

Guia do desenvolvedor

AWS Mobile SDK para Unity



AWS Mobile SDK para Unity: Guia do desenvolvedor

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas 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

| | |
|---|----|
| | vi |
| O que é o AWS Mobile SDK para Unity? | 1 |
| Guias e tópicos relacionados | 1 |
| Conteúdo de referência arquivado | 1 |
| Compatibilidade | 2 |
| Fazer download do Mobile SDK para Unity | 2 |
| O que está incluído no Mobile SDK para Unity? | 2 |
| Configuração do AWS Mobile SDK para Unity | 3 |
| Pré-requisitos | 3 |
| Etapa 1: fazer download do AWS Mobile SDK para Unity | 3 |
| Etapa 2: configurar o AWS Mobile SDK para Unity | 4 |
| Criar um Cenário | 4 |
| Defina a região padrão do Serviço da AWS | 4 |
| Defina as Informações de Log | 4 |
| Trabalho com o arquivo link.xml | 5 |
| Etapa 3: obter o ID do grupo de identidades usando o Amazon Cognito | 6 |
| Próximas etapas | 7 |
| Conceitos básicos do AWS Mobile SDK para Unity | 8 |
| Amazon Cognito Identity | 8 |
| Amazon Cognito Sync | 8 |
| Uso do exemplo do CognitoSyncManager | 9 |
| Dynamo DB | 9 |
| Uso do exemplo do DynamoDB | 10 |
| Mobile Analytics | 10 |
| Configuração do Mobile Analytics | 11 |
| Uso do exemplo do Mobile Analytics | 11 |
| Amazon S3 | 12 |
| Configurando o padrão de assinatura do S3 | 12 |
| Uso do exemplo do S3 | 13 |
| Amazon Simple Notification Service | 13 |
| AWS Lambda | 14 |
| Amazon Cognito Identity | 15 |
| O que é o Amazon Cognito Identity? | 15 |
| Usando um provedor público para autenticar usuários | 15 |

| | |
|---|----|
| Uso de Identidades autenticadas pelo desenvolvedor | 15 |
| Amazon Cognito Sync | 16 |
| Amazon Mobile Analytics | 17 |
| Integração do Amazon Mobile Analytics | 17 |
| Criação de um aplicativo no console do Mobile Analytics | 17 |
| Integração do Mobile Analytics ao aplicativo | 17 |
| Registro de eventos de monetização | 18 |
| Registro de eventos personalizados | 19 |
| Registro de sessões | 19 |
| Amazon Simple Storage Service (S3) | 21 |
| Criação e configuração de um bucket do S3 | 21 |
| Crie um bucket do S3 | 21 |
| Definir permissões para o S3 | 21 |
| Upload de arquivos no console | 22 |
| (opcional) Configuração da versão de assinatura para solicitações do S3 | 23 |
| Criação do cliente Amazon S3 | 23 |
| Listagem de buckets | 23 |
| Listagem de objetos | 24 |
| Download de um objeto | 25 |
| Upload de um objeto | 25 |
| Amazon DynamoDB | 27 |
| Integração ao Amazon DynamoDB | 27 |
| Criação de uma tabela do DynamoDB | 28 |
| Criação de um cliente DynamoDB | 28 |
| Descrição de uma tabela | 29 |
| Salvar um objeto | 30 |
| Criação de um livro | 31 |
| Recuperação de um livro | 31 |
| Atualização de um livro | 32 |
| Exclusão de um livro | 32 |
| Amazon Simple Notification Service | 34 |
| Pré-requisitos | 3 |
| Definir permissões para o SNS | 34 |
| Pré-requisitos do iOS | 35 |
| Pré-requisitos do Android | 35 |
| Configuração do Unity Sample App para iOS | 35 |

| | |
|--|----|
| Configuração do Unity | 36 |
| Configuração do iOS | 36 |
| Configuração de SNS | 38 |
| Uso do Xcode | 38 |
| Amostra do Unity (iOS) | 39 |
| Configuração do Unity Sample App para Android | 40 |
| Configuração do Unity | 40 |
| Configuração do Android | 41 |
| Configuração de SNS | 41 |
| Unity Sample (Android) | 42 |
| AWS Lambda | 44 |
| Permissões | 44 |
| Configuração do projeto | 45 |
| Definir as permissões para o AWS Lambda | 45 |
| Criar uma nova função de execução | 45 |
| Criação de uma função no AWS Lambda | 45 |
| Criar um cliente do Lambda | 46 |
| Criar um objeto de solicitação | 46 |
| Invocar a função do Lambda | 46 |
| Solução de problemas | 48 |
| Verificação da existência de permissões necessárias na função do IAM | 48 |
| Uso de um depurador de proxy HTTP | 49 |

O AWS Mobile SDK para Unity agora está incluído na AWS SDK for .NET. Este guia faz referência à versão arquivada do Mobile SDK para Unity. Para obter mais informações, consulte [O que é o AWS Mobile SDK para Unity?](#).

O que é o AWS Mobile SDK para Unity?

O AWS Mobile SDK para Unity agora está incluído na AWS SDK for .NET. Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for .NET](#).

Este guia não é mais atualizado. Ele faz referência à versão arquivada do Mobile SDK para Unity.

Guias e tópicos relacionados

- Para desenvolvimento de aplicativos front-end e móveis, recomendamos o uso da [AWS Amplify](#).
- Para considerações especiais sobre o uso da AWS SDK for .NET em seus aplicativos Unity, consulte [Considerações especiais sobre o suporte ao Unity](#) no AWS SDK for .NET Guia do desenvolvedor.
- Para fins de referência, você pode encontrar a versão arquivada do [AWS Mobile SDK para Unity](#) no GitHub.

Conteúdo de referência arquivado

O Mobile SDK para Unity arquivado contém um conjunto de classes .NET que permite que jogos escritos com Unity utilizem os Serviços da AWS. Os aplicativos escritos com o Mobile SDK para Unity podem ser executados em dispositivos iOS ou Android.

Entre os Serviços da AWS compatíveis estão:

- [Amazon Cognito](#)
- [Amazon DynamoDB](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Amazon Kinesis Data Streams](#)
- [AWS Lambda](#)
- [Amazon Mobile Analytics](#)
- [Amazon Simple Email Service \(Amazon SES\)](#)
- [Amazon Simple Notification Service \(Amazon SNS\)](#)
- [Amazon Simple Queue Service \(Amazon SQS\)](#)
- [Amazon Simple Storage Service \(Amazon S3\)](#)

Esses serviços permitem autenticar usuários, salvar dados de jogadores e jogos, salvar objetos na nuvem, enviar notificações por push, e coletar e analisar dados de uso.

Compatibilidade

O Mobile SDK para Unity v3 é compatível com Unity versões 4.6 e posteriores.

A versão mais recente do Mobile SDK para Unity apresentou aprimoramentos que podem exigir que você altere seu código durante a incorporação ao seu projeto. Para obter mais informações sobre essas mudanças, consulte [Melhorias no AWS Mobile SDK para Unity](#) no blog Serviços Web e móveis de front-end.

Fazer download do Mobile SDK para Unity

Você também pode fazer download do Mobile SDK para Unity como um arquivo .zip [aqui](#).

O que está incluído no Mobile SDK para Unity?

Para ver a lista completa de pacotes NuGet, amostras e outros arquivos no Mobile SDK para Unity, consulte [AWS SDK for .NET](#) no GitHub.

Configuração do AWS Mobile SDK para Unity

Para começar a usar o AWS Mobile SDK para Unity, você pode configurar o SDK e começar a criar um novo projeto, ou integrar o SDK a um projeto existente. Você também pode clonar e executar os [exemplos](#) para ter uma ideia de como funciona o SDK.

Pré-requisitos

Antes de usar o AWS Mobile SDK para Unity, você precisará do seguinte:

- [Uma conta da AWS](#)
- Unity na versão 4.x ou 5.x (O Unity 4.6.4p4 ou Unity 5.0.1p3 será obrigatório se você deseja gravar aplicativos que são executados no iOS de 64 bits)

Após preencher os pré-requisitos, você precisará fazer o seguinte para começar:

1. Fazer download do AWS Mobile SDK para Unity.
2. Configurar o AWS Mobile SDK para Unity.
3. Obter credenciais da AWS, usando o Amazon Cognito.

Etapa 1: fazer download do AWS Mobile SDK para Unity

Em primeiro lugar, [faça download do AWS Mobile SDK para Unity](#). Cada pacote no SDK é obrigatório para utilizar o serviço correspondente da AWS, de acordo com o nome do pacote. Por exemplo, o pacote `aws-Unity-sdk-dynamodb-2.1.0.0.unitypackage` é usado para chamar o serviço do AWS DynamoDB. Você pode importar todos os pacotes ou apenas os que pretende usar.

1. Abra o editor do Unity editor e crie um novo projeto em branco; use as configurações padrão.
2. Selecione Assets (Ativos) > Import Package (Importar pacote) > Custom Package (Pacote personalizado).
3. Na caixa de diálogo Import (Importar) pacote, navegue e selecione os arquivos `.unitypackage` arquivos que deseja usar.
4. Na caixa de diálogo Importing (Importação) de pacotes, certifique-se de que todos os itens estão selecionados e clique em Import (Importar).

Etapa 2: configurar o AWS Mobile SDK para Unity

Criar um Cenário

Ao trabalhar com o AWS Mobile SDK para Unity, você pode começar a usá-lo ao incluir a seguinte linha de código no comportamento mono do Start ou método Awake de sua classe de conduta:

```
UnityInitializer.AttachToGameObject(this.gameObject);
```

Crie seu cenário, selecionando New Scene (Novo Cenário) no menu File (Arquivo).

O AWS SDK para Unity contém as classes de clientes referentes a cada serviço da AWS compatível. Esses clientes são configurados usando um arquivo chamado awsconfig.xml. A seção a seguir descreve as configurações mais usadas no arquivo awsconfig.xml. Para obter mais informações sobre essas configurações, consulte a [Referência de API para o Unity SDK](#).

Defina a região padrão do Serviço da AWS

Para configurar a região padrão de todos os clientes do serviço:

```
<aws region="us-west-2" />
```

Isso define a região padrão para todos os clientes de serviços no Unity SDK. Esta configuração pode ser substituída, especificando explicitamente a região no momento da criação de uma instância do cliente de serviço, da seguinte forma:

```
IAmazonS3 s3Client = new AmazonS3Client(<credentials>, RegionEndpoint.USEast1);
```

Defina as Informações de Log

As configurações de registro são especificadas da seguinte maneira:

```
<logging logTo="UnityLogger"  
    logResponses="Always"  
    logMetrics="true"  
    logMetricsFormat="JSON" />
```

Esta configuração é usada para configurar o registro em log no Unity. Quando você se registra em UnityLogger, a estrutura imprime internamente o resultado para os registros de depuração. Se

you want to register HTTP responses, define the `logResponses` flag – the values can be `Always`, `Never` or `OnError`. You can also register HTTP metrics, using the `logMetrics` property. The format of the log can be specified using the `logMetricsFormat` property. Valid values are `JSON` or `default`.

The following example shows the most common configurations in the `awsconfig.xml` file. For more information about service-specific configurations, see the following section:

```
<?xml version="1.0" encoding="utf-8"?>
<aws region="us-west-2"
  <logging logTo="UnityLogger"
    logResponses="Always"
    logMetrics="true"
    logMetricsFormat="JSON" />
/>
```

Trabalho com o arquivo link.xml

The SDK uses reflection for platform-specific components. If you are using the `IL2CPP` script backend, `strip_bytecode` is always enabled on iOS, so you need to have a `link.xml` file in your root project, containing the following entries:

```
<linker>
<!-- if you are using AWSConfigs.HttpClient.UnityWebRequest option-->
<assembly fullname="UnityEngine">
  <type fullname="UnityEngine.Networking.UnityWebRequest" preserve="all" />
  <type fullname="UnityEngine.Networking.UploadHandlerRaw" preserve="all" />
  <type fullname="UnityEngine.Networking.UploadHandler" preserve="all" />
  <type fullname="UnityEngine.Networking.DownloadHandler" preserve="all" />
  <type fullname="UnityEngine.Networking.DownloadHandlerBuffer" preserve="all" />
</assembly>
<assembly fullname="mscorlib">
  <namespace fullname="System.Security.Cryptography" preserve="all"/>
</assembly>
<assembly fullname="System">
  <namespace fullname="System.Security.Cryptography" preserve="all"/>
</assembly>
<assembly fullname="AWSSDK.Core" preserve="all"/>
<assembly fullname="AWSSDK.CognitoIdentity" preserve="all"/>
<assembly fullname="AWSSDK.SecurityToken" preserve="all"/>
add more services that you need here...
```

```
</linker>
```

Etapa 3: obter o ID do grupo de identidades usando o Amazon Cognito

Para usar os serviços da AWS em seu aplicativo móvel, você deve obter o ID do grupo de identidades, usando a identidade do Amazon Cognito. Usar o Amazon Cognito para obter o ID do grupo de identidades permite que o seu aplicativo acesse os serviços da AWS sem a necessidade de incorporar suas credenciais privadas ao aplicativo. Esta ação também permite que você defina permissões para controlar a quais serviços da AWS os usuários têm acesso.

Para começar a usar o Amazon Cognito, é necessário criar um grupo de identidades. Um grupo de identidades é um repositório de dados de identidade de usuários específicos para sua conta. Cada grupo de identidades dispõe de funções do IAM configuráveis, que permitem que você especifique quais serviços da AWS os usuários do seu aplicativo poderão acessar. Normalmente, o desenvolvedor usará apenas um grupo de identidades por aplicativo. Para obter mais informações sobre grupos de identidades, consulte o [Guia do desenvolvedor do Amazon Cognito](#).

Para criar um grupo de identidades para o aplicativo:

1. Faça login no [Console do Amazon Cognito](#) e clique em Create new identity pool (Criar novo grupo de identidades).
2. Insira um nome referente ao Grupo de identidades e marque a caixa de seleção para habilitar o acesso a identidades não autenticadas. Clique em Create Pool (Criar grupo) para criar o grupo de identidades.
3. Clique em Allow (Permitir) para criar as duas funções padrão associadas ao grupo de identidades, uma para usuários não autenticados e outra para usuários autenticados. Essas funções padrão oferecem ao grupo de identidades acesso à sincronização do Cognito e ao Mobile Analytics.

A próxima página exibe o código que cria um provedor de credenciais, para que você possa integrar facilmente a identidade do Cognito ao aplicativo Unity. Transmita o objeto do provedor de credenciais para o construtor do cliente da AWS que está usando. O código é semelhante a:

```
CognitoAWSCredentials credentials = new CognitoAWSCredentials (  
    "IDENTITY_POOL_ID", // Identity Pool ID  
    RegionEndpoint.USEast1 // Region  
);
```

Próximas etapas

- Conceitos básicos: leia [Conceitos básicos do AWS Mobile SDK para Unity](#), a fim de obter uma visão geral mais detalhada dos serviços incluídos no SDK.
- Executar as demonstrações: exiba os nossos [aplicativos de exemplo do Unity applications](#) que demonstrem casos comuns de utilização. Para executar aplicativos de exemplo, configure o SDK for Unity conforme descrito anteriormente e, em seguida, siga as instruções contidas nos arquivos README de cada exemplo.
- Leia a Referência da API: visualize a [Referência da API](#) correspondente ao AWS Mobile SDK para Unity.
- Fazer perguntas: publique perguntas nos [Fóruns do AWS Mobile SDK](#) ou [abra um problema no Github](#).

Conceitos básicos do AWS Mobile SDK para Unity

Esta página fornece uma visão geral de cada serviço da AWS no AWS Mobile SDK para Unity, bem como as instruções sobre como configurar os exemplos do Unity. Basta concluir todas as instruções sobre como [Configurar a página do AWS Mobile SDK para Unity](#) antes de começar a usar os serviços a seguir.

Amazon Cognito Identity

Todas as chamadas feitas para a AWS precisam das credenciais da AWS. Em vez de codificar suas credenciais nos aplicativos, recomendamos que utilize o [Amazon Cognito Identity](#) para fornecer credenciais da AWS ao aplicativo. Siga as instruções do item [Configurar o AWS Mobile SDK para Unity](#) para obter credenciais da AWS por meio do Amazon Cognito.

O Cognito permite também a autenticação de usuários que usam provedores públicos de login, como a Amazon, o Facebook, o Twitter e o Google, além de provedores compatíveis com o [OpenID Connect](#). O Cognito também funciona com usuários não autenticados. O Cognito fornece credenciais temporárias, com direitos limitados de acesso especificados por você, usando uma função do [Identity and Access Management](#) (IAM). O Cognito é configurado com a criação de um novo grupo de identidades associado a uma função do IAM. A função do IAM especifica os recursos/serviços que o aplicativo pode acessar.

Para começar a usar o Cognito Identity, consulte o [Guia do desenvolvedor do Amazon Cognito](#).

Amazon Cognito Sync

O [Cognito Sync](#) facilita salvar dados dos usuários finais, como preferências do usuário ou o estado do jogo, na Nuvem AWS. Assim, eles podem ser disponibilizados aos usuários, independentemente do dispositivo utilizado. O Cognito também pode salvar esses dados localmente, o que permite que os aplicativos funcionem, mesmo quando uma conexão com a Internet não estiver disponível. Quando uma conexão com a Internet estiver disponível, os aplicativos poderão sincronizar os dados locais na nuvem.

Para começar a usar o Cognito Sync, consulte o [Guia do desenvolvedor do Amazon Cognito](#).

Uso do exemplo do CognitoSyncManager

No painel Project (Projeto), navegue até Assets (Ativos)/AWSSDK/examples (exemplos)/CognitoSync. No lado direito do painel, selecione o cenário do CognitoSync para abri-lo.

Para executar o exemplo, clique no botão de reprodução na parte superior da tela do editor. Quando o aplicativo for executado, ele exibirá algumas caixas de texto e botões que permitirão a inserção de algumas informações do jogador. A seguir, há uma série de botões que salvam as informações localmente, sincronizam as informações locais e atualizam os dados dele na nuvem do Cognito e excluem as informações locais do jogador. Pressione cada um dos botões para executar uma operação. O exemplo exibe o feedback na parte superior da tela do jogo.

Para configurar o exemplo do CognitoSyncManager, você deve especificar um ID do grupo do Cognito Identity. Para especificar esse valor no editor do Unity, selecione SyncManager no painel Hierarchy (Hierarquia) e insira-o na caixa de texto IDENTITY_POOL_ID no Inspector Pane (Painel Inspector).

Note

O exemplo do CognitoSyncManager contém o código que ilustra a forma de utilização do provedor de identidade do Facebook. Pesquise a macro "USE_FACEBOOK_LOGIN". Esta ação requer o uso do Facebook SDK para Unity. Para obter mais informações, consulte o [Facebook SDK para Unity](#).

Dynamo DB

[O Amazon DynamoDB](#) é um serviço de banco de dados rápido, altamente disponível, altamente escalável, econômico e não relacional. O DynamoDB remove limitações de escalabilidade tradicionais sobre armazenamento de dados, mantendo, ao mesmo tempo, a baixa latência e o desempenho previsível.

O AWS SDK para Unity fornece bibliotecas de níveis inferior e superior, para que você possa trabalhar com o DynamoDB. A biblioteca de nível superior inclui o DynamoDB Object Mapper, que permite o mapeamento de classes do cliente para as tabelas do DynamoDB, a execução de várias operações de criação, leitura, atualização e exclusão (CRUD); além da execução de consultas. Com o uso do DynamoDB Object Mapper, é possível gravar um código legível e simples, que armazena os objetos na nuvem.

Para obter mais informações sobre o DynamoDB, consulte o [Guia do desenvolvedor do DynamoDB](#).

Para obter mais informações sobre o uso do DynamoDB nos aplicativos do Unity, consulte o [Amazon DynamoDB](#).

Uso do exemplo do DynamoDB

No painel Project (Projeto), navegue até Assets (Ativos)/AWSSDK/ examples (exemplos)/DynamoDB. Este exemplo é composto pelos seguintes cenários:

- DynamoDBExample – o cenário inicial do aplicativo
- LowLevelDynamoDbExample – exemplo que usa a API do DynamoDB de nível inferior
- TableQueryAndScanExample – exemplo que mostra como executar consultas
- HighLevelExample – exemplo que usa a API do DynamoDB de nível superior

Adicione estes cenários na compilação (na ordem em que aparecem acima) usando a caixa de diálogo Configurações do desenvolvedor (abrir ao selecionar as configurações File.Build). Esta amostra cria quatro tabelas: ProductCatalog, Forum, Thread, Reply.

Para executar o exemplo, clique no botão de reprodução na parte superior da tela do editor. Quando o aplicativo for executado, ele exibirá uma série de botões:

- Operações da tabela de nível superior – ilustra como criar, listar, atualizar, descrever e excluir tabelas.
- Operações de nível intermediário Consulta e Varredura – ilustra como executar consultas.
- Mapeador de objetos de nível superior – ilustra como criar, atualizar e excluir objetos.

Mobile Analytics

Com o uso do [Amazon Mobile Analytics](#), é possível acompanhar os comportamentos dos clientes, agregar métricas, gerar visualizações de dados, além de identificar padrões significativos. O AWS SDK para Unity oferece integração com o serviço Amazon Mobile Analytics. Para obter mais informações sobre o Mobile Analytics, consulte o [Guia do usuário do Mobile Analytics](#). Para obter mais informações sobre o uso do Mobile Analytics nos aplicativos do Unity, consulte o [Amazon Mobile Analytics](#).

Configuração do Mobile Analytics

O Mobile Analytics define algumas configurações que podem ser especificadas no arquivo `awsconfig.xml`:

```
<mobileAnalytics sessionTimeout = "5"  
    maxDBSize = "5242880"  
    dbWarningThreshold = "0.9"  
    maxRequestSize = "102400"  
    allowUseDataNetwork = "false"/>
```

- `sessionTimeout` – Este é o intervalo de tempo após o aplicativo ficar em segundo plano e quando a sessão pode ser encerrada.
- `maxDBSize` – Este é o tamanho do banco de dados do SQLite. Quando o banco de dados atingir o tamanho máximo, todos os eventos adicionais serão removidos.
- `dbWarningThreshold` – Este é o limite para o tamanho do banco de dados que, uma vez atingido, gerará os logs de aviso.
- `maxRequestSize` – Este é o tamanho máximo da solicitação na qual bytes que devem ser transmitidos em uma solicitação HTTP para o serviço de análise móvel.
- `allowUseDataNetwork` – Um booleano que especifica se os eventos da sessão serão enviados na rede de dados.

Uso do exemplo do Mobile Analytics

No painel Project (Projeto), navegue até Ativos (Assets)/AWSSDK/ examples (exemplos)/Mobile Analytics. No lado direito do painel, selecione o cenário de Amazon Mobile Analytics Sample (Exemplo do Amazon Mobile Analytics) para abrir o cenário. Para usar a amostra, é preciso adicionar o aplicativo, usando o [console do Amazon Mobile Analytics](#). Para obter mais informações sobre o uso do Mobile Analytics no console do Mobile Analytics, consulte o [Guia de usuário do Amazon Mobile Analytics](#).

Siga estas etapas para configurar a amostra antes da execução:

1. Selecione o objeto de jogo do `AmazonMobileAnalyticsSample`.
2. Especifique o ID do aplicativo (criado no [console do Amazon Mobile Analytics](#)) no campo "App Id".
3. Especifique o ID do grupo de identidades do Cognito (criado usando o [console do Amazon Cognito Console do](#)) no campo "Cognito Identity Pool Id".

4. Certifique-se de que suas funções autenticadas e não autenticadas tenham permissões para acessar o serviço Mobile Analytics. Para obter mais informações sobre a aplicação de políticas às Funções do IAM, consulte [Gerenciar funções](#).

Ao executar o aplicativo de amostra, lembre-se de que os eventos não podem ser transmitidos ao serviço de back-end imediatamente. Um segmento de fundo criará eventos de buffer localmente e os enviará em lotes para o back-end do Amazon Mobile Analytics em intervalos regulares (o valor padrão é de 60 segundos), para garantir que o desempenho do seu jogo não seja afetado negativamente. Devido ao processamento complexo que o Amazon Mobile Analytics realiza em seus dados, os eventos enviados e relatórios correspondentes podem não estar visíveis no console da AWS até 60 minutos após o envio inicial.

Para obter mais informações sobre os relatórios fornecidos pela Amazon Mobile Analytics, consulte [Métricas de relatório e de dispositivos móveis](#).

Amazon S3

O Amazon Simple Storage Service (Amazon S3) fornece aos desenvolvedores e às equipes de TI um armazenamento de objetos seguro, duradouro e altamente escalável e econômico. No Unity, você pode usar o S3 para armazenar, listar e recuperar imagens, vídeos, músicas e outros dados usados pelos seus jogos.

Para obter mais informações sobre o S3, consulte [Amazon S3](#) e [Conceitos básicos do S3](#).

Para obter mais informações sobre o uso do S3 nos aplicativos Unity, consulte o [Amazon Simple Storage Service \(S3\)](#).

Configurando o padrão de assinatura do S3

O padrão de assinatura do S3 será configurado conforme a seguir:

```
<s3 useSignatureVersion4="true" />
```

Este item será usado para especificar se você deve usar a assinatura versão 4 nas solicitações do S3.

Uso do exemplo do S3

No painel Project (Projeto), navegue até Assets (Ativos)/AWSSDK/examples (exemplos)/S3e, no lado direito do painel, selecione o cenário do S3Example para abri-lo. A amostra ilustra como listar buckets e objetos dentro de um bucket, publicar objetos em um bucket e fazer o download de objetos em um bucket. Siga estas etapas para configurar a amostra antes da execução:

1. Selecione o objeto de jogo do S3 no painel Hierarchy (Hierarquia).
2. No painel Inspector (Inspetor), insira os valores de S3BucketName SampleFileName. S3BucketName é o nome do bucket usado pela amostra e S3SampleFileName é o nome do arquivo do qual a amostra fará upload no bucket do S3 especificado.
3. Certifique-se de que suas funções autenticadas e não autenticadas tenham permissões para acessar o buckets do S3 em sua conta. Para obter mais informações sobre a aplicação de políticas às Funções do IAM, consulte [Gerenciar funções](#).

Para executar o exemplo, clique no botão de reprodução na parte superior da tela do editor. Quando o aplicativo for executado, ele exibirá uma série de botões:

- Get Objects – Recebe uma lista de todos os objetos em todos os buckets de sua conta da AWS.
- Get Buckets – Recebe uma lista de todos os buckets de sua conta da AWS.
- Post Object – Carrega um objeto em um bucket especificado do S3.
- Delete Object – Exclui todos os objetos de um bucket do S3 especificado.

O exemplo exibe o feedback na parte superior da tela do jogo.

Amazon Simple Notification Service

O Amazon Simple Notification Service é um serviço de notificação por push rápido, flexível e totalmente gerenciado, que permite enviar mensagens individuais ou encaminhá-las para um grande número de destinatários. Com o Amazon Simple Notification Service, é simples e econômico enviar notificações por push para usuários de dispositivos móveis, destinatários de e-mail ou até mesmo enviar mensagens a outros serviços distribuídos. Para saber os conceitos básicos do Amazon Simple Notification Service, consulte o [Amazon Simple Notification Service](#).

AWS Lambda

O AWS Lambda é um serviço de computação que executa seu código em resposta a solicitações ou eventos e gerencia automaticamente os recursos de computação para você, facilitando a construção de aplicativos que respondem rapidamente às novas informações. As funções do AWS Lambda podem ser chamadas diretamente do dispositivo móvel, da IoT e de aplicativos web e envia uma resposta de modo síncrono, facilitando a criação de back-ends escaláveis, seguros e altamente disponíveis para seus aplicativos móveis sem precisar prover nem gerenciar a infraestrutura. Para obter mais informações, consulte [AWS Lambda](#).

Amazon Cognito Identity

O que é o Amazon Cognito Identity?

Usando o Amazon Cognito Identity, é possível criar identidades exclusivas para os seus usuários e autentique-os para proteger o acesso aos seus recursos da AWS, como o Amazon S3 ou o Amazon DynamoDB. O Amazon Cognito Identity é compatível com provedores de identidade públicos, como o Amazon, o Facebook e o Google, ou qualquer provedor compatível com o OpenID Connect, além de identidades não autenticadas. O Cognito é compatível também com as identidades autenticadas pelos desenvolvedores, que permitem a você registrar e autenticar usuários por meio do processo de autenticação de back-end, sem deixar de usar a [Sincronização do Amazon Cognito](#) para sincronizar os dados do usuário e acessar os recursos da AWS.

Para obter mais informações sobre o Cognito Identity, consulte o [Guia do desenvolvedor do Amazon Cognito](#).

Para obter informações sobre a disponibilidade da região de autenticação do Cognito, consulte a [Disponibilidade de regiões do Amazon Cognito Identity](#).

Usando um provedor público para autenticar usuários

Para obter informações sobre como usar provedores públicos de identidade como o Amazon, Facebook, Twitter/Digits, ou o Google para autenticar usuários, consulte os [Provedores externos](#) no guia de Desenvolvedor do Amazon Cognito.

Uso de Identidades autenticadas pelo desenvolvedor

Para obter informações sobre as identidades autenticadas do desenvolvedor, consulte as [Identidades autenticadas pelo desenvolvedor](#) no Guia do desenvolvedor do Amazon Cognito.

Amazon Cognito Sync

O Cognito Sync é um serviço da AWS e uma biblioteca de clientes que permite a sincronização dos dados de usuário relacionados a aplicativo entre dispositivos. Você pode usar a API do Cognito Sync para sincronizar os dados do usuário entre dispositivos. Para usar o Cognito Sync no aplicativo, você deve incluir o AWS Mobile SDK para o Unity em seu projeto.

Para obter instruções sobre como integrar o Amazon Cognito Sync em seu aplicativo, consulte o [Guia do desenvolvedor do Amazon Cognito Sync](#).

Amazon Mobile Analytics

Com o uso do Amazon Mobile Analytics, é possível acompanhar os comportamentos dos clientes, agregar métricas, gerar visualizações de dados, além de identificar padrões significativos. Para obter mais informações sobre o Mobile Analytics, consulte [AWS Mobile Analytics](#).

Integração do Amazon Mobile Analytics

As seções abaixo explicam como integrar o Mobile Analytics ao aplicativo.

Criação de um aplicativo no console do Mobile Analytics

Acesse o [console do Amazon Mobile Analytics](#) e crie um aplicativo. Observe o valor appId, pois ele será necessário em um passo posterior.

Note

Para saber mais informações sobre como trabalhar no console, consulte o [Guia do usuário do Amazon Mobile Analytics](#).

Quando você estiver criando um aplicativo no console do Mobile Analytics, precisará especificar o ID do grupo de identidades do Cognito. Para criar um novo grupo de identidades do Cognito e gerar um ID, consulte [Guia do desenvolvedor do Cognito Identity](#).

Integração do Mobile Analytics ao aplicativo

Para acessar o Mobile Analytics no Unity, você precisará das seguintes instruções:

```
using Amazon.MobileAnalytics.MobileAnalyticsManager;  
using Amazon.CognitoIdentity;
```

A prática recomendada é usar o Amazon Cognito para fornecer ao aplicativo credenciais temporárias da AWS. Essas credenciais permitem que o aplicativo acessar os recursos da AWS. Para criar um provedor de credenciais, siga as instruções em [Amazon Cognito Identity](#).

Criação de uma instância do MobileAnalyticsManager com as seguintes informações:

- cognitoIdentityPoolId - O ID do grupo de identidades do Cognito para o aplicativo

- `cognitoRegion` – a região do grupo de identidades do Cognito, por exemplo, "RegionEndpoint.USEast1"
- `region` – a região do serviço Mobile Analytics, por exemplo, "RegionEndpoint.USEast1"
- `appId` – O valor gerado pelo console do Mobile Analytics quando você adiciona um aplicativo

Use o `MobileAnalyticsClientContextConfig` para inicializar uma instância

`MobileAnalyticsManager` conforme mostrado no trecho de código a seguir:

```
// Initialize the MobileAnalyticsManager
void Start()
{
    // ...
    analyticsManager = MobileAnalyticsManager.GetOrCreateInstance(
        new CognitoAWSCredentials(<cognitoIdentityPoolId>, <cognitoRegion>),
        <region>,
        <appId>);
    // ...
}
```

Note

O ID do aplicativo é gerado para você durante o assistente de criação de aplicativos. Os dois valores devem corresponder aos usados no console do Mobile Analytics.

O `appId` é usado para agrupar seus dados no console do Mobile Analytics. Para localizar o ID do aplicativo após criar o aplicativo no console do Mobile Analytics, navegue até o console do Mobile Analytics e clique no ícone de engrenagem no canto superior direito da tela. A página App Management será exibida, listando todos os aplicativos registrados e seus IDs de aplicativo.

Registro de eventos de monetização

O SDK para Unity fornece a classe `MonetizationEvent`, o que permite a você gerar eventos de monetização para rastrear as compras feitas em aplicativos móveis. O trecho de código a seguir mostra como criar um evento de monetização:

```
// Create the monetization event object
MonetizationEvent monetizationEvent = new MonetizationEvent();
```



```
// Set the details of the monetization event
monetizationEvent.Quantity = 3.0;
monetizationEvent.ItemPrice = 1.99;
monetizationEvent.ProductId = "ProductId123";
monetizationEvent.ItemPriceFormatted = "$1.99";
monetizationEvent.Store = "Your-App-Store";
monetizationEvent.TransactionId = "TransactionId123";
monetizationEvent.Currency = "USD";

// Record the monetization event
analyticsManager.RecordEvent(monetizationEvent);
```

Registro de eventos personalizados

O Mobile Analytics permite que você defina eventos personalizados. Os eventos personalizados são definidos inteiramente por você; eles ajudam a rastrear ações dos usuários específicas do seu aplicativo ou jogo. Para obter mais informações sobre eventos personalizados, consulte [Eventos personalizados](#). Neste exemplo, vamos supor que seu aplicativo seja um jogo e você queira registrar um evento quando um usuário conclui um nível. Crie um evento "LevelComplete", criando uma nova instância `AmazonMobileAnalyticsEvent`:

```
CustomEvent customEvent = new CustomEvent("LevelComplete");

// Add attributes
customEvent.AddAttribute("LevelName", "Level1");
customEvent.AddAttribute("CharacterClass", "Warrior");
customEvent.AddAttribute("Successful", "True");

// Add metrics
customEvent.AddMetric("Score", 12345);
customEvent.AddMetric("TimeInLevel", 64);

// Record the event
analyticsManager.RecordEvent(customEvent);
```

Registro de sessões

Quando o aplicativo perde o foco, é possível pausar a sessão. Em `OnApplicationFocus`, verifique se o aplicativo está sendo pausado. Em caso afirmativo, chame `PauseSession`; do contrário, chame `ResumeSession` conforme mostrado no trecho de código a seguir:

```
void OnApplicationFocus(bool focus)
{
    if(focus)
    {
        analyticsManager.ResumeSession();
    }
    else
    {
        analyticsManager.PauseSession();
    }
}
```

Por padrão, se o usuário tirar o foco do aplicativo por menos de cinco segundos e retornar novamente ao aplicativo, a sessão será retomada. Se o usuário tirar o foco do aplicativo por cinco segundos ou mais, uma nova sessão será criada. Essa configuração pode ser definida no arquivo `awsconfig.xml`. Para obter mais informações, consulte a seção [Configuração do Mobile Analytics em Conceitos básicos do AWS Mobile SDK para Unity](#).

Amazon Simple Storage Service (S3)

O Amazon Simple Storage Service (Amazon S3) fornece aos desenvolvedores e às equipes de TI um armazenamento de objetos seguro, duradouro e altamente escalável e econômico. Os desenvolvedores do Unity podem tirar proveito do S3 para carregar dinamicamente os ativos usados por seus jogos. Inicialmente, isso pode fazer com que o download dos jogos nas lojas de aplicativos seja mais rápido.

Para obter mais informações sobre o S3, consulte [Amazon S3](#).

Para obter informações sobre a disponibilidade de regiões do AWS S3, consulte [Disponibilidade de regiões do Serviço da AWS](#).

Note

Alguns exemplos deste documento supõem o uso de uma variável de caixa de texto chamada ResultText para exibir a saída de rastreamento.

Criação e configuração de um bucket do S3

O Amazon S3 armazena os recursos nos buckets do Amazon S3, contêineres de armazenamento na nuvem que residem em uma [região](#) específica. Cada bucket do Amazon S3 deve ter um nome globalmente exclusivo. Você pode usar o [console do Amazon S3](#) para criar um bucket.

Crie um bucket do S3

1. Faça login no [console do Amazon S3](#) e clique em Create Bucket (Criar bucket).
2. Insira um nome de bucket, selecione uma região e clique em Create (Criar).

Definir permissões para o S3

A política padrão da função do IAM concede ao aplicativo acesso ao Amazon Mobile Analytics e ao Amazon Cognito Sync. Para que o grupo de identidades do Cognito acesse o Amazon S3, você deve modificar as funções do grupo de identidades.

1. Acesse o [console do Identity and Access Management](#) e clique em Roles (Funções) no painel à esquerda.

2. Digite o nome do grupo de identidades na caixa de pesquisa. Duas funções serão listadas: uma para os usuários autenticados e outra para os usuários não autenticados.
3. Clique na função para usuários não autenticados (ela terá "unauth" anexado ao nome do grupo de identidades).
4. Clique em Create Role Policy (Criar política de função), selecione Policy Generator (Gerador de políticas) e, em seguida, clique em Select (Selecionar).
5. Na página Edit Permissions (Editar permissões), insira as configurações mostradas na imagem a seguir, substituindo o Nome de recurso da Amazon (ARN) pelo seu nome. O ARN do bucket do S3 é semelhante ao `arn:aws:s3:::examplebucket/*` e composto pela região na qual o bucket está localizado e pelo nome do bucket. As configurações mostradas abaixo concederão ao grupo de identidades acesso total a todas as ações do bucket especificado.

Edit Permissions

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [Overview of Policies](#) in Using AWS Identity and Access Management.

The screenshot shows the 'Edit Permissions' form in the AWS IAM console. It includes the following fields and options:

- Effect:** Radio buttons for 'Allow' (selected) and 'Deny'.
- AWS Service:** A dropdown menu with 'Amazon S3' selected.
- Actions:** A text box containing 'All Actions Selected'.
- Amazon Resource Name (ARN):** A text box containing 'arn:aws:s3:::examplebucket/*'.
- Buttons:** 'Add Conditions (optional)' (blue link) and 'Add Statement' (grey button).

1. Clique no botão Add Statement (Adicionar instrução) e em Next Step (Próxima etapa).
2. O assistente mostrará a você a configuração gerada. Clique em Apply Policy (Aplicar política).

Para obter mais informações sobre como conceder acesso ao S3, consulte [Conceder acesso a um bucket do Amazon S3](#).

Upload de arquivos no console

Para fazer upload de um arquivo de teste para o bucket:

1. No console do S3, na visualização do bucket, clique em Upload (Fazer upload).

2. Clique em Add Files (Adicionar arquivos) e selecione um arquivo de teste para fazer o upload. Neste tutorial, vamos supor que você esteja fazendo o upload de uma imagem chamada `myImage.jpg`.
3. Com a imagem de teste selecionada, clique em Start Upload (Iniciar upload).

(opcional) Configuração da versão de assinatura para solicitações do S3

Cada interação com o Amazon S3 é autenticada ou anônima. A AWS usa os algoritmos do Signature versão 4 ou Signature versão 2 para autenticar chamadas para o serviço.

Todas as novas regiões da AWS criadas após janeiro de 2014 são compatíveis apenas com o Signature versão 4. No entanto, muitas regiões mais antigas ainda oferecem suporte às solicitações do Signature versão 4 e do Signature versão 2.

Se o bucket estiver em uma das regiões que não oferecem suporte às solicitações do Signature versão 2, conforme listado [nesta página](#), será preciso definir a propriedade `AWSSignatureVersion4` como "true".

Para obter mais informações sobre as versões do AWS Signature, consulte [Solicitações de autenticação \(AWS Signature Version 4\)](#).

Criação do cliente Amazon S3

Para usar o Amazon S3, primeiro é necessário criar uma instância `AmazonS3Client` que faça referência à instância `CognitoAWSCredentials` criada anteriormente:

```
AmazonS3Client s3Client = new AmazonS3Client (credentials);
```

A classe `AmazonS3Client` é o ponto de entrada para a API de alto nível do S3.

Listagem de buckets

Para listar os buckets em uma conta da AWS, chame o método `AmazonS3Client.ListBucketsAsync` conforme mostrado no código de exemplo a seguir:

```
// ResultText is a label used for displaying status information
ResultText.text = "Fetching all the Buckets";
Client.ListBucketsAsync(new ListBucketsRequest(), (responseObject) =>
{
```

```
ResultText.text += "\n";
if (responseObject.Exception == null)
{
    ResultText.text += "Got Response \nPrinting now \n";
    responseObject.Response.Buckets.ForEach((s3b) =>
    {
        ResultText.text += string.Format("bucket = {0}, created date = {1} \n",
            s3b.BucketName, s3b.CreationDate);
    });
}
else
{
    ResultText.text += "Got Exception \n";
}
});
```

Listagem de objetos

Para listar todos os objetos em um buckets, chame o método

`AmazonS3Client.ListObjectsAsync` conforme mostrado no código de exemplo a seguir:

```
// ResultText is a label used for displaying status information
ResultText.text = "Fetching all the Objects from " + S3BucketName;

var request = new ListObjectsRequest()
{
    BucketName = S3BucketName
};

Client.ListObjectsAsync(request, (responseObject) =>
{
    ResultText.text += "\n";
    if (responseObject.Exception == null)
    {
        ResultText.text += "Got Response \nPrinting now \n";
        responseObject.Response.S3Objects.ForEach((o) =>
        {
            ResultText.text += string.Format("{0}\n", o.Key);
        });
    }
    else
    {
```

```
        ResultText.text += "Got Exception \n";
    }
});
```

Download de um objeto

Para fazer download de um objeto, crie uma `GetObjectRequest`, especificando o nome do bucket e a chave, e passe o objeto para uma chamada para `Client.GetObjectAsync`:

```
private void GetObject()
{
    ResultText.text = string.Format("fetching {0} from bucket {1}",
        SampleFileName, S3BucketName);
    Client.GetObjectAsync(S3BucketName, SampleFileName, (responseObj) =>
    {
        string data = null;
        var response = responseObj.Response;
        if (response.ResponseStream != null)
        {
            using (StreamReader reader = new StreamReader(response.ResponseStream))
            {
                data = reader.ReadToEnd();
            }

            ResultText.text += "\n";
            ResultText.text += data;
        }
    });
}
```

`GetObjectAsync` assume uma instância da `GetObjectRequest`, um retorno de chamada e uma instância `AsyncOptions`. O retorno de chamada deve ser do tipo: `AmazonServiceCallback<GetObjectRequest, GetObjectResponse>`. A instância `AsyncOptions` é opcional. Se especificada, ela determinará se o retorno de chamada será executado no thread principal.

Upload de um objeto

Para fazer upload de um objeto, grave o objeto em um fluxo, crie uma nova chave `PostObjectRequest` e especifique a chave, o nome do bucket e os dados do fluxo.

O AWS SDK para Unity usa o cliente WWW HTTP, que não é compatível com a operação HTTP PUT. Para fazer upload de um objeto para o bucket do S3, você precisará usar a publicação de navegador do S3, como mostrado abaixo.

```
public void PostObject(string fileName)
{
    ResultText.text = "Retrieving the file";

    var stream = new FileStream(Application.persistentDataPath +
    Path.DirectorySeparatorChar + fileName,
    FileMode.Open, FileAccess.Read, FileShare.Read);

    ResultText.text += "\nCreating request object";
    var request = new PostObjectRequest()
    {
        Bucket = S3BucketName,
        Key = fileName,
        InputStream = stream,
        CannedACL = S3CannedACL.Private
    };

    ResultText.text += "\nMaking HTTP post call";

    Client.PostObjectAsync(request, (responseObj) =>
    {
        if (responseObj.Exception == null)
        {
            ResultText.text += string.Format("\nobject {0} posted to bucket {1}",
            responseObj.Request.Key, responseObj.Request.Bucket);
        }
        else
        {
            ResultText.text += "\nException while posting the result object";
            ResultText.text += string.Format("\n received error {0}",
            responseObj.Response.HttpStatusCode.ToString());
        }
    });
}
```


Amazon DynamoDB

[O Amazon DynamoDB](#) é um serviço de banco de dados rápido, altamente disponível, altamente escalável, econômico e não relacional. O DynamoDB remove limitações de escalabilidade tradicionais sobre armazenamento de dados, mantendo, ao mesmo tempo, a baixa latência e o desempenho previsível. Para obter informações sobre o DynamoDB, consulte [Amazon DynamoDB](#).

O AWS Mobile SDK para Unity fornece uma biblioteca de alto nível para trabalhar em conjunto com o DynamoDB. Você também pode fazer solicitações diretamente com base na API de nível inferior do DynamoDB, mas, na maioria dos casos de uso, é recomendável o uso da biblioteca de nível superior. O AmazonDynamoDBClient é uma parte muito útil da biblioteca de nível superior. Usando essa classe, você pode realizar várias operações de criação, leitura, atualização e exclusão (CRUD) e executar consultas.

Note

Alguns exemplos deste documento supõem o uso de uma variável de caixa de texto chamada ResultText para exibir a saída de rastreamento.

Integração ao Amazon DynamoDB

Para usar o DynamoDB em um aplicativo Unity, você precisará adicionar o Unity SDK ao projeto. Se você ainda não tiver feito isso, [faça download do SDK para Unity](#) e siga as instruções em [Configurar o AWS Mobile SDK para Unity](#). É recomendável o uso do Amazon Cognito Identity para fornecer credenciais temporárias da AWS para seus aplicativos. Essas credenciais permitem que o aplicativo acessar recursos e serviços da AWS.

Para usar o DynamoDB em um aplicativo, defina as permissões corretas. A seguinte política do IAM permite que o usuário exclua, obtenha, insira, verifique e atualize itens em uma tabela específica do DynamoDB, que é identificada pelo [ARN](#):

```
{
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "dynamodb:DeleteItem",
      "dynamodb:GetItem",
```

```
        "dynamodb:PutItem",
        "dynamodb:Scan",
        "dynamodb:UpdateItem"
    ],
    "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
}]
}
```

Essa política deve ser aplicada às funções atribuídas ao grupo de identidades do Cognito, mas você precisará substituir o valor **Resource** pelo ARN correto da tabela do DynamoDB. O Cognito cria automaticamente uma função para o novo grupo de identidades, e você pode aplicar políticas a essa função no [console do IAM](#).

Você deve adicionar ou remover ações permitidas com base nas necessidades do seu aplicativo. Para saber mais sobre as políticas do IAM, consulte [Usar o IAM](#). Para saber mais sobre as políticas específicas do DynamoDB, consulte [Usar o IAM para controlar o acesso aos recursos do DynamoDB](#).

Criação de uma tabela do DynamoDB

Agora que as permissões e credenciais estão configuradas, vamos criar uma tabela do DynamoDB para o aplicativo. Para criar uma tabela, acesse o [console do DynamoDB](#) e siga estas etapas:

1. Clique em Create Table (Criar tabela).
2. Insira Bookstore como nome da tabela.
3. Selecione Hash como tipo de chave primária.
4. Selecione Number (Número) e insira id como o nome do atributo de hash. Clique em Continue.
5. Clique em Continue (Continuar) novamente para ignorar a adição de índices.
6. Defina a capacidade de leitura para 10 e a capacidade de gravação para 5. Clique em Continue.
7. Insira um e-mail de notificação e clique em Continue (Continuar) para criar alarmes de taxa de transferência.
8. Clique em Criar. O DynamoDB criará seu banco de dados.

Criação de um cliente DynamoDB

Para que o aplicativo interaja com uma tabela do DynamoDB, precisamos de um cliente. Podemos criar um cliente DynamoDB padrão da seguinte forma:

```
var credentials = new CognitoAWSCredentials(IDENTITY_POOL_ID, RegionEndpoint.USEast1);
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials);
DynamoDBContext Context = new DynamoDBContext(client);
```

A classe `AmazonDynamoDBClient` é o ponto de entrada para a API do DynamoDB. A classe fornece métodos de instância para criação, descrição, atualização e exclusão de tabelas, entre outras operações. O contexto adiciona uma camada adicional de abstração sobre o cliente e permite que você use funcionalidades adicionais, como o modelo de persistência de objeto.

Descrição de uma tabela

Para obtermos uma descrição da tabela do DynamoDB, podemos usar o seguinte código:

```
resultText.text += ("\n*** Retrieving table information ***\n");
var request = new DescribeTableRequest
{
    TableName = @"ProductCatalog"
};
Client.DescribeTableAsync(request, (result) =>
{
    if (result.Exception != null)
    {
        resultText.text += result.Exception.Message;
        Debug.Log(result.Exception);
        return;
    }
    var response = result.Response;
    TableDescription description = response.Table;
    resultText.text += ("Name: " + description.TableName + "\n");
    resultText.text += ("# of items: " + description.ItemCount + "\n");
    resultText.text += ("Provision Throughput (reads/sec): " +
        description.ProvisionedThroughput.ReadCapacityUnits + "\n");
    resultText.text += ("Provision Throughput (reads/sec): " +
        description.ProvisionedThroughput.WriteCapacityUnits + "\n");

    }, null);
}
```

Neste exemplo, criamos um cliente e um objeto `DescribeTableRequest`, atribuímos o nome da tabela à propriedade **TableName** e passamos o objeto de solicitação para o método `DescribeTableAsync`

no objeto `AmazonDynamoDBClient`. `DescribeTableAsync` também assume um delegado que será chamado quando a operação assíncrona for concluída.

Note

Todos os métodos assíncronos no `AmazonDynamoDBClient` assumem delegados que serão chamados quando a operação assíncrona for concluída.

Salvar um objeto

Para salvar um objeto no DynamoDB, use o método `SaveAsync<T>` do objeto `AmazonDynamoDBClient`, onde `T` é o tipo de objeto que você está salvando.

Chamamos o banco de dados "Bookstore" e, de acordo com esse tema, implementaremos um modelo de dados que registra atributos relacionados a livros. Aqui estão as classes que definem nosso modelo de dados.

```
[DynamoDBTable("ProductCatalog")]
public class Book
{
    [DynamoDBHashKey] // Hash key.
    public int Id { get; set; }
    [DynamoDBProperty]
    public string Title { get; set; }
    [DynamoDBProperty]
    public string ISBN { get; set; }
    [DynamoDBProperty("Authors")] // Multi-valued (set type) attribute.
    public List<string> BookAuthors { get; set; }
}
```

É claro que, para um aplicativo de livraria real, precisaríamos de campos adicionais para itens como autor e preço. A classe `Book` é decorada com o atributo `[DynamoDBTable]`; isso define a tabela de banco de dados na qual os objetos do tipo `Book` serão gravados. A chave de cada instância da classe `Book` é identificada por meio do atributo `[DynamoDBHashKey]`. As propriedades são identificadas com o atributo `[DynamoDBProperty]`; elas especificam a coluna da tabela de banco de dados na qual a propriedade será gravada. Com o modelo implementado, podemos gravar alguns métodos para criar, recuperar, atualizar e excluir os objetos `Book`.

Criação de um livro

```
private void PerformCreateOperation()
{
    Book myBook = new Book
    {
        Id = bookID,
        Title = "object persistence-AWS SDK for.NET SDK-Book 1001",
        ISBN = "111-1111111001",
        BookAuthors = new List<string> { "Author 1", "Author 2" },
    };

    // Save the book.
    Context.SaveAsync(myBook, (result)=>{
        if(result.Exception == null)
            resultText.text += @"book saved";
    });
}
```

Recuperação de um livro

```
private void RetrieveBook()
{
    this.displayMessage += "\n*** Load book**\n";
    Context.LoadAsync<Book>(bookID,
        (AmazonDynamoResult<Book> result) =>
    {
        if (result.Exception != null)
        {
            this.displayMessage += ("LoadAsync error" +result.Exception.Message);
            Debug.LogException(result.Exception);
            return;
        }
        _retrievedBook = result.Response;
        this.displayMessage += ("Retrieved Book: " +
            "\nId=" + _retrievedBook.Id +
            "\nTitle=" + _retrievedBook.Title +
            "\nISBN=" + _retrievedBook.ISBN);

        string authors = "";
        foreach(string author in _retrievedBook.BookAuthors)
            authors += author + ",";
    });
}
```

```

        this.displayMessage += "\nBookAuthor= "+ authors;
        this.displayMessage += ("\nDimensions= "+ _retrievedBook.Dimensions.Length + "
X " +
                                _retrievedBook.Dimensions.Height + " X " +
                                _retrievedBook.Dimensions.Thickness);

    }, null);
}

```

Atualização de um livro

```

private void PerformUpdateOperation()
{
    // Retrieve the book.
    Book bookRetrieved = null;
    Context.LoadAsync<Book>(bookID, (result)=>
    {
        if(result.Exception == null )
        {
            bookRetrieved = result.Result as Book;
            // Update few properties.
            bookRetrieved.ISBN = "222-2222221001";
            // Replace existing authors list with this
            bookRetrieved.BookAuthors = new List<string> { "Author 1", "Author x" };
            Context.SaveAsync<Book>(bookRetrieved, (res)=>
            {
                if(res.Exception == null)
                    resultText.text += ("\nBook updated");
            });
        }
    });
}

```

Exclusão de um livro

```

private void PerformDeleteOperation()
{
    // Delete the book.
    Context.DeleteAsync<Book>(bookID, (res)=>
    {
        if(res.Exception == null)

```

```
    {
        Context.LoadAsync<Book>(bookID,(result)=>
        {
            Book deletedBook = result.Result;
            if(deletedBook==null)
                resultText.text += ("\nBook is deleted");
        });
    }
});
}
```

Amazon Simple Notification Service

Usando o Amazon Simple Notification Service (SNS) e o Unity SDK, é possível gravar aplicativos para iOS e Android que possam receber notificações por push em dispositivos móveis. Para obter mais informações sobre o SNS, consulte [Amazon Simple Notification Service](#)

Este tópico o orientará durante a configuração do AWS SDK para o aplicativo de amostra do Unity, SNSExample.unity, para receber notificações por push em dispositivos móveis, por meio do Amazon SNS.

É possível criar aplicativos para iOS e Android, usando a amostra SNSExample.unity. As etapas de configuração são diferentes nas plataformas iOS e Android. Leia a seção abaixo, de acordo com a plataforma de destino.

Pré-requisitos

Os seguintes pré-requisitos são necessários para usar essa solução.

Definir permissões para o SNS

Ao criar um grupo de identidades do Cognito, duas funções do IAM são geradas:

- Cognito/_<Identity-Pool-Name>Auth_DefaultRole – a função do IAM referente aos usuários autenticados
- Cognito/_<Identity-Pool-Name>Unauth_DefaultRole – a função do IAM referente aos usuários não autenticados

Você deve adicionar permissões para acessar o serviço do Amazon SNS referente a essas funções. Para fazer isso:

1. Navegue até o [Console do IAM](#) e selecione a função do IAM a ser configurada.
2. Clique em Attach Policy (Anexar política), selecione a política AmazonSNSFullAccess policy e clique em Attach Policy (Anexar política).

Note

Usar o `AmazonSNSFullAccess` não é recomendado em um ambiente de produção. O usamos aqui para permitir que você inicie e avance rapidamente. Para obter mais informações sobre como especificar permissões para uma função do IAM, consulte [Visão geral das permissões de funções do IAM](#).

Pré-requisitos do iOS

- Associação ao Apple iOS Developer Program (Programa de desenvolvedor do iOS da Apple)
- Gerar uma identidade de assinatura
- Criar um perfil de provisionamento configurado para notificações por push

Você precisará executar o aplicativo em um dispositivo físico para receber notificações por push. Para executar seu aplicativo em um dispositivo, é preciso ter uma associação ao [Apple iOS Developer Program \(Programa de desenvolvedor do iOS da Apple\)](#). Quando você tiver uma associação, poderá usar o Xcode para gerar uma identidade de assinatura. Para obter mais informações, consulte a documentação [App Distribution Quick Start](#) da Apple. Depois, você precisará de um perfil de provisionamento configurado para receber notificações por push. Para obter mais informações, consulte a documentação [Configuring Push Notifications](#) da Apple.

Pré-requisitos do Android

- Instale o Android SDK
- Instale o JDK
- `android-support-v4.jar`
- `google-play-services.jar`

Configuração do Unity Sample App para iOS

Abra o editor do Unity e crie um novo projeto. Importe o AWS SDK para o pacote do Unity, selecionando Assets (Ativos)/Import Package (Importar pacote)/Custom Package (Pacote personalizado) e `aws-unity-sdk-sns-2.0.0.1.unitypackage`. Certifique-se de que todos os itens

da caixa de diálogo Importing Package (Importar pacote) estejam marcados e clique em Import (Importar).

Configuração do Unity

Execute as seguintes etapas para configurar o projeto do Unity:

1. No painel Project (Projeto), navegue até Assets (Ativos)/AWSSDK/examples (exemplos) e abra o cenário SNSExample.
2. No painel Hierarchy (Hierarquia), selecione SNSExample.
3. No painel Inspector (Inspetor) especifique seu ID do grupo de identidades do Cognito.
4. Observe que não há uma caixa de texto referente ao iOS Platform Application ARN (ARN do aplicativo para a plataforma iOS). Você gerará essas informações mais tarde.
5. Para adicionar o cenário atual ao desenvolvedor, selecione File (Arquivo)/Build Settings (Configurações do desenvolvedor), na caixa de diálogo Build Settings (Configurações do desenvolvedor), clique no botão Add Current (Adicionar atual) abaixo da caixa da lista Scenes in Build (Cenários do desenvolvedor).
6. Em Platform (Plataforma) selecione iOS e clique no botão Player Settings... (Configurações do Player...), no painel Inspector (Inspetor) do editor do Unity, clique no ícone do iPhone e role para baixo até a seção Identification (Identificação). Especifique um Bundle Identifier (Identificador de pacote).

Configuração do iOS

Execute as seguintes etapas para configurar a amostra, a fim de configurar as configurações específicas do iOS:

1. Em um navegador da web, acesse o [Apple Developer Member Center \(Centro de membros do desenvolvedor da Apple\)](#), clique em Certificates, Identifiers & Profiles (Certificados, Identificadores e Perfis).
2. Clique em Identifiers (Identificadores), em iOS Apps (Aplicativos iOS), clique no sinal de adição no canto superior direito da página da web para adicionar um novo ID do aplicativo iOS e insira uma descrição para o ID do aplicativo.
3. Role a tela para baixo até a seção Add ID Suffix (Adicionar sufixo ao ID) e selecione Explicit App ID (ID explícito do aplicativo) e insira o identificador do pacote.

4. Role para baixo até a seção App Services (Serviços do aplicativo) e, em seguida, selecione Push Notifications (Notificações por push).
5. Clique no botão Continue (Continuar).
6. Clique no botão Submit (Enviar).
7. Clique no botão Done (Concluído).
8. Selecione o ID do aplicativo que você acabou de criar e, em seguida, clique no botão Edit (Editar).
9. Role para baixo até a seção Push Notifications (Notificações por push).
10. Clique no botão Create Certificate (Criar certificado), em Development SSL Certificate (Desenvolvimento do certificado SSL).
11. Siga as instruções para criar a Certificate Signing Request (CSR - Solicitação de assinatura de certificado), fazer o upload da solicitação e fazer o download de um certificado SSL que será usado para se comunicar com o Apple Notification Service (APNS).
12. Voltando à página da web Certificates, Identifiers & Profiles (Certificados, identificadores e perfis), clique em All (Todos) em Provisioning Profiles (Perfis de Provisionamento).
13. Clique no botão de adição no canto superior direito para adicionar um novo perfil de provisionamento.
14. Selecione iOS App Development (Desenvolvimento de aplicativos iOS) e clique no botão Continue (Continuar).
15. Selecione o ID do aplicativo e clique no botão Continue (Continuar).
16. Selecione o certificado do desenvolvedor e clique no botão Continue (Continuar).
17. Selecione o dispositivo e clique no botão Continue (Continuar).
18. Insira um nome de perfil e clique no botão Generate (Gerar).
19. Faça o download e clique duas vezes no arquivo, para instalar o perfil de provisionamento.

Pode ser necessário atualizar os Perfis de provisionamento em Xcode após adicionar um novo. Em Xcode:

1. Selecione o item de menu Xcode/Preferences (Preferências).
2. Selecione a guia Accounts (Contas), selecione seu ID da Apple e clique em View Details (Visualizar detalhes).
3. Clique no botão Atualizar, no canto inferior esquerdo da caixa de diálogo, para atualizar seus perfis de provisionamento e certifique-se de que o novo perfil será exibido.

Configuração de SNS

1. Execute o aplicativo de acesso KeyChain, selecione My Certificates (Meus Certificados) na parte inferior esquerda da tela, clique com o botão direito do mouse no certificado SSL gerado para se conectar ao APNS e selecione Export (Exportar). Você será solicitado a especificar um nome para o arquivo e uma senha para proteger o certificado. O certificado será salvo em um arquivo P12.
2. Em um navegador da web, acesse o [Console do SNS](#) e clique em Applications (Aplicativos) no lado esquerdo da tela.
3. Clique em Create platform application (Criar aplicativo de plataforma) para criar um novo aplicativo para a plataforma SNS.
4. Insira um Application Name (Nome de aplicativo).
5. Selecione o Apple Push Notification Service (Serviço de notificações por push da Apple - APNS_SANDBOX) na Push notification platform (Plataforma de notificações por push).
6. Clique em Choose File (Selecionar arquivo) e selecione o arquivo P12 criado ao exportar o certificado SSL.
7. Insira a senha especificada durante a exportação do certificado SSL e clique em Load Credentials From File (Carregar credenciais a partir do arquivo).
8. Clique em Create platform application (Criar aplicativo de plataforma).
9. Selecione o aplicativo de plataforma que você acabou de criar e copie o Nome de região da Amazon (ARN) do aplicativo.
10. Volte para o projeto no editor do Unity, selecione SNSExample no painel Hierarchy (Hierarquia), no painel Inspector (Inspetor) e cole o Platform Application ARN na caixa de texto intitulada iOS Platform Application ARN (ARN do aplicativo para a plataforma iOS).
11. Selecione File (Arquivo)/Build Settings (Configurações do desenvolvedor) e clique no botão Build (Criar). Esta ação criará um projeto Xcode.

Uso do Xcode

1. Abra o projeto Xcode e selecione o projeto no Navegador do projeto.
2. Verifique se o identificador do pacote foi definido corretamente
3. Verifique se a Apple Developer Account (Conta de desenvolvedor da Apple) foi especificada na Team (Equipe) – isso é obrigatório para que a alteração do perfil de provisionamento ocorra.
4. Crie o projeto e execute-o em seu dispositivo.

5. Toque em Register for Notification (Registrar para receber notificações), toque em OK para permitir notificações. O aplicativo exibirá o token do dispositivo

No [Console do SNS](#), clique em Applications (Aplicativos), selecione o aplicativo de sua plataforma e clique em Create Platform Endpoint (Criar endpoint da plataforma), e insira o dispositivo de token exibido pelo aplicativo.

Neste momento, o aplicativo, o APNS e o NSN estarão integralmente configurados. É possível selecionar o aplicativo da plataforma, selecionar seu endpoint e clicar em Publish to endpoint (Publicar no endpoint) para enviar uma notificação por push para o dispositivo.

Amostra do Unity (iOS)

A amostra cria uma instância `CognitoAWSCredentials`, a fim de gerar credenciais temporárias e de escopo limitado, que permitem que o aplicativo convoque os serviços da AWS. Esta ação cria uma instância do `AmazonSimpleNotificationServiceClient`, que se comunica com o SNS. O aplicativo exibe dois botões, intitulados Register for Notification (Registrar para receber notificações) e Unregister (Cancelar registro).

ao tocar no botão Register for Notifications (Registrar para receber notificações), o método `RegisterDevice()` é chamado. O método `RegisterDevice()` chama `UnityEngine.iOS.NotificationServices.RegisterForNotifications`, que especifica quais tipos de notificação (alerta, som ou distintivo) serão usados. Ele faz também uma chamada assíncrona para APNS, visando obter um token do dispositivo. Como não existe um retorno de chamada definido, `CheckForDeviceToken` será chamado repetidamente (até 10 vezes) para verificar o token do dispositivo.

Quando um token for recuperado

`AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync()` será chamado, a fim de criar um endpoint para o aplicativo da plataforma do SNS.

A amostra agora será configurada para receber notificações por push. É possível navegar até o [Console do SNS](#), clique em Applications (Aplicativos) do lado esquerdo da página, selecione o aplicativo da plataforma, selecione um endpoint e clique em Publish to endpoint (Publicar no endpoint). Selecione o endpoint que será usado e clique em Publish to endpoint (Publicar no Endpoint). Digite uma mensagem de texto na caixa de texto e clique em Publish message (Publicar mensagem) para publicar uma mensagem.

Configuração do Unity Sample App para Android

Abra o editor do Unity e crie um novo projeto. Importe o AWS SDK para o pacote do Unity, selecionando Assets (Ativos)/Import Package (Importar pacote)/Custom Package (Pacote personalizado) e aws-unity-sdk-sns-2.0.0.1.unitypackage. Certifique-se de que todos os itens da caixa de diálogo Importing Package (Importar pacote) estejam marcados e clique em Import (Importar).

Configuração do Unity

Execute as seguintes etapas para configurar o projeto do Unity:

1. No painel Project (Projeto), navegue até Assets (Ativos)/AWSSDK/examples (exemplos) e abra o cenário SNSExample.
2. No painel Hierarchy (Hierarquia), selecione SNSExample.
3. No painel Inspector (Inspetor) especifique seu ID do grupo de identidades do Cognito.
4. Observe que não há uma caixa de texto referente ao Android Platform Application ARN (ARN do aplicativo para a plataforma Android) e ao Google Console Project ID (ID do projeto do console do Google), essas informações serão geradas mais tarde.
5. Para adicionar o cenário atual ao desenvolvedor, selecione File (Arquivo)/Build Settings (Configurações do desenvolvedor), na caixa de diálogo Build Settings (Configurações do desenvolvedor), clique no botão Add Current (Adicionar atual) abaixo da caixa da lista Scenes in Build (Cenários do desenvolvedor).
6. Em Platform (Plataforma), selecione Android e clique no botão Player Settings... (Configurações do Player...), no painel Inspector (Inspetor) do editor do Unity, clique no ícone do Android e role para baixo até a seção Identification (Identificação). Especifique um Bundle Identifier (Identificador de pacote).
7. Copie android-support-v4.jar e google-play-services.jar para o diretório Assets (Ativos)/Plugins/Android no painel Project (Projeto).

Para obter mais informações sobre onde encontrar android-support-v4.jar, consulte [Configuração da biblioteca de suporte ao Android](#). Para obter mais informações sobre como encontrar google-play-services.jar, consulte [APIs do Google para configuração do Android](#).

Configuração do Android

Primeiro, adicione um novo projeto API do Google:

1. Em um navegador da web, acesse o [Console de desenvolvedores do Google](#), clique em Create Project (Criar projeto).
2. Na caixa New Project (Novo projeto), insira um nome de projeto, anote o número do projeto (você precisará dele mais tarde) e clique em Create (Criar).

Em seguida, habilite o serviço Google Cloud Messaging (GCM) do seu projeto:

1. No Console de desenvolvedores do Google, seu novo projeto já deve estar selecionado. Se não estiver, selecione-o no menu suspenso, na parte superior da página.
2. Selecione APIs & auth (APIs e autorizações) na barra lateral à esquerda da página.
3. Na caixa de pesquisa, digite "Google Cloud Messaging for Android" e clique no link Google Cloud Messaging for Android (Google Cloud Messaging para Android) abaixo.
4. Clique em Enable API (Permitir API).

Por fim, obtenha uma chave para a API:

1. No Console de desenvolvedores do Google, selecione APIs & auth (APIs e autorizações) > Credentials (Credenciais).
2. Em Public API access (Acesso à API pública), clique em Create new key (Criar nova chave).
3. Na caixa de diálogo Create a new key (Criar uma nova chave), clique em Server key (Chave do servidor).
4. Na caixa de diálogo resultante, clique em Create (Criar) e copie a chave da API exibida.

Você usará a chave da API para realizar a autenticação posteriormente.

Configuração de SNS

1. Em um navegador da web, acesse o [Console do SNS](#) e clique em Applications (Aplicativos) no lado esquerdo da tela.
2. Clique em Create platform application (Criar aplicativo de plataforma) para criar um novo aplicativo para a plataforma SNS.

3. Insira um Application Name (Nome de aplicativo)
4. Selecione Google Cloud Messaging (GCM) na Push notification platform (Plataforma de notificações por push)
5. Cole a chave da API na caixa de texto intitulada API key (Chave da API).
6. Clique em Create platform application (Criar aplicativo para a plataforma)
7. Selecione o aplicativo de plataforma que você acabou de criar e copie o Nome de região da Amazon (ARN) do aplicativo.
8. Volte para o projeto no editor do Unity, selecione SNSExample no painel Hierarchy (Hierarquia), no painel Inspector (Inspector) e cole o ARN do aplicativo para a plataforma na caixa de texto intitulada Android Platform Application ARN (ARN do aplicativo para a plataforma Android) e o número do projeto na caixa de texto intitulada Google Console Project ID (ID do projeto do console do Google).
9. Conecte o dispositivo Android ao seu computador, selecione File (Arquivo)/Build Settings (Configurações do desenvolvedor) e clique em Build and Run (Criar e executar).

Unity Sample (Android)

A amostra cria uma instância `CognitoAWSCredentials`, a fim de gerar credenciais temporárias e de escopo limitado, que permitem que o aplicativo convoque os serviços da AWS. Esta ação cria uma instância do `AmazonSimpleNotificationServiceClient`, que se comunica com o SNS.

O aplicativo exibe dois botões, intitulados Register for Notification (Registrar para receber notificações) e Unregister (Cancelar registro). Ao tocar no botão Register for Notifications (Registrar para receber notificações), o método `RegisterDevice()` é chamado. O método `RegisterDevice()` chama `GCM.Register`, o que registra o aplicativo no GCM. GCM é uma classe definida no código de exemplo. Ele faz uma chamada assíncrona para registrar o aplicativo no GCM.

Quando houver o retorno de chamada, `AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync` será chamado para criar um endpoint para a plataforma SNS, visando o recebimento de mensagens do SNS.

A amostra agora será configurada para receber notificações por push. É possível navegar até o [Console do SNS](#), clique em Applications (Aplicativos) do lado esquerdo da página, selecione o aplicativo da plataforma, selecione um endpoint e clique em Publish to endpoint (Publicar no

endpoint). Selecione o endpoint que será usado e clique em Publish to endpoint (Publicar no Endpoint). Digite uma mensagem de texto na caixa de texto e clique em Publish message (Publicar mensagem) para publicar uma mensagem.

AWS Lambda

O AWS Lambda é um serviço de computação que executa seu código em resposta a solicitações ou eventos e gerencia automaticamente os recursos de computação para você, facilitando a construção de aplicativos que respondem rapidamente às novas informações. As funções do AWS Lambda podem ser chamadas diretamente do dispositivo móvel, da IoT e de aplicativos web e envia uma resposta de modo síncrono, facilitando a criação de back-ends escaláveis, seguros e altamente disponíveis para seus aplicativos móveis sem precisar prover nem gerenciar a infraestrutura.

O AWS Lambda pode executar suas funções do Lambda em resposta a um dos seguintes:

- Os eventos, como atualizações discretas (por exemplo, os eventos criados por objeto no Amazon S3 ou os alertas do CloudWatch) ou atualizações de streaming (por exemplo, clickstreams de sites ou saídas de dispositivos conectados).
- As entradas JSON ou os comandos HTTPS dos seus aplicativos personalizados.

O AWS Lambda executa seu código somente quando necessário e dimensiona automaticamente, desde algumas solicitações por dia a milhares por segundo. Com esses recursos, você pode usar o Lambda para criar facilmente acionadores de serviços da AWS, como o Amazon S3 e o Amazon DynamoDB, processar dados de streaming armazenados no Amazon Kinesis ou criar seu próprio back-end que opera na escala, no desempenho e na segurança da AWS.

Para saber mais sobre como o AWS Lambda funciona, consulte [AWS Lambda: Como funciona](#).

Permissões

Há dois tipos de permissões relacionadas às funções do Lambda:

- Permissões de execução: as permissões que sua função Lambda precisa para acessar outros recursos da AWS em sua conta. Para conceder essas permissões, basta você criar uma função do IAM, conhecida como função de execução.
- Permissões de invocação: as permissões que a fonte do evento precisa para se comunicar com sua função Lambda. Dependendo do modelo de invocação (modelo push ou pull), você pode conceder essas permissões usando a função de execução ou as políticas de recursos (a política de acesso associada à função do Lambda).

Configuração do projeto

Definir as permissões para o AWS Lambda

1. Abra o [console do AWS IAM](#).
2. Anexe essa política personalizada às suas funções, o que permite que o seu aplicativo faça chamadas para o AWS Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Criar uma nova função de execução

Essa função se aplica à função do Lambda que você criará na próxima etapa e determina quais recursos da AWS essa função poderá acessar.

1. Abra o [console do AWS IAM](#).
2. Clique em Roles (Funções).
3. Clique em Create New Roles (Criar novas funções).
4. Siga as instruções na tela para selecionar os serviços e políticas correspondentes que sua função do Lambda precisará de acesso. Por exemplo, se você deseja que sua função do Lambda crie um bucket do S3, sua política precisará de acesso de gravação ao S3.
5. Clique em Create Role (Criar função).

Criação de uma função no AWS Lambda

1. Abra o [console do AWS Lambda](#).

2. Clique em Create a Lambda function (Criar uma função Lambda).
3. Clique em Skip (Ignorar) para ignorar a criação de um esquema.
4. Configure sua própria função na próxima tela. Insira o nome da função, uma descrição e escolha o tempo de execução. Siga as instruções na tela com base no tempo de execução de sua escolha. Especifique as permissões de execução, atribuindo a função de execução recém-criada à sua função.
5. Quando terminar, clique em Next (Próximo).
6. Clique em Create Function (Criar função).

Criar um cliente do Lambda

```
var credentials = new CognitoAWSCredentials(IDENTITY_POOL_ID, RegionEndpoint.USEast1);
var Client = new AmazonLambdaClient(credentials, RegionEndpoint.USEast1);
```

Criar um objeto de solicitação

Crie um objeto de solicitação para especificar o tipo de invocação e o nome da função:

```
var request = new InvokeRequest()
{
    FunctionName = "hello-world",
    Payload = "{\"key1\" : \"Hello World!\"}",
    InvocationType = InvocationType.RequestResponse
};
```

Invocar a função do Lambda

Ligar para invocação, transmitindo o objeto de solicitação:

```
Client.InvokeAsync(request, (result) =>
{
    if (result.Exception == null)
    {
        Debug.Log(Encoding.ASCII.GetString(result.Response.Payload.ToArray()));
    }
    else
    {

```

```
        Debug.LogError(result.Exception);  
    }  
});
```

Solução de problemas

Devido às limitações da classe `Unity.WWW` usadas pelo AWS SDK para Unity, não são retornadas mensagens de erro detalhadas quando ocorre um problema durante a chamada de um serviço da AWS. Este tópico descreve algumas ideias para solucionar esses problemas.

Verificação da existência de permissões necessárias na função do IAM

Ao chamar os serviços da AWS, seu aplicativo usa uma identidade de um grupo de identidades do Cognito. Cada identidade do grupo é associada a uma função do IAM (Identity and Access Management). A função tem um ou mais arquivos de política associados a ela, que especificam a quais recursos da AWS os usuários atribuídos à função terão acesso. Por padrão, duas funções são criadas: uma para os usuários autenticados e outra para os usuários não autenticados. Você precisará modificar o arquivo de política existente ou associar um novo arquivo de política com as permissões necessárias ao aplicativo. Se o aplicativo permitir usuários autenticados e não autenticados, as duas funções deverão receber permissões para acessar os recursos da AWS necessários ao aplicativo.

O arquivo de política a seguir mostra como conceder acesso a um bucket do S3:

```
{
  "Statement": [
    {
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::MYBUCKETNAME/*",
      "Principal": "*"
    }
  ]
}
```

O arquivo de política a seguir mostra como conceder acesso a um banco de dados do DynamoDB:

```
{
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "dynamodb:DeleteItem",
      "dynamodb:GetItem",
      "dynamodb:PutItem",
      "dynamodb:Scan",
      "dynamodb:UpdateItem"
    ],
    "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
  }]
}
```

Para obter mais informações sobre como especificar políticas, consulte [Políticas do IAM](#).

Uso de um depurador de proxy HTTP

Se o serviço da AWS que o aplicativo está chamando tiver um endpoint HTTP ou HTTPS, você poderá usar um depurador de proxy HTTP/HTTPS para visualizar as solicitações e respostas e, assim, compreender melhor o que está acontecendo. Há uma série de depuradores de proxy HTTP disponíveis, como:

- [Charles](#) – um proxy de depuração da web para OSX
- [Fiddler](#) – um proxyfidd de depuração da web para Windows

Important

Há um problema conhecido do Cognito Credential Provider durante a execução do proxy de depuração da web Charles que impede o funcionamento correto do provedor de credenciais.

Charles e Fiddler exigem que algumas configurações possam exibir o tráfego criptografado por SSL. Leia a documentação dessas ferramentas para obter mais informações. Se você estiver usando um proxy de depuração da web que não possa ser configurado para exibir o tráfego criptografado, abra o arquivo `aws_endpoints.json` (localizado em `AWSUnitySDK/AWSCore/Resources`) e defina a tag HTTP referente ao serviço da AWS que você precisa depurar como `true`.