma solicitação de API para criar uma coleção de geofence chamada.

***GEOCOLLECTION\_NAME***

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "GEOCOLLECTION_NAME",
  "Description": "Geofence collection 1 for shopping center",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

- Crie uma coleção de cercas geográficas usando comandos CLI: AWS

Use o comando `create-geofence-collection`. O exemplo a seguir usa uma AWS CLI para criar uma coleção de geofence chamada. *GEOCOLLECTION\_NAME* Para obter mais informações sobre como usar a AWS CLI, consulte a documentação da [interface de linha de AWS comando](#).

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
```

```
--description "Shopping center geofence collection" \  
--tags Tag1=Value1
```

## Vincule um rastreador a uma coleção de geocercas

Para vincular um rastreador a uma coleção de cercas geográficas, você pode usar o console, a API ou a CLI. Os procedimentos a seguir orientam você em cada opção.

Vincule um recurso de rastreamento a uma coleção de cercas geográficas usando o console do Amazon Location Service:

1. Abra o console de localização da Amazon.
2. No painel de navegação à esquerda, selecione Rastreadores.
3. Em Rastreadores de dispositivos, selecione o link do nome do rastreador de destino.
4. Em Coleções de geocercas vinculadas, selecione Vincular coleção de geocercas.
5. Na janela Coleção de geocercas vinculadas, selecione uma coleção de geocercas no menu suspenso.
6. Escolha Vincular.
7. Depois de vincular o recurso de rastreador, ele receberá um status de Ativo.

Vincule um recurso rastreador a uma coleção de cercas geográficas usando as APIs de localização da Amazon:

Use a operação `AssociateTrackerConsumer` das APIs do Amazon Location Trackers. O exemplo a seguir usa uma solicitação de API `ExampleTracker` associada a uma coleção de cercas geográficas usando seu Amazon Resource Name (ARN).

```
POST /tracking/v0/trackers/ExampleTracker/consumers  
Content-type: application/json  
{  
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-  
collection/GEOCOLLECTION_NAME"  
}
```

Vincule um recurso de rastreador a uma coleção de cercas geográficas usando os comandos da AWS CLI:



Use o comando `associate-tracker-consumer` . O exemplo a seguir usa uma AWS CLI para criar uma coleção de geofence chamada. ***GEOCOLLECTION\_NAME***

```
aws location \
  associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GEOCOLLECTION_NAME" \
    --tracker-name "ExampleTracker"
```

## Usando o AWS Lambda com o MQTT

Para criar uma conexão entre AWS IoT a Amazon Location, você precisa de uma função Lambda para processar mensagens encaminhadas por eventos. EventBridge CloudWatch Essa função extrairá todos os dados posicionais, os formatará para o Amazon Location e os enviará por meio da API Amazon Location Tracker.

O procedimento a seguir mostra como criar essa função por meio do console Lambda:

1. Abra o [console de](#) .
2. No painel de navegação à esquerda, selecione Funções.
3. Em seguida, escolha Criar função e certifique-se de que a opção Autor do zero esteja selecionada.
4. forneça um nome de função e, para a opção Runtime, escolha Node.js 16.x.
5. Escolha a opção Criar função.
6. Abra a guia Código para acessar o editor.
7. Substitua o código do espaço reservado no `index.js` arquivo com o seguinte:

```
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. Pressione Deploy para salvar a função atualizada.
9. Em seguida, abra a guia Configuração.
10. Na seção Acionadores, pressione o botão Adicionar gatilho.
11. Selecione EventBridge (CloudWatch Eventos) no campo Fonte.

12. Selecione a opção Regras existentes.
13. Insira o nome da regra, por exemplo `AmazonLocationMonitor-GEOFENCECOLLECTION_NAME`.
14. Pressione o botão Adicionar.
15. Isso também anexará declarações de políticas baseadas em recursos na guia de permissões

Agora você configurará o cliente de teste AWS IoT MQTT, use o seguinte procedimento:

1. Abra o <https://console.aws.amazon.com/iot/>.
2. No painel de navegação esquerdo, selecione o cliente de teste MQTT.
3. Você verá uma seção intitulada Cliente de teste MQTT, na qual você pode configurar sua conexão MQTT.
4. Depois de definir as configurações necessárias, clique no botão Connect para estabelecer uma conexão com o broker MQTT usando os parâmetros fornecidos.
5. Registre o endpoint, conforme usado posteriormente no tutorial.

Uma vez conectado ao cliente de teste, você pode se inscrever nos tópicos do MQTT ou publicar mensagens nos tópicos usando os respectivos campos de entrada fornecidos na interface do cliente de teste do MQTT. Em seguida, você criará uma AWS IoT política.

6. No menu do lado esquerdo, em Gerenciar, expanda a opção Segurança e clique em Políticas.
7. Clique no botão Criar política.
8. Insira um nome de política.
9. Em Documento de política, selecione a guia JSON.
10. Copie e cole a política mostrada abaixo, mas certifique-se de atualizar todos os elementos com seu *REGION* e *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. Selecione o botão Criar para finalizar.

## Configurando um exemplo de código de aplicativo

Para configurar o código de amostra, você deve ter as seguintes ferramentas instaladas:

- Git
- XCode 15.3 ou posterior
- Simulador iOS 16 ou posterior

Use este procedimento para configurar o código do aplicativo de amostra:

1. [Clone o repositório git a partir desta URL: https://github.com/aws-geospatial/tracking-with-geofence-notifications](https://github.com/aws-geospatial/tracking-with-geofence-notifications)
2. Abra o arquivo de projeto `AWSLocationSampleApp.xcodeproj`.
3. Aguarde a conclusão do processo de resolução do pacote.
4. No menu de navegação do projeto, renomeie `ConfigTemplate.xcconfig` para `Config.xcconfig` e preencha os seguintes valores:

```

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`
```

## Usando o aplicativo de amostra

Depois de configurar o código de amostra, agora você pode executar o aplicativo em um simulador iOS ou em um dispositivo físico.

1. Crie e execute o aplicativo.
2. O aplicativo solicitará permissões de localização e notificação. Você precisa permitir que eles ocorram.
3. Pressione o botão Configuração do Cognito.
4. Salve a configuração.
5. Agora você pode ver as opções de filtro para tempo, distância e precisão. Use-os conforme sua necessidade.
6. Vá para a guia Rastreamento no aplicativo e você verá o mapa e o botão Iniciar rastreamento.
7. Se você instalou o aplicativo em um simulador, talvez queira simular mudanças de localização. Isso pode ser feito em Recursos, na opção do menu Localização. Por exemplo, selecione Características, depois Localização e, em seguida, Freeway Drive.
8. Pressione o botão Iniciar rastreamento. Você deve ver os pontos de rastreamento no mapa.
9. O aplicativo também está rastreando os locais em segundo plano. Portanto, quando você move o aplicativo em segundo plano, ele solicita sua permissão para continuar rastreando no modo de segundo plano.
10. Você pode interromper o rastreamento tocando no botão Parar rastreamento.

## Marcar os recursos do Amazon Location Service

Use a marcação de recursos no Amazon Location para criar tags para categorizar recursos por finalidade, proprietário, ambiente ou critérios. Marcar seus recursos ajuda você a gerenciar, identificar, organizar, pesquisar e filtrar seus recursos.

Por exemplo, com o AWS Resource Groups, é possível criar grupos de recursos da AWS com base em uma ou mais tags ou partes de tags. Também é possível criar grupos baseados em ocorrência em uma pilha do AWS CloudFormation. Usando o Resource Groups e o Tag Editor, é possível

consolidar e visualizar dados de aplicações que consistem em múltiplos serviços, recursos e regiões em um só lugar. Para obter mais informações sobre [estratégias comuns de marcação](#), consulte a Referência geral da AWS.

Cada tag é um rótulo que consiste em uma chave e um valor definidos por você:

- Chave de tag — Um rótulo geral que categoriza os valores da tag. Por exemplo, `CostCenter`.
- Valor da tag — Uma descrição opcional para a categoria da chave da tag. Por exemplo, `MobileAssetTrackingResourcesProd`.

Este tópico ajuda você a começar a marcar por meio da revisão das restrições de marcação. Também mostra como criar tags e usar tags para monitorar o custo da AWS por cada tag ativa usando relatórios de alocação de custos.

## Tópicos

- [Restrições de marcação](#)
- [Conceder permissão para marcar recursos](#)
- [Adicionar uma tag a um recurso do Amazon Location Service](#)
- [Rastreie o custo dos recursos por tag](#)
- [Controlar o acesso a recursos do Amazon Location Service usando tags](#)
- [Saiba mais](#)

## Restrições de marcação

As restrições básicas a seguir se aplicam a tags:

- Máximo de tags por recurso — 50
- Em todos os recursos, cada chave de tag deve ser exclusiva e pode ter apenas um valor.

### Note

Se você adicionar uma nova tag com a mesma chave de uma tag existente, a nova sobrescreverá a tag existente.

- Comprimento máximo da chave — 128 caracteres Unicode em UTF-8
- Comprimento máximo do valor — 256 caracteres Unicode em UTF-8

- Os caracteres permitidos nos serviços são: letras, números e espaços representáveis em UTF-8 e os seguintes caracteres: + - = . \_ : / @.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- O prefixo `aws:` é reservado para uso da AWS. Não é possível editar nem excluir a chave ou o valor de uma tag quando ela tem uma chave de tag com esse prefixo. As tags com o prefixo `aws:` não contam para as tags por limite de recurso.

## Conceder permissão para marcar recursos

Você pode usar as políticas do IAM para controlar o acesso aos seus recursos do Amazon Location e conceder permissão para marcar um recurso na criação. Além de conceder permissão para criar recursos, a política pode incluir permissões `Action` para permitir operações de marcação:

- `geo:TagResource` — Permite que um usuário atribua uma ou mais tags a um recurso do Amazon Location especificado.
- `geo:UntagResource` — Permite que um usuário remova uma ou mais tags de um recurso do Amazon Location especificado.
- `geo:ListTagsForResource` — Permite que um usuário liste todas as tags atribuídas a um recurso do Amazon Location.

Veja a seguir um exemplo de política para permitir que um usuário crie uma coleção de geofences e marque recursos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
      "Effect": "Allow",
      "Action": [
        "geo:CreateGeofenceCollection",
        "geo:TagResource"
      ],
      "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    }
  ]
}
```

## Adicionar uma tag a um recurso do Amazon Location Service

Você pode adicionar tags ao criar seus recursos usando o console do Amazon Location, a AWS CLI, ou as APIs do Amazon Location:

- [Criar um recurso de mapa](#)
- [Criar um recurso de índice de localização](#)
- [Criar um recurso de calculadora de rotas](#)
- [Criar uma coleção de geofences](#)
- [Criar um recurso de rastreador](#)

Para marcar recursos existentes, editar ou excluir tags

1. Abra o console do Amazon Location em <https://console.aws.amazon.com/location/>.
2. No painel de navegação, escolha o recurso o qual você deseja atribuir uma tag. Por exemplo, Mapas.
3. Escolha um recurso na lista.
4. Escolha Gerenciar tags para adicionar, editar ou excluir tags.

## Rastreie o custo dos recursos por tag

Você pode usar tags para alocação de custos para monitorar seus custos da AWS em detalhes. Depois que você ativa as tags de alocação de custos, a AWS usa as tags de alocação de custos para organizar o faturamento do seu recurso em seu relatório de alocação de custos. Isso ajuda você a categorizar e monitorar seus custos de uso.

Há dois tipos de tags de alocação de custos que você pode ativar:

- [Geradas pela AWS](#) — Essas tags são geradas pela AWS. As tags da AWS usam o prefixo `aws:`, por exemplo, `aws:createdBy`.
- [Definido pelo usuário](#) — Essas são tags personalizadas que você cria. As tags definidas pelo usuário usam o prefixo `user:`, por exemplo, `user:CostCenter`.

Você deve ativar cada tipo de tag individualmente. Depois que suas tags forem ativadas, você poderá [ativar AWS Cost Explorer](#) ou visualizar seu relatório mensal de alocação de custos.



## AWS-generated tags

Para ativar tags geradas pela AWS

1. Abra o console do Billing and Cost Management em <https://console.aws.amazon.com/billing/>.
2. No painel de navegação à esquerda, escolha Tags de alocação de custos.
3. Na guia Tags de alocação de custos gerados pela AWS, selecione as chaves de tag que você deseja ativar.
4. Selecione Ativar.

## User-defined tags

Para ativar tags definidas pelo usuário

1. Abra o console do Billing and Cost Management em <https://console.aws.amazon.com/billing/>.
2. No painel de navegação à esquerda, escolha Tags de alocação de custos.
3. Na guia Tags de alocação de custos definidas pelo usuário, selecione as chaves de tag que você deseja ativar.
4. Selecione Ativar.

Depois de ativar suas tags, a AWS gera um [relatório mensal de alocação de custos](#) para o uso e o custo de seus recursos. Esse relatório de alocação de custos inclui todos os seus custos da AWS para cada período de faturamento, incluindo recursos marcados e não marcados. Para obter mais informações, consulte [Usar tags de alocação de custos](#) no Guia do usuário do AWS Billing and Cost Management.

## Controlar o acesso a recursos do Amazon Location Service usando tags

As políticas (IAM) AWS Identity and Access Management oferecem suporte a condições baseadas em etiquetas, o que permite gerenciar a autorização dos recursos com base na chave e nos valores de etiquetas específicos. Por exemplo, uma política de função IAM pode incluir condições para limitar o acesso a ambientes específicos, como desenvolvimento, teste ou produção, com base nas tags.

Para obter mais informações, consulte o tópico sobre [controle de acesso a recursos com base em tags](#).

## Saiba mais

Para obter mais informações:

- Para saber mais sobre as práticas recomendadas para tags, consulte [Marca recursos do AWS](#) na Referência geral do AWS.
- Usando tags para controlar o acesso a recursos da AWS, consulte [Controle do acesso aos recursos da AWS usando tags](#) no Guia do usuário do AWS Identity and Access Management.

## Conceder acesso ao Amazon Location Service

Para usar o Amazon Location Service, um usuário deve ter acesso aos recursos e APIs que compõem o Amazon Location. Há três estratégias que você pode usar para conceder acesso aos seus recursos.

- Use o IAM — Para conceder acesso aos usuários autenticados com AWS IAM Identity Center ou AWS Identity and Access Management (IAM), crie uma política do IAM que permita o acesso aos recursos que você deseja. Para obter mais informações sobre o IAM e o Amazon Location, consulte [Gerenciamento de identidade e acesso do Amazon Location Service](#).
- Use chaves da API: para conceder acesso a usuários não autenticados, você pode criar chaves da API que dão acesso somente de leitura aos seus recursos do Amazon Location Service. Isso é útil caso você não deseje autenticar todos os usuários. Por exemplo, um aplicativo da web. Para obter mais informações sobre chaves da API, consulte [Permitindo o acesso de convidados não autenticados ao seu aplicativo usando chaves de API](#).
- Use o Amazon Cognito: uma alternativa às chaves da API é usar o Amazon Cognito para conceder acesso anônimo. O Amazon Cognito permite que você crie uma autorização mais rica com uma política para definir o que pode ser feito pelos usuários não autenticados. Para obter mais informações sobre como usar o Amazon Cognito, consulte [Permitindo acesso de convidado não autenticado às aplicações usando o Amazon Cognito](#).

### Note

Você também pode usar o Amazon Cognito para usar seu próprio processo de autenticação ou combinar vários métodos de autenticação usando as Identidades Federadas do Amazon Cognito. Para obter mais informações, consulte [Conceitos básicos das identidades federadas](#) no Guia do desenvolvedor do Amazon Cognito.

## Tópicos

- [Permitindo o acesso de convidados não autenticados ao seu aplicativo usando chaves de API](#)
- [Permitindo acesso de convidado não autenticado às aplicações usando o Amazon Cognito.](#)

## Permitindo o acesso de convidados não autenticados ao seu aplicativo usando chaves de API

Quando você chama as APIs do Amazon Location Service em seus aplicativos, você normalmente faz essa chamada como um usuário autenticado que está autorizado a fazer as chamadas de API. Contudo, há alguns casos em que não é necessário autenticar todos os usuários do seu aplicativo. Por exemplo, talvez você queira que um aplicativo da web que mostra a localização da sua empresa esteja disponível para qualquer pessoa que use o site, esteja ela conectada ou não. Nesse caso, uma alternativa é usar as chaves de API para fazer as chamadas de API.

As chaves de API são um valor-chave associado a recursos específicos do Amazon Location Service em você Conta da AWS e a ações específicas que você pode realizar nesses recursos. Você pode usar uma chave de API em seu aplicativo para fazer chamadas não autenticadas para as APIs do Amazon Location para esses recursos. Por exemplo, se você associar uma chave de API ao recurso de mapa MyMap e às ações GetMap\*, um aplicativo que usa essa chave de API poderá visualizar mapas criados com esse recurso e sua conta será cobrada como qualquer outro uso da sua conta. Essa mesma chave de API não daria permissões para alterar ou atualizar o recurso de mapa — somente o uso do recurso é permitido.

### Note

As chaves de API estão disponíveis para uso somente com recursos de mapa, lugar e rota e você não pode modificar ou criar esses recursos. Se seu aplicativo precisar acessar outros recursos ou ações para usuários não autenticados, você poderá usar o Amazon Cognito para fornecer acesso junto com, ou em vez das, chaves de API. Para ter mais informações, consulte [Permitindo acesso de convidado não autenticado às aplicações usando o Amazon Cognito.](#)

As chaves de API incluem um valor de texto não criptografado que dá acesso a um ou mais recursos em seu Conta da AWS. Se alguém copiar sua chave de API, poderá acessar esses mesmos recursos. Para evitar isso, você pode especificar os domínios em que a chave de API pode ser usada

ao criar a chave. Esses domínios são chamados referenciadores. Se necessário, você também pode criar chaves de API de curto prazo definindo prazos de expiração para suas chaves de API.

## Tópicos

- [Chaves de API comparadas ao Amazon Cognito](#)
- [Criar uma chave do KMS](#)
- [Usando uma chave de API para chamar uma API do Amazon Location](#)
- [Usando uma chave de API para renderizar um mapa](#)
- [Gerenciando o tempo de vida útil das chaves de API](#)

## Chaves de API comparadas ao Amazon Cognito

As chaves de API e o Amazon Cognito são usados de forma semelhante em cenários semelhantes, então por que você usaria um em vez do outro? A tabela a seguir destaca algumas das principais diferenças entre os dois.

- As chaves de API estão disponíveis somente para recursos de mapas, lugares e rotas, e somente para determinadas ações. O Amazon Cognito pode ser usado para autenticar o acesso à maioria das APIs do Amazon Location Service.
- Geralmente, o desempenho das solicitações de mapas com chaves de API é mais rápido do que cenários semelhantes com o Amazon Cognito. Uma autenticação mais simples significa menos idas e vindas ao serviço e menos solicitações em cache para se obter o mesmo bloco de mapa novamente em um curto período de tempo.
- Com o Amazon Cognito, você pode usar seu próprio processo de autenticação ou combinar vários métodos de autenticação usando as identidades federadas do Amazon Cognito. Para obter mais informações, consulte [Conceitos básicos das identidades federadas](#) no Guia do desenvolvedor do Amazon Cognito.

## Criar uma chave do KMS

Você pode criar uma chave de API e associá-la a um ou mais recursos em seu Conta da AWS.

Você pode criar uma chave de API usando o console do Amazon Location Service AWS CLI, o ou as APIs Amazon Location.

## Console

Para criar uma chave de API usando o console Amazon Location Service

1. No [console do Amazon Location](#), selecione chaves de API no menu à esquerda.
2. Na página Chaves de API, selecione Criar chave de API.
3. Na página Criar chave de API, preencha as seguintes informações:
  - Nome – um nome para sua chave de API, do tipo MyWebAppKey.
  - Descrição – uma descrição opcional para a sua chave de API.
  - Recursos – escolha os recursos do Amazon Location aos quais conceder acesso com essa chave de API no menu suspenso. Você pode adicionar mais de um recurso escolhendo Adicionar recurso.
  - Ações – especifique as ações que você deseja autorizar com essa chave de API. Você deve selecionar pelo menos uma ação para corresponder a cada tipo de recurso selecionado. Por exemplo, se você selecionou um recurso de local, deverá selecionar pelo menos uma das opções em Ações de locais.
  - Prazo – opcional, adicione uma data e um horário de expiração para sua chave de API. Para ter mais informações, consulte [Gerenciando o tempo de vida útil das chaves de API](#).
  - Referenciadores – opcional, adicione um ou mais domínios nos quais você pode usar a chave de API. Por exemplo, se a chave de API for para permitir que um aplicativo seja executado no site `example.com`, você poderá colocar `*.example.com/` como referenciador permitido.
  - Tags – opcional, adicione tags à chave de API.
4. Selecione Criar chave de API para criar a chave de API.
5. Na página de detalhes da chave de API, você pode ver informações sobre a chave de API que você criou. Selecione Mostrar chave de API para ver o valor da chave que você usa ao chamar as APIs do Amazon Location. O valor da chave terá o formato `v1.public.a1b2c3d4...`. Para obter mais informações sobre como usar a chave de API para renderizar mapas, consulte [Usando uma chave de API para renderizar um mapa](#).

## API

Para criar uma chave de API usando as APIs do Amazon Location

Use a [CreateKey](#) operação das APIs do Amazon Location.

O exemplo a seguir é uma solicitação de API para criar uma chave de API chamada *ExampleKey* sem data de expiração e acesso a um único recurso de mapa.

```
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
}
```

A resposta inclui o valor da chave de API a ser usada ao acessar recursos em seus aplicativos. O valor da chave terá o formato `v1.public.a1b2c3d4...`. Para saber mais sobre como usar a chave de API para renderizar mapas, consulte [Usando uma chave de API para renderizar um mapa](#).

Você também pode usar a [DescribeKey](#) API para encontrar o valor da chave posteriormente.

## AWS CLI

Para criar uma chave de API usando comandos AWS CLI

Use o comando [create-key](#).

O exemplo a seguir cria uma chave de API chamada *ExampleKey* sem data de expiração e acesso a um único recurso de mapa.

```
aws location \
  create-key \
  --key-name ExampleKey \
  --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":
["arn:aws:geo:region:map/mapname"]}' \
  --no-expiry
```

A resposta inclui o valor da chave de API a ser usada ao acessar recursos em seus aplicativos. O valor da chave terá o formato `v1.public.a1b2c3d4...`. Para saber mais sobre como usar a chave de API para renderizar mapas, consulte [Usando uma chave de API para renderizar um mapa](#). A resposta para `create-key` é semelhante à seguinte.

```
{
  "Key": "v1.public.a1b2c3d4...",
  "KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",
  "KeyName": "ExampleKey",
  "CreateTime": "2023-02-06T22:33:15.693Z"
}
```

Você também pode usar a `describe-key` para encontrar o valor da chave posteriormente. O exemplo a seguir mostra como chamar `describe-key` uma chave de API chamada *ExampleKey*.

```
aws location describe-key \
  --key-name ExampleKey
```

## Usando uma chave de API para chamar uma API do Amazon Location

Depois de criar uma chave de API, você pode usar o valor da chave para fazer chamadas para as APIs do Amazon Location em seu aplicativo.

As APIs que oferecem suporte a chaves de API têm um parâmetro adicional que usa o valor da chave de API. Por exemplo, se você chamar a API `GetPlace`, você poderá preencher o parâmetro [chave](#), da seguinte forma

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyValue
```

Se você preencher esse valor, não precisará autenticar a chamada da API com o AWS Sig v4 como faria normalmente.

Para JavaScript desenvolvedores, você pode usar o Amazon Location [JavaScript Auxiliar de autenticação](#) para ajudar na autenticação de operações de API com chaves de API.

Para desenvolvedores móveis, você pode usar os seguintes SDKs de autenticação móvel Amazon Location:

- [SDK de autenticação móvel do Amazon Location Service para iOS](#)
- [SDK de autenticação móvel do Amazon Location Service para Android](#)

Para AWS CLI os usuários, ao usar o `--key` parâmetro, você também deve usar o `--no-sign-request` parâmetro, para evitar assinar com o Sig v4.

#### Note

Se você incluir uma assinatura Sig v4 key e uma assinatura AWS Sig v4 em uma chamada para o Amazon Location Service, somente a chave de API será usada.

## Usando uma chave de API para renderizar um mapa

Você pode usar o valor da chave da API para renderizar um mapa em seu aplicativo usando MapLibre. Isso é um pouco diferente de usar as chaves de API em outras APIs de localização da Amazon que você está chamando diretamente, porque MapLibre faz essas chamadas para você.

O código de exemplo a seguir mostra o uso da chave de API para renderizar um mapa em uma página da Web simples usando o controle de mapa MapLibre GL JS. Para que esse código funcione corretamente, substitua o *v1.public.your-api-key-value*, *us-east-1* e *ExampleMapstrings* com valores que correspondem aos seus. Conta da AWS

```
<!-- 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>
```

## Gerenciando o tempo de vida útil das chaves de API

Você pode criar chaves de API que funcionem indefinidamente. No entanto, se você quiser criar uma chave de API temporária, fizer rodízio das chaves de API regularmente ou revogar uma chave de API existente, você pode usar a expiração da chave de API.

Ao criar uma nova chave de API ou atualizar uma existente, você pode definir o tempo de expiração dessa chave de API.

- Ao atingir o tempo de expiração, a chave de API é desativada automaticamente. As chaves inativas não podem mais ser usadas para fazer solicitações de mapas.
- Você pode excluir uma chave de API 90 dias após desativá-la.
- Se você tiver uma chave inativa que ainda não tenha sido excluída, você poderá restaurá-la atualizando o prazo de expiração para um horário futuro.
- Para criar uma chave permanente, você pode remover o tempo de expiração.
- Se você tentar desativar uma chave de API que tenha sido usada nos últimos 7 dias, você será solicitado a confirmar que deseja fazer a alteração. Se você estiver usando a API do Amazon Location Service, ou a AWS CLI, você receberá um erro, a menos que defina o `ForceUpdate` parâmetro como verdadeiro.

## Permitindo acesso de convidado não autenticado às aplicações usando o Amazon Cognito.

Você pode usar a autenticação do Amazon Cognito como alternativa ao uso direto AWS Identity and Access Management (IAM) com SDKs de front-end e solicitações diretas de HTTPS.

Talvez você queira usar essa forma de autenticação pelos seguintes motivos:

- **Usuários não autenticados** — Se você tiver um site com usuários anônimos, poderá usar os bancos de identidades do Amazon Cognito. Para obter mais informações, consulte a seção em [the section called “Usando Amazon Cognito”](#).
- **Sua própria autenticação** — Se você quiser usar seu próprio processo de autenticação ou combinar vários métodos de autenticação, você pode usar as Identidades Federadas do Amazon Cognito. Para obter mais informações, consulte [Conceitos básicos das identidades federadas](#) no Guia do desenvolvedor do Amazon Cognito.

O Amazon Cognito fornece autenticação, autorização e gerenciamento de usuários para suas aplicações Web e móveis. Você pode usar grupos de identidade não autenticados do Amazon Cognito com o Amazon Location como uma forma de os aplicativos recuperarem credenciais temporárias com escopo reduzido. AWS

Para obter mais informações, consulte [Introdução aos Grupos de Usuários](#) no Guia do Desenvolvedor do Amazon Cognito.

### Note

Para desenvolvedores móveis, o Amazon Location fornece SDKs de autenticação móvel para iOS e Android. Consulte os seguintes repositórios do github para obter mais informações:

- [SDK de autenticação móvel do Amazon Location Service para iOS](#)
- [SDK de autenticação móvel do Amazon Location Service para Android](#)

## Como criar um grupo de identidades do Amazon Cognito

Você pode criar grupos de identidade do Amazon Cognito para permitir o acesso de convidados não autenticados ao seu aplicativo por meio do console do Amazon Cognito, do ou das APIs AWS CLI do Amazon Cognito.

**⚠ Important**

O pool que você cria deve estar na mesma Conta da AWS AWS região dos recursos do Amazon Location Service que você está usando.

Você pode usar políticas do IAM associadas a funções de identidade não autenticadas com as seguintes ações:

- `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`

A inclusão de outras ações do Amazon Location não terá efeito, e identidades não autenticadas não poderão chamá-las.

### Example

Para criar um banco de identidades usando o console do Amazon Cognito

1. Acesse o [console do Amazon Cognito](#).
2. Selecione Gerenciar grupos de identidades.

3. Selecione Criar novo banco de identidades e, em seguida, insira um nome para seu banco de identidades.
4. Na seção expansível Identidades não autenticadas, selecione Habilitar acesso a identidades não autenticadas.
5. Selecione Criar grupo.
6. Escolha quais perfis do IAM você deseja utilizar com seu banco de identidades.
7. Expanda Exibir detalhes.
8. Em Identidades não autenticadas, insira um nome de perfil.
9. Expanda a seção Exibir documento da política e selecione Editar para adicionar sua política.
10. Adicione sua política para conceder acesso aos seus recursos.

Veja a seguir exemplos de políticas para mapas, lugares, rastreadores e rotas. Para utilizar os exemplos na sua própria política, substitua os marcadores de posição *Região* e *ID da conta*:

#### Maps policy example


A política a seguir concede acesso somente para leitura a um recurso de mapa chamado.

#### *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"
    }
  ]
}
```

Adicionar uma [condição do IAM](#) que corresponda a `aws:referer` permite limitar o acesso do navegador aos seus recursos a uma lista de URLs ou prefixos de URL. O exemplo a

seguir permite o acesso a um recurso de mapa chamado `RasterEsriImagery` somente do site `example.com`:

 Warning

Embora `aws:referer` possa limitar o acesso, não é um mecanismo de segurança. É perigoso incluir um valor de cabeçalho do indicador conhecido publicamente. Partes não autorizadas podem usar navegadores personalizados ou modificados para fornecer qualquer valor de `aws:referer` que escolherem. Como resultado, não `aws:referer` deve ser usado para impedir que partes não autorizadas façam AWS solicitações diretas. Ele é oferecido apenas para permitir que os clientes impeçam que seu conteúdo digital, como o conteúdo armazenado no Amazon S3, seja indicado em sites de terceiros não autorizados. Para obter mais informações, consulte [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/*"
          ]
        }
      }
    }
  ]
}
```

Se você estiver [usando o Tangram](#) para exibir um mapa, ele não usa os descritores de estilo, glifos ou sprites retornados pela API do Maps. Em vez disso, ele é configurado apontando para um arquivo `.zip` que contém regras de estilo e ativos necessários. A política a seguir

concede acesso somente para leitura a um recurso de mapa nomeado *ExampleMap* para a GetMapTile operação.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile"
      ],
      "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
    }
  ]
}
```

### Places policy example

A política a seguir concede acesso somente para leitura a um recurso de índice de lugares chamado *ExamplePlaceIndex* para pesquisar lugares por texto ou posições.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PlacesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndex*",
        "geo:GetPlace"
      ],
      "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
    }
  ]
}
```

Adicionar uma [condição do IAM](#) que corresponda a `aws:referer` permite limitar o acesso do navegador aos seus recursos a uma lista de URLs ou prefixos de URL. O exemplo a seguir nega o acesso a um recurso de índice de local chamado *ExamplePlaceIndex* de todos os sites de referência, exceto `example.com`.

**⚠ Warning**

Embora `aws:referer` possa limitar o acesso, não é um mecanismo de segurança. É perigoso incluir um valor de cabeçalho do indicador conhecido publicamente. Partes não autorizadas podem usar navegadores personalizados ou modificados para fornecer qualquer valor de `aws:referer` que escolherem. Como resultado, não `aws:referer` deve ser usado para impedir que partes não autorizadas façam AWS solicitações diretas. Ele é oferecido apenas para permitir que os clientes impeçam que seu conteúdo digital, como o conteúdo armazenado no Amazon S3, seja indicado em sites de terceiros não autorizados. Para obter mais informações, consulte [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**

A política a seguir concede acesso a um recurso rastreador chamado *ExampleTracker* para atualizar as posições do dispositivo.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "UpdateDevicePosition",
    "Effect": "Allow",
    "Action": [
      "geo:BatchUpdateDevicePosition"
    ],
    "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
  }
]
}

```

Adicionar uma [condição do IAM](#) que corresponda a `aws:referer` permite limitar o acesso do navegador aos seus recursos a uma lista de URLs ou prefixos de URL. O exemplo a seguir nega o acesso a um recurso rastreador chamado *ExampleTracker* de todos os sites de referência, exceto. `example.com`

#### Warning

Embora `aws:referer` possa limitar o acesso, não é um mecanismo de segurança. É perigoso incluir um valor de cabeçalho do indicador conhecido publicamente. Partes não autorizadas podem usar navegadores personalizados ou modificados para fornecer qualquer valor de `aws:referer` que escolherem. Como resultado, não `aws:referer` deve ser usado para impedir que partes não autorizadas façam AWS solicitações diretas. Ele é oferecido apenas para permitir que os clientes impeçam que seu conteúdo digital, como o conteúdo armazenado no Amazon S3, seja indicado em sites de terceiros não autorizados. Para obter mais informações, consulte [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

A política a seguir concede acesso a um recurso de calculadora de rotas chamado *ExampleCalculator* para calcular uma rota.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:region:accountID:route-
calculator/ExampleCalculator"
    }
  ]
}

```

Adicionar uma [condição do IAM](#) que corresponda a `aws:referer` permite limitar o acesso do navegador aos seus recursos a uma lista de URLs ou prefixos de URL. O exemplo a seguir nega o acesso a uma calculadora de rotas chamada *ExampleCalculator* de todos os sites de referência, exceto `example.com`.

### Warning

Embora `aws:referer` possa limitar o acesso, não é um mecanismo de segurança. É perigoso incluir um valor de cabeçalho do indicador conhecido publicamente. Partes não autorizadas podem usar navegadores personalizados ou modificados

para fornecer qualquer valor de `aws:referer` que escolherem. Como resultado, não `aws:referer` deve ser usado para impedir que partes não autorizadas façam AWS solicitações diretas. Ele é oferecido apenas para permitir que os clientes impeçam que seu conteúdo digital, como o conteúdo armazenado no Amazon S3, seja indicado em sites de terceiros não autorizados. Para obter mais informações, consulte [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

Embora os grupos de identidade não autenticados sejam destinados à exposição em sites da Internet não seguros, observe que eles serão trocados por credenciais padrão com limite de tempo. AWS

É importante definir corretamente o escopo dos perfis do IAM associados aos bancos de identidades não autenticados.

11. Selecione Permitir para criar seus bancos de identidades.

O banco de identidades resultante segue a sintaxe `<region>:<GUID>`.

Por exemplo: .

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Para obter mais exemplos de políticas específicas para Amazon Location, consulte [the section called “Exemplos de políticas baseadas em identidade”](#).

## Usando os grupos de identidade do Amazon Cognito em JavaScript

O exemplo a seguir troca o pool de identidade não autenticado que você criou por credenciais que são usadas para buscar o descritor de estilo para seu recurso de mapa. *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

As credenciais recuperadas de identidades não autenticadas são válidas por uma hora.

Veja a seguir um exemplo de uma função que renova automaticamente as credenciais antes que elas expirem.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

```
}
```

Para simplificar esse trabalho, você pode usar o Amazon Location [JavaScript Auxiliar de autenticação](#). Isso serve tanto para obter as credenciais quanto para atualizá-las. Este exemplo usa o AWS SDK para JavaScript v3.

```
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));
```

## Próximas etapas

- Para modificar funções, consulte o [console do IAM](#).
- Para gerenciar seus bancos de identidades, acesse o [console do Amazon Cognito](#).

## Monitorar o Amazon Location Service

Ao usar o Amazon Location Service, você pode monitorar seu uso e seus recursos ao longo do tempo usando:

- Amazon CloudWatch. Monitora seus recursos do Amazon Location Service e fornece métricas com estatísticas quase em tempo real.
- AWS CloudTrail. Fornece rastreamento de eventos de todas as chamadas para as APIs do Amazon Location Service.

Esta seção fornece informações sobre esses processos.

### Tópicos

- [Monitorando o Amazon Location Service com a Amazon CloudWatch](#)
- [Registro e monitoramento com o AWS CloudTrail](#)

## Monitorando o Amazon Location Service com a Amazon CloudWatch

A Amazon CloudWatch monitora seus AWS recursos e os aplicativos nos quais você executa quase AWS em tempo real. Você pode monitorar os recursos de localização da Amazon usando CloudWatch, que coleta dados brutos e processa métricas em estatísticas significativas quase em tempo real. Você pode visualizar informações históricas de até 15 meses ou pesquisar métricas para visualizar no CloudWatch console da Amazon para obter mais perspectivas sobre seus recursos de localização na Amazon. É possível definir alarmes para determinados limites e enviar notificações ou realizar ações quando esses limites são atingidos.

Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#)

### Tópicos

- [Métricas do Amazon Location Service exportadas para a Amazon CloudWatch](#)
- [Ver indicadores do Amazon Location Service](#)
- [Crie CloudWatch alarmes para as métricas do Amazon Location Service](#)
- [Usando CloudWatch para monitorar o uso em relação às cotas](#)
- [CloudWatch exemplos de métricas para Amazon Location Service](#)

## Métricas do Amazon Location Service exportadas para a Amazon CloudWatch

As métricas são pontos de dados ordenados por tempo que são exportados para o CloudWatch. Dimensão é um par de nome/valor que identifica exclusivamente um indicador. Para obter mais

informações, consulte [Uso de CloudWatch métricas](#) e [CloudWatch dimensões](#) no Guia CloudWatch do usuário da Amazon.

A seguir estão as métricas para as quais o Amazon Location Service exporta CloudWatch no AWS/Location namespace.

Métrica	Descrição
CallCount	<p>O número de chamadas feitas para um determinado endpoint da API.</p> <p>Dimensões válidas: nomes da API do Amazon Location Service</p> <p>Estatística válida: soma</p> <p>Unidades: contagem</p>
ErrorCount	<p>O número de respostas de erro de chamadas feitas para um determinado endpoint de API.</p> <p>Dimensões válidas: nomes da API do Amazon Location Service</p> <p>Estatística válida: soma</p> <p>Unidades: contagem</p>
SuccessCount	<p>O número de chamadas bem-sucedidas feitas para um determinado endpoint da API.</p> <p>Dimensões válidas: nomes da API do Amazon Location Service</p> <p>Estatística válida: soma</p> <p>Unidades: contagem</p>
CallLatency	<p>A quantidade de tempo que a operação leva para processar e retornar uma resposta quando uma chamada é feita para um determinado endpoint da API.</p> <p>Dimensões válidas: nomes da API do Amazon Location Service</p> <p>Estatística válida: média</p>

Métrica	Descrição
	Unidade: milissegundos

## Ver indicadores do Amazon Location Service

Você pode visualizar os indicadores do Amazon Location Service no console Amazon CloudWatch ou usando a API Amazon CloudWatch.

Para visualizar métricas usando o CloudWatch console

### Example

1. Abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Métricas.
3. Na guia Todos os Indicadores, escolha o namespace Amazon Location.
4. Selecione o tipo de indicador que você deseja visualizar.
5. Selecione um indicador para adicioná-lo ao gráfico.

Para obter mais informações, consulte [Exibir métricas disponíveis](#) no Guia CloudWatch do usuário da Amazon.

## Crie CloudWatch alarmes para as métricas do Amazon Location Service

Você pode usar CloudWatch para definir alarmes em suas métricas do Amazon Location Service. Por exemplo, você pode criar um alarme CloudWatch para enviar um e-mail sempre que ocorrer um pico na contagem de erros.

Os tópicos a seguir fornecem uma visão geral de alto nível sobre como definir alarmes usando o CloudWatch. Para obter instruções detalhadas, consulte Como [usar alarmes](#) no Guia do CloudWatch usuário da Amazon.

Para definir alarmes usando o console CloudWatch

### Example

1. Abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Alarmes.

3. Escolha Create Alarm.
4. Escolha Seleccionar métrica.
5. Na guia Todos os indicadores, selecione o namespace Amazon Location.
6. Selecione uma categoria de indicador.
7. Encontre a linha com o indicador para o qual você deseja criar um alarme e marque a caixa de seleção ao lado dessa linha.
8. Escolha Seleccionar métrica.
9. Em Indicador, preencha os valores.
10. Especifique as Condições do alarme.
11. Escolha Próximo.
12. Se você quiser enviar uma notificação quando as condições do alarme forem atendidas:
  - Em Acionador do estado do alarme, selecione o estado do alarme para solicitar o envio de uma notificação.
  - Em Selecione um tópico do SNS, escolha Criar novo tópico para criar um novo tópico do Amazon Simple Notification Service (Amazon SNS). Insira o nome do tópico e o e-mail para o qual enviar a notificação.
  - Em Enviar uma notificação para inserir endereços de e-mail adicionais para os quais enviar a notificação.
  - Escolha Adicionar notificação. Essa lista é salva e aparece no campo para alarmes futuros.
13. Ao concluir, escolha Next (Avançar).
14. Insira um nome e uma descrição para o alarme e selecione Avançar.
15. Confirme os detalhes do alarme e escolha Avançar.

#### Note

Se você criar um novo tópico do Amazon SNS, os endereços de e-mail deverão ser verificados para que recebam notificações. Se o e-mail não for verificado, a notificação não será recebida quando um alarme for iniciado por uma mudança de estado.

Para obter mais informações sobre como definir alarmes usando o CloudWatch console, consulte [Criar um alarme que envia e-mail](#) no Guia do CloudWatch usuário da Amazon.



## Usando CloudWatch para monitorar o uso em relação às cotas

Você pode criar CloudWatch alarmes da Amazon para notificá-lo quando a utilização de uma determinada cota exceder um limite configurável. Isso permite que você reconheça quando está próximo dos limites de cota e adapte sua utilização para evitar custos excessivos ou solicite um aumento de cota, se necessário. Para obter informações sobre como usar CloudWatch para monitorar cotas, consulte [Visualizando suas cotas de serviço e definindo alarmes no Guia do usuário da Amazon. CloudWatch](#)

## CloudWatch exemplos de métricas para Amazon Location Service

Você pode usar a [GetMetricData](#) API para recuperar métricas para Amazon Location.

- Por exemplo, você pode monitorar `CallCount` e definir um alarme para quando ocorrer uma queda no número.

O monitoramento dos indicadores `sCallCount` para `SendDeviceLocation` pode ajudar a dar uma perspectiva sobre os ativos rastreados. Se a `CallCount` cair, significa que os ativos rastreados, como uma frota de caminhões, pararam de enviar suas localizações atuais. Configurar um alarme para isso pode ajudar a notificá-lo de que um problema ocorreu.

- Em outro exemplo, você pode monitorar `ErrorCount` e definir um alarme para quando ocorrer um aumento no número.

Os rastreadores devem estar associados a coleções de geocercas para que as localizações dos dispositivos sejam avaliadas em relação às geocercas. Se você tem uma frota de dispositivos que exige atualizações contínuas de localização, ver o valor `CallCount` para `BatchEvaluateGeofence` ou `BatchPutDevicePosition` cair para zero indica que as atualizações não estão mais fluindo.

A seguir está um exemplo de saída [GetMetricData](#) com as métricas para `CallCount` e `ErrorCount` para a criação de recursos de mapa.

```
{
  "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"
  }
}
]
}

```

## Registro e monitoramento com o AWS CloudTrail

AWS CloudTrail é um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço. CloudTrail registra todas as chamadas de API como eventos. Você pode usar o Amazon Location Service com CloudTrail para monitorar suas chamadas de API, que incluem

chamadas do console do Amazon Location Service e chamadas de AWS SDK para as operações de API do Amazon Location Service.

Ao criar uma trilha, você pode habilitar a entrega contínua de CloudTrail eventos para um bucket do S3, incluindo eventos para o Amazon Location Service. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Location Service, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para obter mais informações sobre CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

## Tópicos

- [Informações do Amazon Location Service em CloudTrail](#)
- [Entendendo as entradas de arquivo de log do Amazon Location Service](#)

## Informações do Amazon Location Service em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre no Amazon Location Service, essa atividade é registrada em um CloudTrail evento junto com outros eventos do AWS serviço no histórico de eventos. É possível visualizar, pesquisar e baixar os eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para obter um registro contínuo de eventos na sua conta da AWS, incluindo eventos para o Amazon Location Service, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na partição da AWS e fornece os arquivos de log ao bucket do S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros.

Para obter mais informações, consulte:

- [Visão Geral para Criar uma Trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

Todas as ações do Amazon Location Service são registradas CloudTrail e documentadas nas [referências da API do Amazon Location Service](#). Por exemplo, chamadas para o `CreateTracker`, `UpdateTracker` e `DescribeTracker` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar se a solicitação foi feita:

- Com as credenciais de usuário raiz ou AWS Identity and Access Management (IAM).
- Com credenciais de segurança temporárias para uma função ou um usuário federado.
- Por outro serviço da AWS.

Para obter mais informações, consulte [Elemento `userIdentity` do CloudTrail](#).

## Entendendo as entradas de arquivo de log do Amazon Location Service

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do S3 que você especifica ou para o Amazon CloudWatch Logs. Para obter mais informações, consulte Como [trabalhar com arquivos de CloudTrail log](#) no Guia AWS CloudTrail do usuário.

CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a operação solicitada, a data e a hora da operação, os parâmetros de solicitação etc.

### Note

CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica. Para determinar a ordem das operações, use [eventTime](#).

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateTracker` operação, que cria um recurso de rastreador.

```
{
  "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",
}

```

Veja a seguir uma entrada de log da operação DescribeTracker, que retorna os detalhes de um recurso do rastreador.

```
{
  "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",
```

```
}
```

## Criação de recursos Amazon Location Service com AWS CloudFormation

O Amazon Location Service é integrado ao AWS CloudFormation, um serviço que ajuda você a modelar e configurar seus recursos de AWS para que você possa passar menos tempo criando e gerenciando recursos e infraestrutura. Você cria um modelo que descreve todos os recursos de AWS desejados (como os recursos Amazon Location) e o AWS CloudFormation fornece e configura esses recursos para você.

Quando você usa o AWS CloudFormation, você pode reutilizar o modelo para configurar os recursos Amazon Location de modo consistente e repetidamente. Descreva seus recursos uma vez e, depois, provisione os mesmos recursos repetidamente em várias contas e regiões da AWS.

### Modelos Amazon Location e AWS CloudFormation

Para fornecer e configurar recursos para o Amazon Location e serviços relacionados, é preciso entender os [modelos AWS CloudFormation](#). Os modelos são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os atributos que você deseja provisionar nas suas pilhas AWS CloudFormation. Se não estiver familiarizado com JSON ou YAML, você pode usar AWS CloudFormation Designer para ajudá-lo a começar a usar os modelos AWS CloudFormation. Para obter mais informações, consulte [O que é o AWS CloudFormation Designer?](#) no Manual do usuário da AWS CloudFormation.

O Amazon Location é compatível com a criação dos seguintes tipos de recursos no AWS CloudFormation:

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Para obter mais informações, incluindo exemplos de modelos JSON e YAML para os recursos do Amazon Location, consulte [Amazon Location Service resource type reference](#) (Referência do tipo de recurso do Amazon Location Service) no Guia do Usuário AWS CloudFormation.

## Saiba mais sobre a AWS CloudFormation

Para saber mais sobre a AWS CloudFormation, consulte os seguintes recursos:

- [AWS CloudFormation](#)
- [Guia do Usuário AWS CloudFormation](#)
- [Referência de API AWS CloudFormation](#)
- [Guia do Usuário da Interface de Linha de Comando AWS CloudFormation](#)



# Segurança no Amazon Location Service

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Location Service, consulte [AWS Services in Scope by Compliance Program AWS](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon Location. Os tópicos a seguir mostram como configurar o Amazon Location para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de localização na Amazon.

## Tópicos

- [Proteção de dados no Amazon Location Service](#)
- [Gerenciamento de identidade e acesso do Amazon Location Service](#)
- [Resposta a incidentes no Amazon Location Service](#)
- [Validação de compatibilidade do Amazon Location Service](#)
- [Resiliência no Amazon Location Service](#)
- [Segurança da infraestrutura no Amazon Location Service](#)
- [Análise de configuração e vulnerabilidade no Amazon Location](#)
- [Prevenção do problema do substituto confuso entre serviços](#)
- [Melhores práticas de segurança para o Amazon Location Service](#)

- [Melhores práticas do Amazon Location Service](#)

## Proteção de dados no Amazon Location Service

O [modelo de responsabilidade AWS compartilhada](#) de se aplica à proteção de dados no Amazon Location Service. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com a Amazon Location ou outros Serviços da AWS pessoa usando o console, API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou

campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

## Privacidade de dados

Com o Amazon Location Service, você retém o controle dos dados da sua organização. O Amazon Location torna anônimas todas as consultas enviadas aos provedores de dados, removendo metadados do cliente e informações da conta.

O Amazon Location não usa provedores de dados para rastreamento e delimitação geográfica. Isso significa que seus dados confidenciais permanecem na sua AWS conta. Isso ajuda a proteger informações confidenciais de localização, como localização de instalações, ativos e funcionários, de terceiros, protege a privacidade do usuário e reduz o risco de segurança do seu aplicativo.

Para obter informações adicionais, consulte a [Privacidade de AWS dados FAQ](#).

## Retenção de dados no Amazon Location

As seguintes características estão relacionadas à forma como o Amazon Location coleta e armazena dados para o serviço:

- Rastreadores do Amazon Location Service — Quando você usa os Rastreadores APIs para rastrear a localização de entidades, suas coordenadas podem ser armazenadas. As localizações dos dispositivos são armazenadas por 30 dias antes de serem apagadas pelo serviço.
- Amazon Location Service Geofences — Quando você usa as Geofences APIs para definir áreas de interesse, o serviço armazena as geometrias que você forneceu. Elas devem ser excluídas explicitamente.

### Note

Excluindo sua AWS conta, exclua todos os recursos nela contidos. Para obter informações adicionais, consulte a [Privacidade de AWS dados FAQ](#).

## Criptografia de dados em repouso para o Amazon Location Service

O Amazon Location Service fornece criptografia por padrão para proteger dados confidenciais de clientes em repouso usando chaves de criptografia AWS próprias.

- **AWS chaves próprias** — A Amazon Location usa essas chaves por padrão para criptografar automaticamente dados de identificação pessoal. Você não pode visualizar, gerenciar ou usar chaves AWS próprias nem auditar seu uso. No entanto, não é necessário tomar nenhuma medida nem alterar qualquer programa para proteger as chaves que criptografam seus dados. Para obter mais informações, consulte [chaves de propriedade da AWS](#) no Guia do desenvolvedor do AWS Key Management Service .

A criptografia de dados em repouso por padrão reduz a sobrecarga operacional e a complexidade envolvidas na proteção de dados confidenciais. Ao mesmo tempo, ela permite que você crie aplicações seguras que atendam aos rigorosos requisitos regulatórios e de conformidade de criptografia.

Embora você não possa desativar essa camada de criptografia ou selecionar um tipo de criptografia alternativo, você pode adicionar uma segunda camada de criptografia sobre as chaves de criptografia de AWS propriedade existentes escolhendo uma chave gerenciada pelo cliente ao criar seus recursos de rastreamento e coleta de cerca geográfica:

- **Chaves gerenciadas pelo cliente** — O Amazon Location suporta o uso de uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia para adicionar uma segunda camada de criptografia sobre a criptografia existente AWS . Como você tem controle total dessa camada de criptografia, é possível realizar tarefas como:
  - Estabelecer e manter as políticas de chave
  - Estabelecendo e mantendo IAM políticas e subsídios
  - Habilitar e desabilitar políticas de chaves
  - Alternar os materiais de criptografia de chave
  - Adicionar etiquetas
  - Criar réplicas de chaves
  - Programar chaves para exclusão

Para obter mais informações, consulte [chave gerenciada pelo cliente](#) no Guia do desenvolvedor do AWS Key Management Service .

A tabela abaixo resume como o Amazon Location criptografa os dados de identificação pessoal.

Tipo de dados	AWS criptografia de chave própria	Criptografia de chave gerenciada pelo cliente (opcional)
<p><code>Position</code></p> <p>Uma geometria de pontos contendo <a href="#">os detalhes da posição do dispositivo</a>.</p>	Habilitado	Habilitado
<p><code>PositionProperties</code></p> <p>Um conjunto de pares de chave-valor <a href="#">associados à atualização da posição</a>.</p>	Habilitado	Habilitado
<p><code>GeofenceGeometry</code></p> <p>Uma <a href="#">geometria de geocerca</a> poligonal representando a área geocercada.</p>	Habilitado	Habilitado
<p><code>DeviceId</code></p> <p>Identificador do dispositivo especificado ao <a href="#">fazer o upload de uma atualização da posição do dispositivo</a> para um recurso do rastreador.</p>	Habilitado	Sem compatibilidade
<p><code>GeofenceId</code></p> <p>Um identificador especificado ao <a href="#">armazenar uma geometria de geocercas</a> ou um <a href="#">lote de geocercas</a> em uma determinada coleção de geocercas.</p>	Habilitado	Sem compatibilidade

**Note**

O Amazon Location habilita automaticamente a criptografia em repouso usando chaves AWS próprias para proteger dados de identificação pessoal sem nenhum custo.

No entanto, AWS KMS cobranças são aplicadas pelo uso de uma chave gerenciada pelo cliente. Para obter mais informações sobre preços, consulte [definição de preços da AWS Key Management Service](#).

Para obter mais informações sobre AWS KMS, consulte [O que é AWS Key Management Service?](#)

## Como o Amazon Location Service usa subsídios em AWS KMS

O Amazon Location exige uma [concessão](#) para usar sua chave gerenciada pelo cliente.

Quando você cria um [recurso de rastreamento](#) ou uma [coleção de cercas geográficas](#) criptografada com uma chave gerenciada pelo cliente, a Amazon Location cria uma concessão em seu nome enviando uma [CreateGrants](#) solicitação para AWS KMS. As concessões AWS KMS são usadas para dar à Amazon Location acesso a uma KMS chave na conta de um cliente.

O Amazon Location exige a concessão para usar sua chave gerenciada pelo cliente para as seguintes operações internas:

- Envie [DescribeKey](#) solicitações AWS KMS para verificar se o ID simétrico da KMS chave gerenciada pelo cliente inserido ao criar um rastreador ou uma coleção de cercas geográficas é válido.
- Envie [GenerateDataKeyWithoutPlaintext](#) solicitações AWS KMS para gerar chaves de dados criptografadas pela chave gerenciada pelo cliente.
- Envie solicitações de [descriptografia para AWS KMS descriptografar](#) as chaves de dados criptografadas para que elas possam ser usadas para criptografar seus dados.

É possível revogar o acesso à concessão, ou remover o acesso do serviço à chave gerenciada pelo cliente a qualquer momento. Se você fizer isso, o Amazon Location não poderá acessar nenhum dos dados criptografados com a chave gerenciada pelo cliente, o que afetará as operações que dependerem desses dados. Por exemplo, se você tentar [obter as posições de dispositivos](#) a partir de um rastreador criptografado que o Amazon Location não pode acessar, a operação retornará um erro `AccessDeniedException`.

## Criar uma chave gerenciada pelo cliente

Você pode criar uma chave simétrica gerenciada pelo cliente usando o AWS Management Console, ou o. AWS KMS APIs

Para criar uma chave simétrica gerenciada pelo cliente

Siga as etapas de [Criar uma chave simétrica gerenciada pelo cliente](#) no Guia do desenvolvedor do AWS Key Management Service .

### Política de chave

As políticas de chaves controlam o acesso à chave gerenciada pelo seu cliente. Cada chave gerenciada pelo cliente deve ter exatamente uma política de chaves, que contém declarações que determinam quem pode usar a chave e como pode usá-la. Ao criar a chave gerenciada pelo cliente, você pode especificar uma política de chaves. Para obter mais informações, consulte [Managing access to customer managed keys](#) (Administrando o acesso a chaves gerenciadas pelo cliente) no Guia do desenvolvedor do AWS Key Management Service .

Para usar sua chave gerenciada pelo cliente com seus recursos de localização da Amazon, as seguintes API operações devem ser permitidas na política de chaves:

- [kms:CreateGrant](#): Adiciona uma concessão a uma chave gerenciada pelo cliente. Concede acesso de controle a uma KMS chave especificada, o que permite o acesso às [operações de concessão](#) exigidas pela Amazon Location. Para obter mais informações sobre [Utilizar concessões](#), consulte o Guia do desenvolvedor do AWS Key Management Service .

Com isso, o Amazon Location pode:

- Ligar para `GenerateDataKeyWithoutPlainText` para gerar uma chave de dados criptografada e armazená-la, porque a chave de dados não é usada imediatamente para criptografar.
- Ligar para `Decrypt` para usar a chave de dados criptografada armazenada para acessar os dados criptografados.
- Definir uma entidade principal de retirada para permitir o serviço a `RetireGrant`.
- [kms:DescribeKey](#): Fornecer os principais detalhes gerenciados pelo cliente para permitir que o Amazon Location valide a chave.

A seguir estão exemplos de declarações de política que você pode adicionar ao Amazon Location:

```

"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" : "*"
  }
]

```



]

Para obter mais informações sobre [specifying permissions in a policy](#) (especificações de permissões em uma política), consulte o Guia do desenvolvedor do AWS Key Management Service .

Para obter mais informações sobre [solução de problemas de acesso à chave](#), consulte o Guia do Desenvolvedor do AWS Key Management Service .

## Especificação de uma chave gerenciada pelo cliente para o Amazon Location

Você pode especificar uma chave gerenciada pelo cliente para fornecer uma segunda camada de criptografia para os seguintes recursos:

- [Recurso de rastreador](#)
- [Coleção de Geocercas](#)

Ao criar um recurso, você pode especificar a chave de dados inserindo um KMSID, que a Amazon Location usa para criptografar os dados pessoais identificáveis armazenados pelo recurso.

- KMSID — Um [identificador de chave](#) para uma chave gerenciada pelo AWS KMS cliente. Insira um ID de chave, chaveARN, nome de alias ou ARN alias.

## Contexto de criptografia do Amazon Location Service

Um [contexto de criptografia](#) é um conjunto opcional de pares chave-valor que pode conter informações contextuais adicionais sobre os dados.

AWS KMS usa o contexto de criptografia como [dados autenticados adicionais](#) para oferecer suporte à criptografia [autenticada](#). Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, AWS KMS vincula o contexto de criptografia aos dados criptografados. Para descriptografar os dados, você deve incluir o mesmo contexto de criptografia na solicitação.

## Contexto de criptografia do Amazon Location Service

O Amazon Location usa o mesmo contexto de criptografia em todas as operações AWS KMS criptográficas, onde a chave está `aws:geo:arn` e o valor é o [recurso Amazon Resource Name](#) (ARN).

## Example

```
"encryptionContext": {
  "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

### Uso do contexto de criptografia para monitoramento

Ao usar uma chave simétrica gerenciada pelo cliente para criptografar seu tracker (rastreador) ou sua geofence collection (coleção de geocercas), você também pode usar o contexto de criptografia nos registros e logs de auditoria para identificar como a chave gerenciada pelo cliente está sendo usada. O contexto de criptografia também aparece nos [registros gerados pelo AWS CloudTrail ou Amazon CloudWatch Logs](#).

### Uso do contexto de criptografia para controlar o acesso à chave gerenciada pelo cliente

Você pode usar o contexto de criptografia nas principais políticas e IAM políticas conditions para controlar o acesso à sua chave simétrica gerenciada pelo cliente. Você também pode usar restrições no contexto de criptografia em uma concessão.

O Amazon Location utiliza uma restrição ao contexto de criptografia em concessões para controlar o acesso à chave gerenciada pelo cliente na sua conta ou região. A restrição da concessão exige que as operações permitidas pela concessão usem o contexto de criptografia especificado.

## Example

Veja a seguir exemplos de declarações de políticas de chave para conceder acesso a uma chave gerenciada pelo cliente para um contexto de criptografia específico. A condição nesta declaração de política exige que as concessões tenham uma restrição no contexto de criptografia que especifique o contexto da criptografia.

```
{
  "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"
  }
}
}
```

## Monitoramento das suas chaves de criptografia para o Amazon Location Service

Ao usar uma chave gerenciada pelo AWS KMS cliente com seus recursos do Amazon Location Service, você pode usar [AWS CloudTrail](#) [Amazon CloudWatch Logs](#) para rastrear solicitações enviadas para a Amazon Location AWS KMS.

Os exemplos a seguir são AWS CloudTrail eventos para `CreateGrant`, `GenerateDataKeyWithoutPlainTextDecrypt`, e `DescribeKey` para monitorar KMS operações chamadas pela Amazon Location para acessar dados criptografados pela chave gerenciada pelo cliente:

### CreateGrant

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de rastreamento ou coleta geográfica, a Amazon Location envia uma `CreateGrant` solicitação em seu nome para acessar a KMS chave em sua conta. AWS A concessão que o Amazon Location cria é específica para o recurso associado à chave gerenciada pelo cliente AWS KMS . Além disso, o Amazon Location usa a operação `RetireGrant` para remover uma concessão quando você exclui um recurso.

O evento de exemplo a seguir registra a operação `CreateGrant`:

```
{
  "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

Quando você ativa uma chave gerenciada pelo AWS KMS cliente para seu rastreador ou recurso de coleta de cerca geográfica, a Amazon Location cria uma chave de tabela exclusiva. Ele envia uma `GenerateDataKeyWithoutPlainText` solicitação AWS KMS que especifica a chave gerenciada pelo AWS KMS cliente para o recurso.

O evento de exemplo a seguir registra a operação `GenerateDataKeyWithoutPlainText`:

```

{
  "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

Quando você acessa um rastreador ou uma coleção de geocercas criptografados, o Amazon Location chama a operação Decrypt para usar a chave de dados criptografada e armazenada para acessar os dados criptografados.

O evento de exemplo a seguir registra a operação Decrypt:

```

{
  "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

O Amazon Location usa a operação `DescribeKey` para verificar se a chave gerenciada pelo cliente AWS KMS associada ao seu rastreador ou à coleção de geocercas existe na conta e na região.

O evento de exemplo a seguir registra a operação `DescribeKey`:

```

{
  "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"
}
```



## Saiba mais

Os recursos a seguir fornecem mais informações sobre a criptografia de dados em pausa.

- Para obter mais informações sobre [conceitos básicos do AWS Key Management Service](#), consulte o Guia do desenvolvedor do AWS Key Management Service .
- Para obter mais informações sobre [as melhores práticas de segurança para AWS Key Management Service](#), consulte o Guia do AWS Key Management Service desenvolvedor.

## Criptografia de dados em trânsito para o Amazon Location Service

O Amazon Location protege os dados em trânsito, à medida que viajam de e para o serviço, criptografando automaticamente todos os dados entre redes usando o protocolo de criptografia Transport Layer Security (TLS) 1.2. HTTPSAs solicitações diretas enviadas ao Amazon Location Service APIs são assinadas usando o [algoritmo AWS Signature versão 4](#) para estabelecer uma conexão segura.

## Gerenciamento de identidade e acesso do Amazon Location Service

AWS Identity and Access Management (IAM) é uma ferramenta Serviço da AWS que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos de localização da Amazon. IAMé um Serviço da AWS que você pode usar sem custo adicional.

### Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como o Amazon Location Service funciona com IAM](#)
- [Como o Amazon Location Service funciona com usuários não autenticados](#)
- [Exemplos de políticas baseadas em identidade do Amazon Location Service](#)

- [Resolução de problemas de identidade e acesso no Amazon Location Service](#)

## Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz na Amazon Location.

**Usuário do serviço:** se você usar o Amazon Location Service para fazer o seu trabalho, o seu administrador fornecerá a você as credenciais e as permissões necessárias. À medida que você usa mais atributos do Amazon Location para fazer o seu trabalho, é possível que você precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um atributo no Amazon Location, consulte [Resolução de problemas de identidade e acesso no Amazon Location Service](#).

**Administrador do serviço:** se você for o responsável pelos recursos do Amazon Location na sua empresa, você provavelmente terá acesso total ao Amazon Location. Cabe a você determinar quais atributos e recursos do Amazon Location os usuários do seu serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM no Amazon Location, consulte [Como o Amazon Location Service funciona com IAM](#).

**IAM administrador** — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso ao Amazon Location. Para ver exemplos de políticas baseadas em identidade de localização da Amazon que você pode usar no IAM, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#)

## Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte [Como fazer login Conta da AWS no](#) Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no](#) Guia do IAM usuário.

## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um

conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

## Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

## IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Usando IAM funções](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter

informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .

- **Permissões temporárias IAM de IAM usuário** — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas** — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.
- **Acesso entre serviços** — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
- **Sessões de acesso direto (FAS)** — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um Serviço da AWS, combinadas com a solicitação Serviço da AWS para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- **Função de serviço** — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma Serviço da AWS](#) no Guia do IAM usuário.
- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um Serviço da AWS. O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de

propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

## Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

## Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

## Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões** — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.
- **Políticas de controle de serviço (SCPs)** — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.



## Como o Amazon Location Service funciona com IAM

Antes de usar IAM para gerenciar o acesso ao Amazon Location, saiba quais IAM recursos estão disponíveis para uso com o Amazon Location.

IAM recursos que você pode usar com o Amazon Location Service

IAM recurso	Suporte do Amazon Location
<a href="#">Políticas baseadas em identidade</a>	Sim
<a href="#">Políticas baseadas em recursos</a>	Não
<a href="#">Ações das políticas</a>	Sim
<a href="#">Atributos de políticas</a>	Sim
<a href="#">Chaves de condição de política (específicas do serviço)</a>	Sim
<a href="#">ACLs</a>	Não
<a href="#">ABAC(tags nas políticas)</a>	Sim
<a href="#">Credenciais temporárias</a>	Sim
<a href="#">Permissões de entidade principal</a>	Não
<a href="#">Perfis de serviço</a>	Não
<a href="#">Funções vinculadas ao serviço</a>	Não

Para ter uma visão de alto nível de como a Amazon Location e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

### Políticas baseadas em identidade para o Amazon Location

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

## Exemplos de políticas baseadas em identidade para o Amazon Location

Para ver exemplos de políticas baseadas em identidade do Amazon Location, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#).

## Políticas baseadas em recursos no Amazon Location

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a

uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no](#) Guia do IAM usuário.

## Ações de políticas para o Amazon Location

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente com permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do Amazon Location, consulte [Ações definidas pelo Amazon Location Service](#) em Service Authorization Reference (Referência de autorização do serviço).

As ações de política no Amazon Location usam o seguinte prefixo antes da ação:

```
geo
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
  "geo:action1",  
  "geo:action2"  
]
```

Você também pode especificar várias ações usando caracteres-curinga (\*). Por exemplo, para especificar todas as ações que começam com a palavra `Get`, inclua a seguinte ação:

```
"Action": "geo:Get*"
```

Para ver exemplos de políticas baseadas em identidade do Amazon Location, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#).

## Recursos de políticas para o Amazon Location

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*" 
```

Para ver uma lista dos tipos de recursos de localização da Amazon e seus ARNs, consulte [Resources Defined by Amazon Location Service](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Actions Defined by Amazon Location Service](#).

ARN

Para ver exemplos de políticas baseadas em identidade do Amazon Location, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#).

## Chaves de condição de políticas para o Amazon Location

Compatível com chaves de condição de política específicas de serviço: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões

condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

Para ver uma lista de chaves de condição do Amazon Location, consulte [Condition Keys for Amazon Location Service](#) (Chaves de condição do Amazon Location Service) em Service Authorization Reference (Referência de autorização do serviço). Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Actions Defined by Amazon Location Service](#) (Ações definidas pelo Amazon Location Service).

O Amazon Location é compatível com chaves de condição para permitir ou negar o acesso a geocercas ou dispositivos específicos em suas declarações de política. As seguintes chaves de condição estão disponíveis:

- `geo:GeofenceIds` para uso com ações de Geocercas. O tipo é `ArrayOfString`.
- `geo:DeviceIds` para uso com ações do Rastreador. O tipo é `ArrayOfString`.

As seguintes ações podem ser usadas `geo:GeofenceIds` em sua IAM política:

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

As seguintes ações podem ser usadas `geo:DeviceIds` em sua IAM política:

- `BatchDeleteDevicePositionHistory`
- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

#### Note

Você não pode usar essas chaves de condição com as ações `BatchEvaluateGeofences`, `ListGeofences`, ou `ListDevicePosition`.

Para ver exemplos de políticas baseadas em identidade do Amazon Location, consulte [Exemplos de políticas baseadas em identidade do Amazon Location Service](#).

## ACLs na localização da Amazon

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

## ABAC com Amazon Location

Suportes ABAC (tags nas políticas): Sim

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre ABAC, consulte [O que é ABAC?](#) no Guia do IAM usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

Para ter mais informações sobre recursos de marcação do Amazon Location, consulte [Marcar os recursos do Amazon Location Service](#).

Para visualizar um exemplo de política baseada em identidade para limitar o acesso a um recurso baseado em tags desse recurso, consulte [Controlar o acesso a recursos com base em tags](#).

## Uso de credenciais temporárias com o Amazon Location

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS esse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternando para uma função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

## Permissões de entidades principais entre serviços para o Amazon Location

Suporta sessões de acesso direto (FAS): Não

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um Serviço da AWS, combinadas com a solicitação Serviço da AWS para fazer solicitações aos serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

## Perfis de serviço para o Amazon Location

Compatível com perfis de serviço: não

Uma função de serviço é uma [IAMfunção](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma Serviço da AWS](#) no Guia do IAM usuário.

### Warning

A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon Location. Edite perfis de serviço somente quando o Amazon Location fornecer orientação para isso.

## Perfis vinculados ao serviço para o Amazon Location

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. Serviço da AWS O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte [AWS serviços que funcionam](#) com. IAM Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.



## Como o Amazon Location Service funciona com usuários não autenticados

Muitos cenários para usar o Amazon Location Service, incluindo a exibição de mapas na web ou em um aplicativo móvel, exigem a permissão de acesso a usuários que não tenham feito login com IAM. Nesses casos não autenticados, você tem duas opções.

- Use API chaves — Para conceder acesso a usuários não autenticados, você pode criar API chaves que dão acesso somente de leitura aos seus recursos do Amazon Location Service. Isso é útil caso você não deseje autenticar todos os usuários. Por exemplo, um aplicativo da web. Para obter mais informações sobre API chaves, consulte [Permitindo o acesso de convidados não autenticados ao seu aplicativo usando chaves de API](#).
- Use o Amazon Cognito — Uma alternativa às API chaves é usar o Amazon Cognito para conceder acesso anônimo. O Amazon Cognito permite que você crie uma autorização mais rica com uma IAM política para definir o que pode ser feito pelos usuários não autenticados. Para obter mais informações sobre como usar o Amazon Cognito, consulte [Permitindo acesso de convidado não autenticado às aplicações usando o Amazon Cognito](#).

Para obter uma visão geral sobre como fornecer acesso a usuários não autenticados, consulte [Conceder acesso ao Amazon Location Service](#).

## Exemplos de políticas baseadas em identidade do Amazon Location Service

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amazon Location. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criação de IAM políticas no Guia](#) do IAM usuário.

Para obter detalhes sobre ações e tipos de recursos definidos pela Amazon Location, incluindo o formato de cada um dos ARNs tipos de recursos, consulte [Actions, Resources, and Condition Keys for Amazon Location Service](#) na Referência de Autorização de Serviço.

### Tópicos

- [Melhores práticas de políticas](#)
- [Usar o console do Amazon Location](#)
- [Permitir que usuários visualizem suas próprias permissões](#)
- [Usar os recursos do Amazon Location Service em políticas](#)
- [Permissões para atualizar as posições do dispositivo](#)
- [Política somente leitura para recursos do rastreador](#)
- [Política para a criação de geocercas](#)
- [Política somente leitura para geocercas](#)
- [Permissões para renderizar um recurso de mapa](#)
- [Permissões para operações de pesquisa](#)
- [Política somente leitura para calculadoras de rotas](#)
- [Controlar o acesso aos recursos com base nas chaves de condições](#)
- [Controlar o acesso a recursos com base em tags](#)

## Melhores práticas de políticas

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon Location em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte [políticas AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.
- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode

escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica Serviço da AWS, como AWS CloudFormation. Para obter mais informações, consulte [elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.

- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação da política do IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [Configurando o API acesso MFA protegido](#) no Guia do IAM usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

## Usar o console do Amazon Location

Para acessar o console do Amazon Location Service, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos de localização da Amazon em seu Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para garantir que usuários e perfis ainda possam usar o console do Amazon Location, anexe também a política para as entidades. Para obter mais informações, consulte [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

A política a seguir dá acesso ao console do Amazon Location Service para poder criar, excluir, listar e visualizar detalhes sobre os recursos de localização do Amazon Location em sua conta AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Alternativamente, você pode conceder permissões de somente leitura para facilitar o acesso como somente leitura. Com permissões somente leitura, uma mensagem de erro aparecerá se o usuário tentar escrever ações, como criar ou excluir recursos. Como exemplo, consulte [the section called “Política somente leitura para rastreadores”](#)

## Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou AWS API

```
{
  "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": "*"
}
]
}

```

## Usar os recursos do Amazon Location Service em políticas

O Amazon Location Service usa os seguintes prefixos para recursos:

Prefixo do recurso do Amazon Location

Recurso	Prefixo do recurso
Recursos de mapa	map
Recursos de local	place-index
Recursos de rota	route-calculator
Recursos de rastreamento	tracker
Recursos Geofence Collection (Coleção de Geocercas)	geofence-collection

Use a seguinte ARN sintaxe:

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

Para obter mais informações sobre o formato de ARNs, consulte [Amazon Resource Names \(ARNs\) e AWS Service Namespaces](#).

## Exemplos

- Use o seguinte ARN para permitir o acesso a um recurso de mapa especificado.

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
```

- Para especificar o acesso a todos os recursos da map que pertencem a uma conta específica, use o caractere curinga (\*):

```
"Resource": "arn:aws:geo:us-west-2:account-id:map/*"
```

- Algumas ações do Amazon Location, como as que servem para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (\*).

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos de localização da Amazon e seus ARNs, consulte [Resources Defined by Amazon Location Service](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Actions Defined by Amazon Location Service](#).

## ARN

### Permissões para atualizar as posições do dispositivo

Para atualizar as posições do dispositivo para vários rastreadores, você deve conceder a um usuário o acesso a um ou mais dos seus recursos de rastreador. Você também deverá permitir que o usuário atualize um lote de posições do dispositivo.

Neste exemplo, além de conceder acesso ao *Tracker1* e *Tracker2* recursos, a política a seguir concede permissão para usar a `geo:BatchUpdateDevicePosition` ação contra o *Tracker1* e *Tracker2* recursos.

```
{
  "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"
    ]
  }
]
}

```

Se você quiser limitar o usuário a poder atualizar somente as posições de um dispositivo específico, você pode adicionar uma chave de condição para a ID desse dispositivo.

```

{
  "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"
          ]
        }
      }
    }
  ]
}

```

## Política somente leitura para recursos do rastreador

Para criar uma política somente de leitura para todos os recursos do rastreador em sua AWS conta, você precisará conceder acesso a todos os recursos do rastreador. Você também deve conceder ao

usuário o acesso a ações que permitam obter a posição do dispositivo para vários dispositivos, obter a posição do dispositivo a partir de um único dispositivo e obter o histórico de posições.

Neste exemplo, a política a seguir concede permissão para as seguintes ações:

- `geo:BatchGetDevicePosition` para recuperar as posições de vários dispositivos.
- `geo:GetDevicePosition` para recuperar a posição de um único dispositivo.
- `geo:GetDevicePositionHistory` para recuperar o histórico de posições de um dispositivo.

```
{
  "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/*"
    }
  ]
}
```

## Política para a criação de geocercas

Para criar uma política que permita a um usuário criar geocercas, você precisará conceder acesso a ações específicas que permitam aos usuários criar uma ou mais geocercas em uma coleção de geocercas.

A política abaixo concede permissão para as seguintes ações em *Collection*:

- `geo:BatchPutGeofence` para criar várias geocercas.
- `geo:PutGeofence` para criar uma única geocerca.

```
{
  "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"
}
```

## Política somente leitura para geocercas

Para criar uma política somente leitura para geocercas armazenadas em uma coleção de geocercas na sua conta AWS , você precisará conceder acesso às ações que são lidas da coleção de geocercas que armazena as geocercas.

A política abaixo concede permissão para as seguintes ações em *Collection*:

- `geo:ListGeofences` para listar geocercas na coleção de geocercas especificada.
- `geo:GetGeofence` para recuperar uma geocerca da coleção de geocercas.

```
{
  "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"
    }
  ]
}
```

## Permissões para renderizar um recurso de mapa

Para conceder permissões suficientes para renderizar mapas, você precisará conceder acesso aos blocos de mapas, “sprites”, glifos e ao descritor de estilo:

- o `geo:GetMapTile` recupera blocos de mapas usados para renderizar seletivamente os atributos em um mapa.
- `geo:GetMapSprites` recupera a planilha PNG sprite e o JSON documento correspondente descrevendo os deslocamentos dentro dela.
- o `geo:GetMapGlyphs` recupera os glifos usados para exibir texto.
- o `geo:GetMapStyleDescriptor` recupera o descritor de estilo do mapa, contendo regras de renderização.

```
{
  "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"
    }
  ]
}
```

## Permissões para operações de pesquisa

Para criar uma política que permita operações de pesquisa, primeiro você precisará conceder acesso ao recurso de índice de lugares em sua AWS conta. Você também deverá conceder acesso a ações que permitam ao usuário pesquisar usando texto por geocodificação e pesquisar usando uma posição por geocodificação reversa.

Neste exemplo, além de conceder acesso a *PlaceIndex*, a política a seguir também concede permissão para as seguintes ações:

- o `geo:SearchPlaceIndexForPosition` permite que você pesquise lugares ou pontos de interesse próximos a uma determinada posição.
- o `geo:SearchPlaceIndexForText` permite que você pesquise um endereço, nome, cidade ou região usando texto em formato livre.

```
{
  "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"
    }
  ]
}
```

## Política somente leitura para calculadoras de rotas

Você pode criar uma política somente leitura para permitir que um usuário acesse um recurso de calculadora de rotas para calcular uma rota.

Neste exemplo, além de conceder acesso a *ExampleCalculator*, a política a seguir concede permissão para a seguinte operação:

- o `geo:CalculateRoute` calcula uma rota com base em uma posição de partida, posição de destino e uma lista de posições de pontos de referência.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
    }
  ]
}
```

```

    "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
  }
]
}

```

## Controlar o acesso aos recursos com base nas chaves de condições

Ao criar uma IAM política para conceder acesso ao uso de cercas geográficas ou posições de dispositivos, você pode usar [operadores de condição](#) para obter um controle mais preciso sobre quais cercas geográficas ou dispositivos um usuário pode acessar. Você pode fazer isso incluindo o ID da geocerca ou o ID do dispositivo no elemento Condition da sua política.

O exemplo de política a seguir mostra como você pode criar uma política que permita que um usuário atualize as posições de um dispositivo específico.

```

{
  "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"
          ]
        }
      }
    }
  ]
}

```

## Controlar o acesso a recursos com base em tags

Ao criar uma IAM política para conceder acesso para usar seus recursos de localização da Amazon, você pode usar o [controle de acesso baseado em atributos](#) para controlar melhor quais recursos

um usuário pode modificar, usar ou excluir. Você pode fazer isso incluindo as informações da tag no elemento `Condition` da sua política para controlar o acesso com base nas suas [tags](#) de recursos.

O exemplo de política a seguir mostra como você pode criar uma política que permita que um usuário crie geocercas. Isso concede a permissão para as seguintes ações para criar uma ou mais cercas geográficas em uma coleção de cercas geográficas chamada *Collection*:

- `geo:BatchPutGeofence` para criar várias geocercas.
- `geo:PutGeofence` para criar uma única geocerca.

No entanto, essa política usa o `Condition` elemento para conceder a permissão somente se *Collection* tag, `Owner`, tem o valor do nome de usuário desse usuário.

- Por exemplo, se um usuário chamado `richard-roe` tentar visualizar uma localização na Amazon *Collection*, o *Collection* deve ser marcado `Owner=richard-roe` ou `owner=richard-roe`. Caso contrário, o usuário terá o acesso negado.

#### Note

A chave da tag de condição `Owner` corresponde a `Owner` e a `owner` porque os nomes das chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte [Elementos da IAM JSON política: condição](#) no Guia IAM do usuário.

```
{
  "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}"}
      }
    }
  ]
}
```

```
}
```

Para ver um tutorial sobre [como definir permissões para acessar AWS recursos com base em tags](#), consulte o Guia AWS Identity and Access Management do usuário.

## Resolução de problemas de identidade e acesso no Amazon Location Service

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com a Amazon Location e IAM.

### Tópicos

- [Não tenho autorização para executar uma ação no Amazon Location](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de localização da Amazon](#)

### Não tenho autorização para executar uma ação no Amazon Location

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O exemplo de erro a seguir ocorre quando o `mateojackson` IAM usuário tenta usar o console para ver detalhes sobre um `my-example-widget` recurso fictício, mas não tem as permissões fictícias `geo: GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
geo: GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso `my-example-widget` usando a ação `geo: GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Não estou autorizado a realizar iam: PassRole

Caso receba uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, as suas políticas deverão ser atualizadas para permitir que você atribua um perfil ao Amazon Location.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar uma ação na Amazon Location. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de localização da Amazon

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon Location é compatível com esses recursos, consulte [Como o Amazon Location Service funciona com IAM](#).
- Para saber como fornecer acesso aos seus recursos em todas as Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outra Conta da AWS de sua propriedade](#) no Guia do IAM usuário.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

## Resposta a incidentes no Amazon Location Service

A segurança é a maior prioridade na AWS. Como parte do [modelo de responsabilidade compartilhada AWS](#) na nuvem, AWS gerencia uma arquitetura de data center e rede que atende aos requisitos das organizações mais sensíveis à segurança. Como AWS cliente, você compartilha a responsabilidade de manter a segurança na nuvem. Isso significa que você controla a segurança que escolhe implementar a partir das AWS ferramentas e recursos aos quais tem acesso.

Ao estabelecer um nível básico de segurança que atenda aos objetivos de seus aplicativos executados na nuvem, você pode detectar desvios aos quais pode reagir. Como a resposta a incidentes de segurança pode ser um tópico complexo, recomendamos que você analise os seguintes recursos para entender melhor o impacto que a resposta a incidentes (IR) e suas escolhas têm em suas metas corporativas: [Guia de resposta a incidentes de AWS segurança](#), whitepaper de [melhores práticas de AWS segurança](#) e [Estrutura de adoção da AWS nuvem \(\) AWS CAF](#).

## Registro em log e monitoramento no Amazon Location Service

O registro em log e o monitoramento são uma parte importante da resposta a incidentes. Ele permite que você estabeleça uma linha de base de segurança para detectar desvios que você pode investigar e aos quais pode responder. Ao implementar o registro em log e o monitoramento do Amazon Location Service, você pode manter a confiabilidade, a disponibilidade e o desempenho de seus projetos e recursos.

AWS fornece várias ferramentas que podem ajudá-lo a registrar e coletar dados para resposta a incidentes:

### AWS CloudTrail

O Amazon Location Service se integra ao AWS CloudTrail, que é um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço. Isso inclui ações do console do Amazon Location Service e chamadas programáticas para as API operações



de localização da Amazon. Esses registros de ações são chamados de eventos. Para obter mais informações, consulte [Registro e monitoramento do Amazon Location Service com AWS CloudTrail](#).

## Amazon CloudWatch

Você pode usar CloudWatch a Amazon para coletar e analisar métricas relacionadas à sua conta do Amazon Location Service. Você pode ativar CloudWatch os alarmes para notificá-lo se uma métrica atende a determinadas condições e atingiu um limite especificado. Quando você cria um alarme, CloudWatch envia uma notificação para um Amazon Simple Notification Service que você define. Para obter mais informações, consulte o [Monitoramento do Amazon Location Service com a Amazon CloudWatch](#).

## AWS Health Painéis

Ao usar os [painéis do AWS Health](#), você pode verificar o status do serviço Amazon Location Service. Você também pode monitorar e visualizar dados históricos sobre quaisquer eventos ou problemas que possam afetar seu AWS ambiente. Para obter mais informações, consulte o [Guia do usuário do AWS Health](#).

# Validação de compatibilidade do Amazon Location Service

Para saber se um Serviço da AWS está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

**Note**

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso Serviço da AWS fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso Serviço da AWS detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso Serviço da AWS ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Resiliência no Amazon Location Service

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas

de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, o Amazon Location oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

## Segurança da infraestrutura no Amazon Location Service

Como um serviço gerenciado, o Amazon Location Service é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa API chamadas AWS publicadas para acessar a Amazon Location pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

## Análise de configuração e vulnerabilidade no Amazon Location

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você, nosso cliente. Para obter mais informações, consulte o [modelo de responsabilidade AWS compartilhada](#).

## Prevenção do problema do substituto confuso entre serviços

O problema do “confused deputy” é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executar a ação. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

O Amazon Location Service não atua como um serviço de chamadas em seu nome para outros AWS serviços, portanto, você não precisa adicionar essas proteções nesse caso. Para saber mais sobre o problema de “confused deputy”, consulte [O problema confused deputy](#) no Guia do usuário do AWS Identity and Access Management .

## Melhores práticas de segurança para o Amazon Location Service

O Amazon Location Service disponibiliza diversos atributos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

### Melhores práticas de segurança de detecção recomendadas para o Amazon Location Service

As melhores práticas a seguir para o Amazon Location Service podem ajudar a detectar incidentes de segurança:

#### Implemente ferramentas AWS de monitoramento

O monitoramento é fundamental para a resposta a incidentes e mantém a confiabilidade e a segurança dos recursos do Amazon Location Service e de suas soluções. Você pode implementar ferramentas de monitoramento a partir das várias ferramentas e serviços disponíveis AWS para monitorar seus recursos e seus outros AWS serviços.

Por exemplo, a Amazon CloudWatch permite que você monitore métricas para o Amazon Location Service e permite que você configure alarmes para notificá-lo se uma métrica atende a determinadas condições que você definiu e atingiu um limite definido por você. Ao criar um alarme, você pode configurar o envio de uma notificação CloudWatch para alertar usando o Amazon Simple Notification Service. Para obter mais informações, consulte [the section called “Registro e Monitoramento”](#).

### Ativar ferramentas de AWS registro

O registro fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Location Service. Você pode implementar ferramentas de registro, como AWS CloudTrail coletar dados sobre ações para detectar API atividades incomuns.

Ao criar uma trilha, você pode configurar CloudTrail para registrar eventos. Eventos são registros de operações de recursos realizadas em ou dentro de um recurso, como a solicitação feita ao Amazon Location, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando a solicitação foi feita, bem como dados adicionais. Para obter mais informações, consulte [Registro de eventos de dados para trilhas](#) no Guia AWS CloudTrail do usuário.

## Melhores práticas recomendadas de segurança preventiva para o Amazon Location Service

As seguintes melhores práticas para o Amazon Location Service podem ajudar a evitar incidentes de segurança:

### Use conexões seguras

Sempre use conexões criptografadas, como as que começam com `https://` para manter seguras as informações confidenciais em trânsito.

### Implemente o acesso de privilégio mínimo a recursos

Conceder privilégio mínimo ao criar políticas personalizadas Amazon e conceder apenas as permissões necessárias para a execução de uma tarefa. Recomendamos que se inicie com um conjunto mínimo de permissões e que permissões adicionais sejam concedidas quando forem necessárias. A implementação do privilégio mínimo de acesso é fundamental para se reduzir o risco de segurança e o impacto que pode resultar de erros ou ataques maliciosos. Para obter mais informações, consulte [the section called “Identity and Access Management”](#).

## Use um dispositivo globalmente exclusivo IDs IDs

Use as seguintes convenções para o dispositivoIDs.

- O dispositivo IDs deve ser exclusivo.
- O dispositivo não IDs deve ser secreto, pois eles podem ser usados como chaves estrangeiras para outros sistemas.
- O dispositivo não IDs deve conter informações de identificação pessoal (PII), como dispositivo telefônico ou endereços de e-mail. IDs
- O dispositivo não IDs deve ser previsível. Identificadores opacos como esses UUIDs são recomendados.

Não inclua PII nas propriedades de posição do dispositivo

Ao enviar atualizações do dispositivo (por exemplo, usando [DevicePositionUpdate](#)), não inclua informações de identificação pessoal (PII), como número de telefone ou endereço de e-mail no. `PositionProperties`

## Melhores práticas do Amazon Location Service

Este tópico apresenta as melhores práticas para ajudar você a usar o Amazon Location Service. Embora essas melhores práticas possam ajudar você a aproveitar ao máximo o Amazon Location Service, elas não representam uma solução completa. Você deve seguir somente as recomendações aplicáveis ao seu ambiente.

Tópicos

- [Segurança](#)
- [Gerenciamento de recursos](#)
- [Faturamento e gerenciamento de custos](#)
- [Cotas e uso](#)

## Segurança

Para ajudar a controlar ou até mesmo evitar riscos de segurança, considere as seguintes práticas recomendadas:

- Use federação de identidade e IAM funções para gerenciar, controlar ou limitar o acesso aos seus recursos de localização da Amazon. Para obter mais informações, consulte [as IAM melhores práticas](#) no Guia IAM do usuário.
- Siga o Princípio do Privilégio Mínimo para conceder somente o acesso mínimo necessário aos seus recursos do Amazon Location Service. Para obter mais informações, consulte [the section called “Gerenciando acesso usando políticas”](#).
- Para recursos do Amazon Location Service usados em aplicativos web, restrinja o acesso usando uma `aws:referer` IAM condição, limitando o uso por sites que não sejam aqueles incluídos na lista de permissões.
- Use ferramentas de monitoramento e registro para rastrear o acesso e o uso de recursos. Para obter mais informações, consulte [the section called “Registro e Monitoramento”](#) [Registrar eventos de dados para trilhas](#) no Guia AWS CloudTrail do usuário.
- Use conexões seguras, como as que começam com `https://`, para aumentar a segurança e proteger os usuários de ataques enquanto os dados são transmitidos entre o servidor e o navegador.

Para obter mais informações sobre as práticas recomendadas de segurança preventiva e detecção, consulte o tópico sobre [the section called “Melhores práticas de segurança”](#).

## Gerenciamento de recursos

Para ajudar a gerenciar com eficiência os seus recursos de localização no Amazon Location Service, considere as seguintes práticas recomendadas:

- Use endpoints regionais que sejam fundamentais para a sua base de usuários esperada e para melhorar a experiência deles. Para obter mais informações sobre os endpoints da região, consulte [Regiões e endpoints do Amazon Location](#).
- Para recursos que usam provedores de dados, como recursos de mapas e recursos de índice de local, lembre-se de seguir o contrato de termos de uso do provedor de dados específico. Para obter mais informações, consulte [Provedores de dados](#).
- Minimize a criação de recursos utilizando um recurso para cada configuração de mapa, índice de lugares, ou rotas. Em uma região, você normalmente precisa de apenas um recurso por provedor de dados ou estilo de mapa. A maioria dos aplicativos usa recursos existentes e não cria recursos em tempo de execução (run time).
- Ao usar recursos diferentes em um único aplicativo, como um recurso de mapa e uma calculadora de rotas, use o mesmo provedor de dados em cada recurso para garantir que os dados se

correspondam. Por exemplo, que uma geometria de rota que você cria com a sua calculadora de rotas esteja alinhada com as ruas no mapa desenhado pelo recurso de mapas.

## Faturamento e gerenciamento de custos

Para ajudar a gerenciar seus custos e seu faturamento, considere as seguintes práticas recomendadas:

- Use ferramentas de monitoramento, como a Amazon CloudWatch, para rastrear seu uso de recursos. Você pode definir alertas que notificam você quando o uso estiver prestes a exceder os limites especificados. Para obter mais informações, consulte [Criação de um alarme de cobrança para monitorar suas AWS cobranças estimadas](#) no Guia do CloudWatch usuário da Amazon.

## Cotas e uso

Você Conta da AWS inclui cotas que definem um limite padrão para a quantidade de uso. Você pode configurar alarmes para alertar você quando o seu uso estiver próximo do limite e solicitar um aumento a uma cota, quando precisar. Para obter mais informações sobre como trabalhar com cotas, consulte os tópicos a seguir.

- [Amazon Location Service quotas](#)
- [Usando CloudWatch para monitorar o uso em relação às cotas](#)
- [Visualizando suas cotas de serviço e definindo alarmes no Guia](#) do usuário da Amazon CloudWatch .

Você pode criar alarmes para avisar com antecedência quando você estiver perto de exceder seus limites. Recomendamos definir alarmes para cada cota em cada Região da AWS local em que você usa a Amazon Location. Por exemplo, você pode monitorar o seu uso da operação `SearchPlaceIndexForText` e criar um alarme quando você exceder 80% da sua cota atual.

Ao receber um aviso de alarme sobre sua cota, você deve decidir o que fazer. Talvez você esteja usando recursos adicionais porque a sua base de clientes cresceu. Nesse caso, talvez você queira solicitar um aumento em sua cota, como um aumento de 50% na cota para uma API chamada nessa região. Ou talvez haja um erro no seu serviço que leve você a fazer chamadas adicionais desnecessárias para o Amazon Location. Nesse caso, talvez você deva resolver o problema com a sua operadora.



## Histórico do documento

A tabela a seguir descreve a documentação do Amazon Location Service. Para receber notificações sobre atualizações, inscreva-se em um feed RSS.

Alteração	Descrição	Data
<a href="#">Amazon Location Service lança um novo SDK para JavaScript</a>	Para facilitar o desenvolvimento de aplicativos Amazon Location com front-ends na web, o Amazon Location adiciona um novo SDK de código aberto que suporta o AWS SDK para JavaScript v3, simplificando a autenticação e usando o GeoJSON. Para obter mais informações, consulte <a href="#">SDK do Amazon Location</a> .	6 de julho de 2023
<a href="#">O Amazon Location Service lança chaves de API para disponibilidade geral</a>	O Amazon Location adiciona suporte para lugares e rotas e anuncia a disponibilidade geral do atributo de chaves de API. Para obter mais informações, consulte <a href="#">Usando as chaves de API</a> .	6 de julho de 2023
<a href="#">Amazon Location Service adiciona EventBridge eventos da Amazon para atualizações de posição</a>	O Amazon Location adiciona suporte para enviar eventos de atualização da posição do rastreador para EventBridge. Para obter mais informações, incluindo como ativar os eventos para um rastreado	6 de julho de 2023

---

<a href="#">O Amazon Location adiciona metadados às geocercas</a>	<p>r, consulte <a href="#">Reagindo aos eventos com</a>. EventBridge</p> <p>Usando a API do Amazon Location, agora você pode adicionar propriedades de metadados às suas geocercas . Eles são armazenados com sua cerca geográfica e incluídos em eventos relacionados à cerca geográfica na Amazon. EventBridge Para obter mais informações, consulte <a href="#">Desenhe geocercas</a> e <a href="#">Crie regras para eventos</a>.</p>	15 de junho de 2023
<a href="#">O Amazon Location adiciona categorias para lugares</a>	<p>O Amazon Location adiciona categorias aos resultados da pesquisa por lugares e aos resultados da filtragem por categoria. Para obter mais informações, consulte <a href="#">Categorias e filtragem</a>.</p>	15 de junho de 2023
<a href="#">O Amazon Location apresenta visualizações políticas</a>	<p>O Amazon Location adiciona visualizações políticas a determinados estilos de mapa. Para obter mais informações, consulte <a href="#">Visualizações políticas</a>.</p>	23 de maio de 2023

[O Amazon Location apresenta novos sites de demonstração e de amostra](#)

O Amazon Location anuncia um site novo que lhe dá acesso a demonstrações e amostras do Amazon Location. Para obter mais informações, consulte o [Site de demonstração do Amazon Location](#).

3 de maio de 2023

[Amazon Location introduz rotas mais longas em CalculateRouteMatrix](#)

Agora, o Amazon Location permite rotas de comprimento ilimitado para rotas de matriz de rotas criadas com o provedor de dados HERE. Para obter mais informações, consulte [Planejamento de rotas mais longas](#).

24 de abril de 2023

[A documentação do Amazon Location adiciona diferenças de atributos por provedor de dados](#)

A documentação do Amazon Location foi atualizada com informações sobre as diferenças entre cada provedor de dados nos mapas, na pesquisa por lugares e nas rotas. Para obter mais informações, consulte [Atributos do provedor de dados](#).

30 de março de 2023

[Disponibilidade geral dos mapas do Amazon Location Open Data](#)

Disponibilidade geral do provedor e estilo de dados do Amazon Location Service, com base nos mapas OpenStreetMap do Daylight. Para obter mais informações, consulte [Open Data](#).

7 de março de 2023

[O Amazon Location adiciona um novo método de autorização na pré-visualização](#)

O Amazon Location Service adiciona chaves de API como um novo método de autorização para usuários anônimos, em modo de pré-visualização. Para obter mais informações, consulte [Permitindo o acesso de convidados não autenticados ao seu aplicativo usando chaves de API](#).

23 de fevereiro de 2023

[Atualização da documentação do Amazon Location com as melhores práticas mais recentes do IAM](#)

A documentação do Amazon Location Service foi atualizada para atender às AWS Identity and Access Management melhores práticas mais recentes. Para obter mais informações, consulte [Segurança no Amazon Location Service](#).

26 de janeiro de 2023

[Amazon Location Service é adicionado GrabMaps como provedor de dados no sudeste da Ásia](#)

A Amazon Location é introduzida da GrabMaps como provedora de dados no sudeste da Ásia. Para obter mais informações, consulte [GrabMaps](#).

10 de janeiro de 2023

[Mapas do Amazon Location Service Open Data em pré-visualização](#)

Novo provedor e estilo de dados de localização da Amazon adicionados em versão prévia pública, com base nos OpenStreetMap mapas Daylight. Para obter mais informações, consulte [Pré-visualização](#).

15 de dezembro de 2022

### [Novos estilos de imagens de satélite do HERE](#)

Dois estilos novos de mapa foram adicionados para mapas que usam o HERE como provedor de dados, os estilos de mapa do HERE Satellite Imagery e do HERE Hybrid. Para obter mais informações, consulte [Estilos de mapa do HERE](#).

25 de outubro de 2022

### [Unidades em endereços](#)

Agora, o Amazon Location Service oferece suporte para unidades em endereços, por exemplo, “Rua principal, 123 - Apartamento 3B, Qualquer Cidade, Brasil”.

20 de setembro de 2022

### [Ache lugares com um ID](#)

O Amazon Location Service agora inclui suporte para encontrar a localização exata sugerida pela operação `SearchPlaceIndexForSuggestions` usando a operação `GetPlace`. Consulte [Usando o preenchimento automático](#).

20 de setembro de 2022

### [Chaves de condição adicionais para a política do IAM](#)

Agora, o Amazon Location Service oferece suporte a chaves de condição adicionais que permitem que você defina o acesso para geocercas ou dispositivos específicos na política do IAM. Consulte [Chaves de condição](#).

23 de agosto de 2022

---

<a href="#">Geocercas circulares</a>	Agora, o Amazon Location Service oferece suporte a geocercas definidas como um círculo com um ponto central e um raio, para obter eventos quando os dispositivos estão a uma certa distância de um local. Consulte <a href="#">Adicionando geocercas circulares</a> .	11 de agosto de 2022
<a href="#">Referência combinada para APIs</a>	Agora, o Amazon Location Service tem um único guia de referência para APIs, em vez de guias diferentes para cada subserviço. Para obter mais informações sobre as APIs, consulte <a href="#">APIs do Amazon Location</a> .	7 de julho de 2022
<a href="#">Integração com o Service Quotas</a>	Agora, o Amazon Location está <a href="#">integrado ao Service Quotas</a> , permitindo que você visualize e gerencie suas cotas por meio do AWS Management Console ou usando o AWS CLI.	6 de julho de 2022
<a href="#">Capítulo de documentação de conceitos atualizado</a>	O <a href="#">capítulo de conceitos do Amazon Location</a> foi atualizado com mais informações para seus usuários.	22 de abril de 2022

---

<a href="#">Novo tutorial de início rápido para Android</a>	Um novo <a href="#">tutorial de início rápido</a> para desenvolvimento para Android usando o Kotlin foi adicionado para que os desenvolvedores estejam prontos rapidamente.	15 de abril de 2022
<a href="#">Novos estilos de mapa do HERE</a>	Dois novos estilos de mapa foram adicionados para mapas que usam o HERE como provedor de dados. Para obter mais informações, consulte <a href="#">Estilos de mapa do HERE</a> .	15 de março de 2022
<a href="#">Reestruture a documentação com mais exemplos de código e tutoriais</a>	Este guia do desenvolvedor foi reestruturado para que você encontre tópicos com mais facilidade, incluindo capítulos novos de <a href="#">Início rápido</a> e <a href="#">Exemplos de código</a> .	25 de fevereiro de 2022
<a href="#">Filtragem de posições baseada em precisão para rastreadores</a>	Agora você pode usar filtros baseados em precisão ao <a href="#">criar um recurso de rastreador</a> .	7 de dezembro de 2021
<a href="#">Preenchimento automático para os índices de local</a>	Agora você pode usar o <a href="#">preenchimento automático</a> ao pesquisar os índices de local.	6 de dezembro de 2021
<a href="#">Novo tutorial do Amplify para o uso de mapas</a>	Um novo tutorial está disponível mostrando como usar AWS Amplify para exibir mapas em uma aplicativo web. O tutorial está disponível em <a href="#">Usando a biblioteca do Amplify com o Amazon Location Service</a> .	24 de novembro de 2021

## [Extensões de consulta para locais](#)

Agora, o Amazon Location Service oferece suporte para a configuração de um idioma preferencial para os resultados durante a geocodificação, ou geocodificação reversa, e adiciona o fuso horário e outras informações aos resultados. Para obter mais informações sobre geocodificação e geocodificação reversa, consulte [Geocodificação, geocodificação reversa e pesquisas](#).

16 de novembro de 2021

## [Filtragem de posições do rastreador](#)

O Amazon Location Service adiciona um novo atributo de filtragem de posições aos rastreadores que pode ajudar você a controlar custos. Esse atributo filtra algumas atualizações de posição nos dispositivos antes que as atualizações sejam armazenadas ou avaliadas em relação às geocercas. Para obter mais informações sobre a filtragem de posições, consulte [Rastreadores](#).

5 de outubro de 2021



Operações de atualização

As seguintes operações foram adicionadas às referências da API do Amazon Location Service: [UpdateMapUpdatePlaceIndexUpdateRouteCalculator](#), [UpdateGeofenceCollection](#), [UpdateTracker](#).

19 de julho de 2021

Atualização do tutorial: Amazon Aurora PostgreSQL funções definidas pelo usuário

Foi adicionado um novo tutorial sobre como usar [Amazon Aurora PostgreSQL funções definidas pelo usuário com o Amazon Location](#) para validar, limpar e enriquecer os dados geoespaciais.

19 de julho de 2021

CloudFormation Recursos da AWS

O Amazon Location agora oferece suporte [à criação dos seguintes tipos de CloudFormation recursos nos recursos da AWS](#):

```
AWS::Location::Map
  AWS::Location::PlaceIndex
  AWS::Location::RouteCalculator
  AWS::Location::Tracker
  AWS::Location::TrackerConsumer
  AWS::Location::GeofenceCollection
```

7 de junho de 2021

---

<a href="#">Marcação de recursos com tags</a>	Agora você pode <a href="#">adicionar tags aos seus recursos do Amazon Location</a> para ajudar a gerenciar, identificar, organizar, pesquisar e filtrar seus recursos.	1º de junho de 2021
<a href="#">Disponibilidade geral</a>	Versão para disponibilidade geral da documentação para o desenvolvedor do Amazon Location Service: <a href="#">Região e endpoints</a> e <a href="#">Service Quotas</a> atualizadas.	1º de junho de 2021
<a href="#">Esri Imagery</a>	Agora, o Amazon Location oferece suporte para o uso do estilo de mapa da Esri: <a href="#">Esri Imagery</a> . Para obter mais informações, consulte <a href="#">Esri World Imagery</a> no site da Esri.	1º de junho de 2021
<a href="#">Calculando rotas</a>	Agora você pode <a href="#">usar as calculadoras de rotas de localização da Amazon</a> para calcular rotas e estimar o tempo de viagem com base na rede up-to-date rodoviária e nas informações de tráfego ao vivo do provedor de dados escolhido.	1º de junho de 2021

[AWS KMS criptografia de chave gerenciada pelo cliente para dados em repouso](#)

Agora, o Amazon Location oferece suporte para o uso de uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia para [adicionar uma segunda camada de criptografia sobre a criptografia existente AWS](#).

1º de junho de 2021

[Versão de pré-visualização pública](#)

Versão inicial da documentação para pré-visualização pública.

16 de dezembro de 2020

[Atualização do tutorial: exibindo mapas](#)

Os tutoriais para exibir mapas usando MapLibre para Android e iOS foram atualizados para usar o SDK MapLibre nativo.

17 de março de 2020

# AWS Glossário

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.