

Guida per gli sviluppatori

AWS SDK for Unity per dispositivi mobili



AWS SDK for Unity per dispositivi mobili: Guida per gli sviluppatori

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione ad alcun prodotto o servizio che non sia di Amazon, in alcun modo che possa causare confusione tra i clienti, né in alcun modo che possa denigrare o screditare Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

.....	vi
Cos'è il AWS Mobile SDK for Unity?	1
Guide e argomenti correlati	1
Contenuto di riferimento archiviato	1
Compatibilità	2
Scarica l'SDK per dispositivi mobili per Unity	2
Cosa è incluso nel Mobile SDK for Unity?	2
Configura l'SDK AWS Mobile per Unity	3
Prerequisiti	3
Passaggio 1: scarica l'SDK AWS Mobile per Unity	3
Fase 2: configurare l'SDK AWS Mobile per Unity	4
Crea una scena	4
Imposta la regione di servizio AWS predefinita	4
Imposta le informazioni di registrazione	4
Utilizzo del file link.xml	5
Passaggio 3: ottenere l'ID del pool di identità utilizzando Amazon Cognito	6
Fasi successive	7
Guida introduttiva all'SDK AWS Mobile per Unity	8
Amazon Cognito Identity	8
Amazon Cognito Sync	8
Utilizzo dell'esempio CognitoSyncManager	8
Dynamo DB	9
Utilizzo dell'esempio di DynamoDB	9
Mobile Analytics	10
Configurazione di Mobile Analytics	10
Utilizzo dell'esempio di Mobile Analytics	11
Simple Storage Service (Amazon S3)	12
Configurazione della firma predefinita di S3	12
Utilizzo dell'esempio S3	12
Amazon Simple Notification Service	13
AWS Lambda	13
Amazon Cognito Identity	14
Cos'è Amazon Cognito Identity?	14
Utilizzo di un provider pubblico per autenticare gli utenti	14

Utilizzo di identità autenticate dagli sviluppatori	14
Sincronizzazione con Amazon Cognito	15
Amazon Mobile Analytics	16
Integrazione di Amazon Mobile Analytics	16
Creare un'app nella Mobile Analytics Console	16
Integra Mobile Analytics nella tua app	16
Registra gli eventi di monetizzazione	17
Registra eventi personalizzati	18
Sessioni di registrazione	18
Servizio Amazon Simple Storage (S3)	20
Crea e configura un bucket S3	20
Creare un bucket S3	20
Imposta le autorizzazioni per S3	20
Carica file dalla console	21
(opzionale) Configura la versione di firma per le richieste S3	22
Crea il client Amazon S3	22
Creazione di un elenco di bucket	22
Elenco di oggetti	23
Download di un oggetto	24
Caricamento di un oggetto	24
Amazon DynamoDB	26
Integrazione di Amazon DynamoDB	26
Creare una tabella DynamoDB	27
Creare un client DynamoDB	27
Descrivi una tabella	28
Salvare un oggetto	29
Crea un libro	29
Recupera un libro	30
Aggiorna un libro	31
Eliminare un libro	31
Servizio di notifica Amazon Simple	33
Prerequisiti	3
Imposta le autorizzazioni per SNS	33
Prerequisiti iOS	34
Prerequisiti per Android	34
Configurazione dell'app Unity Sample per iOS	34

Configurazione Unity	34
Configurazione iOS	35
Configurazione SNS	36
Usare Xcode	37
Esempio di Unity (iOS)	37
Configurazione dell'app Unity Sample per Android	38
Configurazione Unity	38
Configurazione Android	39
Configurazione SNS	39
Esempio di Unity (Android)	40
AWS Lambda	41
Permissions	41
Configurazione del progetto	42
Imposta le autorizzazioni per AWS Lambda	42
Crea un nuovo ruolo di esecuzione	42
Creazione di una funzione in AWS Lambda	43
Creare un client Lambda	43
Crea un oggetto di richiesta	43
Invoca la tua funzione Lambda	43
Risoluzione dei problemi	45
Assicurati che IAM Role disponga delle autorizzazioni richieste	45
Utilizzo di un debugger proxy HTTP	46

Il AWS Mobile SDK for Unity è ora incluso in AWS SDK per .NET. Questa guida fa riferimento alla versione archiviata di Mobile SDK for Unity. Per ulteriori informazioni, consulta [Cos'è il AWS Mobile SDK for Unity?](#)

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Cos'è il AWS Mobile SDK for Unity?

Il AWS Mobile SDK for Unity è ora incluso in SDK per .NET. Per ulteriori informazioni, consulta la [Guida per gli sviluppatori di AWS SDK per .NET](#).

Questa guida non è più aggiornata: fa riferimento alla versione archiviata di Mobile SDK for Unity.

Guide e argomenti correlati

- Per lo sviluppo di app front-end e mobili, consigliamo di utilizzare. [AWS Amplify](#)
- Per considerazioni speciali sull'utilizzo di per le tue app Unity, consulta [Considerazioni speciali per il supporto di Unity nella Guida per gli sviluppatori](#). AWS SDK per .NET AWS SDK per .NET
- A scopo di riferimento, puoi trovare la versione archiviata di [AWS Mobile SDK for Unity](#) su. GitHub

Contenuto di riferimento archiviato

L'SDK mobile archiviato per Unity contiene un set di classi.NET che consentono ai giochi scritti con Unity di utilizzare AWS i servizi. Le applicazioni scritte con Mobile SDK for Unity possono essere eseguite su dispositivi iOS o Android.

AWS I servizi supportati includono:

- [Amazon Cognito](#)
- [Amazon DynamoDB](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Flusso di dati Amazon Kinesis](#)
- [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 \(Amazon S3\)](#)

Questi servizi consentono di autenticare gli utenti, salvare dati di giocatori e giochi, salvare oggetti nel cloud, inviare notifiche push e raccogliere e analizzare i dati di utilizzo.

Compatibilità

Mobile SDK for Unity v3 è compatibile con le versioni di Unity 4.6 e successive.

L'ultima versione di Mobile SDK for Unity ha introdotto miglioramenti che potrebbero richiedere la modifica del codice quando lo incorpori nel progetto. Per ulteriori informazioni su queste modifiche, consulta [Improvements in the AWS Mobile SDK for Unity](#) sul blog Front-End Web & Mobile.

Scarica l'SDK per dispositivi mobili per Unity

[Puoi anche scaricare Mobile SDK for Unity come file.zip qui.](#)

Cosa è incluso nel Mobile SDK for Unity?

Per l'elenco completo di NuGet pacchetti, esempi e altri file in Mobile SDK for Unity, [AWS SDK per .NET](#) vedi GitHub avanti.

Configura l'SDK AWS Mobile per Unity

Per iniziare a usare l'SDK AWS Mobile for Unity, puoi configurare l'SDK e iniziare a creare un nuovo progetto oppure puoi integrare l'SDK con un progetto esistente. Puoi anche clonare ed eseguire gli [esempi](#) per avere un'idea di come funziona l'SDK.

Prerequisiti

Prima di poter utilizzare l'SDK AWS Mobile per Unity, avrai bisogno di quanto segue:

- [Un account AWS](#)
- Unity versione 4.x o 5.x (Unity 4.6.4p4 o Unity 5.0.1p3 è necessario se si desidera scrivere applicazioni che funzionano su iOS a 64 bit)

Dopo aver completato i prerequisiti, dovrai fare quanto segue per iniziare:

1. Scarica l'SDK AWS Mobile per Unity.
2. Configura l'SDK AWS Mobile per Unity.
3. Ottieni credenziali AWS utilizzando Amazon Cognito.

Passaggio 1: scarica l'SDK AWS Mobile per Unity

Innanzitutto, [scarica l'SDK AWS Mobile per Unity](#). Ogni pacchetto nell'SDK è necessario per utilizzare il servizio AWS corrispondente in base al nome del pacchetto. Ad esempio, il pacchetto `aws-unity-sdk-dynamodb -2.1.0.0.unitypackage` viene utilizzato per chiamare il servizio AWS DynamoDB. Puoi importare tutti i pacchetti o solo quelli che intendi utilizzare.

1. Apri l'editor Unity e crea un nuovo progetto vuoto, usa le impostazioni predefinite.
2. Selezionate Risorse > Importa pacchetto > Pacchetto personalizzato.
3. Nella finestra di dialogo Importa pacchetto, accedete e selezionate i file.unitypackage che desiderate utilizzare.
4. Nella finestra di dialogo Importazione del pacchetto, assicuratevi che tutti gli elementi siano selezionati e fate clic su Importa.

Fase 2: configurare l'SDK AWS Mobile per Unity

Crea una scena

Quando lavori con l'SDK AWS Mobile per Unity, puoi iniziare includendo la seguente riga di codice nel metodo `Awake` o `Start` della tua classe di comportamento mono:

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

Crea la tua scena scegliendo Nuova scena dal menu File.

L'SDK AWS for Unity contiene classi client per ogni servizio AWS supportato. Questi client sono configurati utilizzando un file denominato `awsconfig.xml`. La sezione seguente descrive le impostazioni più comunemente utilizzate nel file `awsconfig.xml`. Per ulteriori informazioni su queste impostazioni, consulta [Unity SDK API Reference](#).

Imposta la regione di servizio AWS predefinita

Per configurare la regione predefinita per tutti i client di servizio:

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

Questo imposta la regione predefinita per tutti i client di servizio in Unity SDK. Questa impostazione può essere sovrascritta specificando esplicitamente la regione al momento della creazione di un'istanza del client di servizio, in questo modo:

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

Imposta le informazioni di registrazione

Le impostazioni di registrazione sono specificate come segue:

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

Questa impostazione viene utilizzata per configurare la registrazione in Unity. Quando si accede `UnityLogger`, il framework stampa internamente l'output nei Debug Logs. Se desideri

registrare le risposte HTTP, imposta il flag `LogResponses`: i valori possono essere `Always`, `Never` o `OnError`. È inoltre possibile registrare le metriche delle prestazioni per le richieste HTTP utilizzando la proprietà `LogMetrics`, il formato di registro può essere specificato `LogMetricsFormat` utilizzando la proprietà, i valori validi sono `JSON` o `standard`.

L'esempio seguente mostra le impostazioni più comunemente utilizzate nel file `awsconfig.xml`. Per ulteriori informazioni su impostazioni specifiche del servizio, consulta la sezione dedicata ai servizi di seguito:

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

Utilizzo del file `link.xml`

L'SDK utilizza la riflessione per i componenti specifici della piattaforma. Se si utilizza il backend di scripting IL2 CPP, `strip bytecode` è sempre abilitato su iOS, quindi è necessario disporre di un `link.xml` file nella radice dell'assembly con le seguenti voci:

```
<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>
```

Passaggio 3: ottenere l'ID del pool di identità utilizzando Amazon Cognito

Per utilizzare i servizi AWS nella tua applicazione mobile, devi ottenere l'ID del pool di identità utilizzando Amazon Cognito Identity. L'utilizzo di Amazon Cognito per ottenere l'ID Identity Pool consente alla tua app di accedere ai servizi AWS senza dover incorporare le tue credenziali private nell'applicazione. Ciò consente inoltre di impostare le autorizzazioni per controllare a quali servizi AWS hanno accesso gli utenti.

Per iniziare a usare Amazon Cognito, devi creare un pool di identità. Un pool di identità è un archivio di dati sull'identità dell'utente specifici per il tuo account. Ogni pool di identità dispone di ruoli IAM configurabili che consentono di specificare a quali servizi AWS possono accedere gli utenti dell'applicazione. In genere, uno sviluppatore utilizza un pool di identità per applicazione. Per ulteriori informazioni sui pool di identità, consulta la [Amazon Cognito Developer Guide](#).

Per creare un pool di identità per la tua applicazione:

1. Accedi alla [console Amazon Cognito](#) e fai clic su Crea nuovo pool di identità.
2. Inserisci un nome per il tuo pool di identità e seleziona la casella di controllo per abilitare l'accesso alle identità non autenticate. Fai clic su Crea pool per creare il tuo pool di identità.
3. Fai clic su Consenti per creare i due ruoli predefiniti associati al tuo pool di identità, uno per gli utenti non autenticati e uno per gli utenti autenticati. Questi ruoli predefiniti forniscono al tuo pool di identità l'accesso a Cognito Sync e Mobile Analytics.

La pagina successiva mostra il codice che crea un provider di credenziali in modo da poter integrare facilmente Cognito Identity con la tua applicazione Unity. Passi l'oggetto del provider di credenziali al costruttore del client AWS che stai utilizzando. Il codice sarà il seguente:

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

Fasi successive

- Inizia: leggi [Getting Started with the AWS Mobile SDK for Unity per](#) ottenere una panoramica più dettagliata dei servizi inclusi nell'SDK.
- Esegui le demo: visualizza le nostre [applicazioni Unity di esempio](#) che mostrano casi d'uso comuni. Per eseguire le app di esempio, configura l'SDK for Unity come descritto sopra, quindi segui le istruzioni contenute nei file README dei singoli esempi.
- Leggi il riferimento all'API: visualizza il [riferimento all'API](#) per l'SDK AWS Mobile per Unity.
- Fai domande: pubblica domande sui [forum dell'SDK AWS Mobile](#) o [apri un problema su Github](#).

Guida introduttiva all'SDK AWS Mobile per Unity

Questa pagina fornisce una panoramica di ogni servizio AWS nell'SDK AWS Mobile per Unity, oltre a istruzioni su come configurare gli esempi di Unity. È necessario completare tutte le istruzioni nella pagina [Configurazione dell'SDK AWS Mobile per Unity](#) prima di iniziare a utilizzare i servizi seguenti.

Amazon Cognito Identity

Tutte le chiamate effettuate verso AWS richiedono credenziali AWS. Invece di codificare le tue credenziali nelle tue app, ti consigliamo di utilizzare Amazon [Cognito Identity](#) per fornire le credenziali AWS alla tua applicazione. Segui le istruzioni in [Configurare l'SDK AWS Mobile per Unity per](#) ottenere le credenziali AWS tramite Amazon Cognito.

[Cognito consente inoltre di autenticare gli utenti utilizzando provider di accesso pubblici come Amazon, Facebook, Twitter e Google, nonché provider che supportano OpenID Connect.](#) Cognito funziona anche con utenti non autenticati. Cognito fornisce credenziali temporanee con diritti di accesso limitati specificate con un ruolo [Identity and Access Management](#) (IAM). Cognito viene configurato creando un nuovo pool di identità associato a un ruolo IAM. Il ruolo IAM specifica a cui i resources/services app può accedere.

Per iniziare a usare Cognito Identity, consulta la [Amazon Cognito Developer Guide](#).

Amazon Cognito Sync

[Cognito Sync](#) semplifica il salvataggio dei dati degli utenti finali, come le preferenze utente o lo stato del gioco, nel cloud AWS, in modo che possano essere resi disponibili agli utenti indipendentemente dal dispositivo che stanno utilizzando. Cognito può anche salvare questi dati localmente, permettendo alle app di funzionare anche quando non è disponibile una connessione Internet. Quando una connessione Internet diventa disponibile, le tue app possono sincronizzare i dati locali con il cloud.

Per iniziare a usare Cognito Sync, consulta la [Amazon Cognito Developer Guide](#).

Utilizzo dell'esempio CognitoSyncManager

Nel riquadro Progetto, accedete a Risorse AWSSDK//examples/CognitoSynce nella parte destra del riquadro selezionate la CognitoSynccena per aprirla.

Per eseguire l'esempio, fai clic sul pulsante play nella parte superiore della schermata dell'editor. Quando l'app è in esecuzione, vengono visualizzate alcune caselle di testo e pulsanti che consentono

di inserire alcune informazioni sul giocatore. Al di sotto, ci sono una serie di pulsanti che salvano le informazioni sul giocatore localmente, sincronizzano le informazioni sul giocatore locale con Cognito Cloud, aggiornano le informazioni sul giocatore da Cognito Cloud ed eliminano le informazioni sul giocatore locale. Premi ogni pulsante per eseguire un'operazione. L'esempio mostra il feedback nella parte superiore della schermata di gioco.

Per configurare l' `CognitoSyncManager` esempio, è necessario specificare un ID Cognito Identity Pool. Per specificare questo valore, nell'editor Unity, selezionalo `SyncManager` nel riquadro Hierarchy e inseriscilo nella casella di testo `IDENTITY_POOL_ID` nel riquadro Inspector.

Note

L' `CognitoSyncManager` esempio contiene codice che illustra come utilizzare il provider di identità di Facebook, cercate la macro «`USE_FACEBOOK_LOGIN`». Ciò richiede l'uso del Facebook SDK for Unity. Per ulteriori informazioni, consulta [Facebook SDK for Unity](#).

Dynamo DB

[Amazon DynamoDB](#) è un servizio di database non relazionale, conveniente, veloce e altamente scalabile e disponibile. DynamoDB rimuove le tradizionali limitazioni di scalabilità sullo storage dei dati mantenendo una bassa latenza e prestazioni prevedibili.

L'SDK AWS for Unity fornisce librerie di basso e alto livello per lavorare con DynamoDB. La libreria di alto livello include DynamoDB Object Mapper, che consente di mappare le classi lato client alle tabelle DynamoDB, eseguire varie operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) ed eseguire query. Utilizzando DynamoDB Object Mapper, puoi scrivere codice semplice e leggibile che archivia oggetti nel cloud.

[Per ulteriori informazioni su DynamoDB, consulta DynamoDB Developer Guide.](#)

[Per ulteriori informazioni sull'uso di Dynamo DB di Unity Applications, consulta Amazon DynamoDB.](#)

Utilizzo dell'esempio di DynamoDB

Nel riquadro Progetto, accedi a `Assets//examples AWSSDK/DynamoDB`. Questo esempio è composto dalle seguenti scene:

- `DynamoDBExample` : la scena iniziale dell'app

- `LowLevelDynamoDbExample` - esempio di utilizzo dell'API DynamoDB di basso livello
- `TableQueryAndScanExample` - esempio che mostra come eseguire le interrogazioni
- `HighLevelExample` - esempio di utilizzo di API DynamoDB di alto livello

Aggiungi queste scene alla build (nell'ordine in cui appaiono sopra) utilizzando la finestra di dialogo Build Settings (aperta selezionando File.Build Settings). Questo esempio crea quattro tabelle: Forum, ProductCatalog, Thread, Reply.

Per eseguire l'esempio, fai clic sul pulsante play nella parte superiore della schermata dell'editor. Quando l'app è in esecuzione, mostra una serie di pulsanti:

- Operazioni con tabelle di basso livello: illustra come creare, elencare, aggiornare, descrivere ed eliminare tabelle.
- Operazioni di interrogazione e scansione di livello intermedio: illustra come eseguire le interrogazioni.
- High Level Object Mapper: illustra come creare, aggiornare ed eliminare oggetti.

Mobile Analytics

Utilizzando [Amazon Mobile Analytics](#), puoi monitorare i comportamenti dei clienti, aggregare metriche, generare visualizzazioni di dati e identificare modelli significativi. L'SDK AWS for Unity fornisce l'integrazione con il servizio Amazon Mobile Analytics. Per informazioni su Mobile Analytics, consulta la [Guida per l'utente di Mobile Analytics](#). Per ulteriori informazioni sull'utilizzo di Mobile Analytics da Unity Applications, consulta [Amazon Mobile Analytics](#).

Configurazione di Mobile Analytics

Mobile Analytics definisce alcune impostazioni che possono essere configurate nel file `awsconfig.xml`:

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

- `SessionTimeout` - Questo è l'intervallo di tempo dopo che un'applicazione passa in background e quando la sessione può essere terminata.

- `maxDBSize` - Questa è la dimensione del database. SQLite Quando il database raggiunge la dimensione massima, tutti gli eventi aggiuntivi vengono eliminati.
- `dbWarningThreshold` - Questo è il limite alla dimensione del database che, una volta raggiunto, genererà dei log di avviso.
- `maxRequestSize` - Questa è la dimensione massima della richiesta in byte che deve essere trasmessa in una richiesta HTTP al servizio di analisi mobile.
- `allowUseDataRete`: un valore booleano che specifica se gli eventi della sessione vengono inviati sulla rete dati.

Utilizzo dell'esempio di Mobile Analytics

Nel riquadro Progetto, vai a Assets/AWSSDK/examples/Mobile Analytics e, nella parte destra del riquadro, seleziona la scena di esempio di Amazon Mobile Analytics per aprire la scena. Per utilizzare l'esempio, devi aggiungere la tua app utilizzando la [console Amazon Mobile Analytics](#). Per ulteriori informazioni sull'uso della console Mobile Analytics, consulta la [Guida per l'utente di Amazon Mobile Analytics](#).

Segui questi passaggi per configurare l'esempio prima dell'esecuzione:

1. Seleziona l'oggetto `AmazonMobileAnalyticsSample` del gioco.
2. Specificate il vostro ID app (creato nella [console Amazon Mobile Analytics](#)) nel campo «App Id».
3. Specificare l'ID del pool di identità di Cognito (creato utilizzando la [console Amazon Cognito all'indirizzo](#)) nel campo «Cognito Identity Pool Id».
4. Assicurati che i ruoli autenticati e non autenticati dispongano delle autorizzazioni per accedere al servizio Mobile Analytics. [Per ulteriori informazioni sull'applicazione delle policy ai ruoli IAM, consulta Managing Roles.](#)

Quando esegui l'applicazione di esempio, tieni presente che gli eventi potrebbero non essere trasmessi immediatamente al servizio di backend. Un thread in background memorizzerà gli eventi localmente e li invierà in batch al backend di Amazon Mobile Analytics a intervalli regolari (il valore predefinito è 60 secondi) per garantire che le prestazioni del gioco non subiscano ripercussioni negative. A causa della complessa elaborazione che Amazon Mobile Analytics esegue sui tuoi dati, gli eventi inviati e i report corrispondenti potrebbero non essere visibili nella console AWS fino a 60 minuti dopo l'invio iniziale.

Per ulteriori informazioni sui report forniti da Amazon Mobile Analytics, consulta [Report and Mobile Metrics](#).

Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) fornisce agli sviluppatori e ai team IT uno storage di oggetti sicuro, durevole e altamente scalabile. Da Unity puoi usare S3 per archiviare, elencare e recuperare immagini, video, musica e altri dati utilizzati dai tuoi giochi.

Per ulteriori informazioni su S3, consulta [Amazon S3](#) e [Guida introduttiva a S3](#).

Per ulteriori informazioni sull'utilizzo di S3 dalle applicazioni Unity, consulta [Amazon Simple Storage Service \(S3\)](#).

Configurazione della firma predefinita di S3

La firma S3 predefinita è configurata come segue:

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

Viene utilizzato per specificare se è necessario utilizzare la versione 4 della firma per le richieste S3.

Utilizzo dell'esempio S3

Nel riquadro Progetto, accedi a Assets/AWSSDK/examples/S3 e, nella parte destra del riquadro, seleziona la scena S3Example per aprire la scena. L'esempio illustra come elencare i bucket, elencare gli oggetti all'interno di un bucket, inserire oggetti in un bucket e scaricare oggetti da un bucket. Segui questi passaggi per configurare l'esempio prima dell'esecuzione:

1. Seleziona l'oggetto di gioco S3 nel riquadro Gerarchia.
2. Nel riquadro Inspector, inserisci i valori per BucketNameS3 e SampleFileName S3 BucketName è il nome del bucket utilizzato dall'esempio e S3 SampleFileName è il nome del file che l'esempio caricherà nel bucket S3 specificato.
3. Assicurati che i ruoli autenticati e non autenticati dispongano delle autorizzazioni per accedere ai bucket S3 del tuo account. [Per ulteriori informazioni sull'applicazione delle policy ai ruoli IAM, consulta Managing Roles](#).

Per eseguire l'esempio, fai clic sul pulsante play nella parte superiore della schermata dell'editor. Quando l'app è in esecuzione, mostra una serie di pulsanti:

- Get Objects: ottiene un elenco di tutti gli oggetti in tutti i bucket del tuo account AWS.
- Get Buckets: ottiene un elenco di tutti i bucket presenti nel tuo account AWS.
- Post Object: carica un oggetto in un bucket S3 specificato.
- Elimina oggetto: elimina tutti gli oggetti da un bucket S3 specificato.

L'esempio mostra il feedback nella parte superiore della schermata di gioco.

Amazon Simple Notification Service

Amazon Simple Notification Service è un servizio di notifica push veloce, flessibile e completamente gestito che ti consente di inviare messaggi singoli o di inviare messaggi a un gran numero di destinatari. Amazon Simple Notification Service semplifica ed economica l'invio di notifiche push a utenti di dispositivi mobili, destinatari di e-mail o persino l'invio di messaggi ad altri servizi distribuiti. Per iniziare a usare Amazon Simple Notification Service, consulta [Amazon Simple Notification Service](#).

AWS Lambda

AWS Lambda è un servizio di elaborazione che esegue il codice in risposta a richieste o eventi e gestisce automaticamente le risorse di calcolo per te, semplificando la creazione di applicazioni che rispondono rapidamente a nuove informazioni. Le funzioni di AWS Lambda possono essere richiamate direttamente da app mobili, IoT e Web e inviano una risposta in modo sincrono, semplificando la creazione di backend scalabili, sicuri e altamente disponibili per le app mobili senza la necessità di fornire o gestire l'infrastruttura. Per ulteriori informazioni, consulta [AWS Lambda](#).

Amazon Cognito Identity

Cos'è Amazon Cognito Identity?

Utilizzando Amazon Cognito Identity, puoi creare identità uniche per i tuoi utenti e autenticarli per un accesso sicuro alle tue risorse AWS come Amazon S3 o Amazon DynamoDB. Amazon Cognito Identity supporta provider di identità pubblici (Amazon, Facebook, Twitter/Digits, Google o qualsiasi provider compatibile con OpenID Connect) oltre a identità non autenticate. Cognito supporta anche identità autenticate dagli sviluppatori, che consentono di registrare e autenticare gli utenti utilizzando il proprio processo di autenticazione di backend, pur continuando a utilizzare Amazon Cognito Sync [per sincronizzare i dati degli utenti e accedere alle risorse AWS](#).

Per ulteriori informazioni su Cognito Identity, consulta la [Amazon Cognito Developer Guide](#).

Per informazioni sulla disponibilità della regione di autenticazione di Amazon Cognito, consulta [Amazon Cognito Identity Region Availability](#).

Utilizzo di un provider pubblico per autenticare gli utenti

Per informazioni sull'utilizzo di provider di identità pubblici come Amazon, Facebook, Twitter/Digits o Google per autenticare gli utenti, consulta i provider esterni [nella Amazon Cognito Developer Guide](#).

Utilizzo di identità autenticate dagli sviluppatori

Per informazioni sulle identità autenticate degli sviluppatori, consulta la [Developer Authenticated Identities](#) nella Amazon Cognito Developer Guide.

Sincronizzazione con Amazon Cognito

Cognito Sync è un servizio AWS e una libreria client che consente la sincronizzazione tra dispositivi dei dati utente relativi alle applicazioni. Puoi utilizzare l'API Cognito Sync per sincronizzare i dati degli utenti tra i dispositivi. Per utilizzare Cognito Sync nella tua app, devi includere l'SDK AWS Mobile per Unity nel tuo progetto.

Per istruzioni su come integrare Amazon Cognito Sync nella tua applicazione, consulta la [Amazon Cognito Sync Developer Guide](#).

Amazon Mobile Analytics

Utilizzando Amazon Mobile Analytics, puoi monitorare i comportamenti dei clienti, aggregare metriche, generare visualizzazioni di dati e identificare modelli significativi. Per informazioni su Mobile Analytics, consulta [AWS Mobile Analytics](#).

Integrazione di Amazon Mobile Analytics

Le sezioni seguenti spiegano come integrare Mobile Analytics con la tua app.

Creare un'app nella Mobile Analytics Console

Vai alla [console Amazon Mobile Analytics](#) e crea un'app. Annota il appId valore, poiché ti servirà in seguito.

Note

Per ulteriori informazioni su come lavorare nella console, consulta la [Amazon Mobile Analytics User Guide](#).

Quando crei un'app nella Mobile Analytics Console, dovrai specificare un ID Cognito Identity Pool. Per creare un nuovo pool di identità di Cognito e generare un ID, consulta la Guida per sviluppatori di [Cognito Identity](#).

Integra Mobile Analytics nella tua app

Per accedere a Mobile Analytics da Unity sono necessarie le seguenti istruzioni di utilizzo:

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

La best practice consiste nell'utilizzare Amazon Cognito per fornire credenziali AWS temporanee alla tua applicazione. che occorrono per accedere alle risorse AWS. Per creare un provider di credenziali, segui le istruzioni su [Amazon Cognito Identity](#).

Crea un' `MobileAnalyticsManager` istanza con le seguenti informazioni:

- `cognitoidentityPoolId`: l'ID del pool di identità di Cognito per la tua app

- **CognitoRegion**: la regione per il tuo pool di identità di Cognito, ad esempio «. RegionEndpoint USEast1»
- **region** - La regione per il servizio Mobile Analytics, ad esempio «RegionEndpoint. USEast1»
- **AppID**: il valore generato dalla Mobile Analytics Console quando aggiungi un'app

Utilizzate il `MobileAnalyticsClientContextConfig` per inizializzare un'**MobileAnalyticsManager** istanza come mostrato nel seguente frammento di codice:

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

Note

L'ID dell'app viene generato automaticamente durante la procedura guidata di creazione dell'app. Entrambi questi valori devono corrispondere a quelli della Mobile Analytics Console.

`appId` viene utilizzato per raggruppare i dati nella console Mobile Analytics. Per trovare l'ID dell'app dopo averla creata nella console Mobile Analytics, accedi alla Mobile Analytics Console e fai clic sull'icona a forma di ingranaggio nell'angolo in alto a destra dello schermo. Verrà visualizzata la pagina di gestione delle app che elenca tutte le app registrate e le relative app IDs.

Registra gli eventi di monetizzazione

L'SDK for Unity fornisce `MonetizationEvent` la classe che consente di generare eventi di monetizzazione per tenere traccia degli acquisti effettuati all'interno delle applicazioni mobili. Il seguente frammento di codice mostra come creare un evento di monetizzazione:

```
// 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);
```

Registra eventi personalizzati

Mobile Analytics consente di definire eventi personalizzati. Gli eventi personalizzati sono definiti interamente da te; ti aiutano a tenere traccia delle azioni degli utenti specifiche per la tua app o il tuo gioco. Per ulteriori informazioni sugli eventi personalizzati, consulta [Custom-Events](#). Per questo esempio, supponiamo che la tua app sia un gioco e desideri registrare un evento quando un utente completa un livello. Crea un evento «LevelComplete» creando una nuova `AmazonMobileAnalyticsEvent` istanza:

```
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);
```

Sessioni di registrazione

Quando l'applicazione perde il focus, puoi mettere in pausa la sessione.

`OnApplicationFocus` controlla se l'app è in pausa. In tal caso, chiama `PauseSession` altrimenti chiama `ResumeSession` come mostrato nel seguente frammento di codice:

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

Per impostazione predefinita, se l'utente distoglie l'attenzione dall'app per meno di 5 secondi e torna all'app, la sessione verrà ripresa. Se l'utente distoglie l'attenzione dall'app per 5 secondi o più, verrà creata una nuova sessione. Questa impostazione è configurabile nel file `awsconfig.xml`. Per ulteriori informazioni, consulta la sezione Configurazione di Mobile Analytics di [Getting Started with the AWS Mobile SDK for Unity](#).

Servizio Amazon Simple Storage (S3)

Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) fornisce agli sviluppatori e ai team IT uno storage di oggetti sicuro, durevole e altamente scalabile. Gli sviluppatori di Unity possono sfruttare S3 per caricare dinamicamente le risorse utilizzate dai loro giochi. Ciò può velocizzare il download iniziale dei giochi dagli app store.

Per ulteriori informazioni su S3, consulta [Amazon S3](#).

Per informazioni sulla disponibilità della regione AWS S3, consulta [AWS Service Region Availability](#).

Note

Alcuni esempi di questo documento presuppongono l'uso di una variabile di casella di testo chiamata ResultText per visualizzare l'output della traccia.

Crea e configura un bucket S3

[Amazon S3 archivia le tue risorse in bucket Amazon S3, contenitori di archiviazione cloud che risiedono in una regione specifica.](#) Ogni bucket Amazon S3 deve avere un nome univoco globale. Puoi usare la [console Amazon S3](#) per creare un bucket.

Creare un bucket S3

1. Accedi alla [console Amazon S3](#) e fai clic su Create Bucket.
2. Inserisci il nome del bucket, seleziona una regione e fai clic su Crea.

Imposta le autorizzazioni per S3

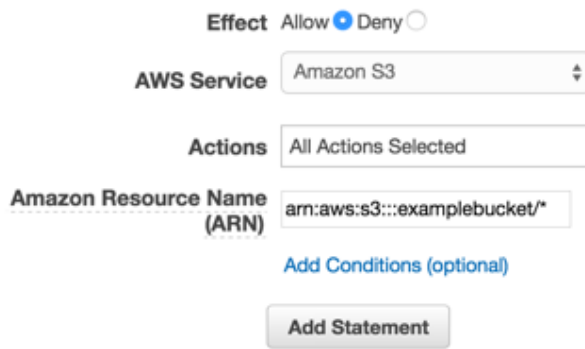
La policy di ruolo IAM predefinita consente all'applicazione di accedere ad Amazon Mobile Analytics e Amazon Cognito Sync. Affinché il pool di identità di Cognito possa accedere ad Amazon S3, è necessario modificare i ruoli del pool di identità.

1. Vai alla [Identity and Access Management Console](#) e fai clic su Ruoli nel riquadro a sinistra.
2. Digitare il nome del pool di identità nella casella di ricerca. Vengono elencati due ruoli, relativi, rispettivamente, agli utenti autenticati e non.

3. Fai clic sul ruolo per gli utenti non autenticati (verrà aggiunto unauth al nome del tuo pool di identità).
4. Fai clic su Crea politica sul ruolo, seleziona Policy Generator, quindi fai clic su Seleziona.
5. Nella pagina Modifica autorizzazioni, inserisci le impostazioni mostrate nell'immagine seguente, sostituendo l'Amazon Resource Name (ARN) con il tuo. L'ARN di un bucket S3 ha l'aspetto `arn:aws:s3:::examplebucket/*` ed è composto dalla regione in cui si trova il bucket e dal nome del bucket. Le impostazioni mostrate di seguito consentiranno al pool di identità di accedere a tutte le azioni per il bucket specificato.

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 AWS Policy Generator configuration interface. It includes the following elements:

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

1. Fai clic sul pulsante Aggiungi dichiarazione, quindi su Passaggio successivo.
2. La procedura guidata ti mostrerà la configurazione che hai generato. Fai clic su Applica politica.

Per ulteriori informazioni sulla concessione dell'accesso a S3, consulta [Garantire l'accesso a un bucket Amazon S3](#).

Carica file dalla console

Per caricare un file di test nel tuo bucket:

1. Nella console S3, nella visualizzazione bucket, fai clic su Carica.
2. Fai clic su Aggiungi file e seleziona un file di test da caricare. Per questo tutorial, supponiamo che tu stia caricando un'immagine chiamata `Image.jpg`.
3. Con l'immagine di prova selezionata, fai clic su Avvia caricamento.

(opzionale) Configura la versione di firma per le richieste S3

Ogni interazione con Amazon S3 è autenticata o anonima. AWS utilizza gli algoritmi Signature Version 4 o Signature Version 2 per autenticare le chiamate al servizio.

Tutte le nuove regioni AWS create dopo gennaio 2014 supportano solo la versione Signature 4. Tuttavia, molte regioni precedenti supportano ancora le richieste Signature Version 4 e Signature Version 2.

Se il tuo bucket si trova in una delle regioni che non supportano le richieste Signature Version 2 elencate in [questa pagina](#), devi impostare `AWSSignatureVersionProperty` a 4 impostata su «true».

Per ulteriori informazioni sulle versioni di AWS Signature, consulta [Authenticating Requests \(AWS Signature versione 4\)](#).

Crea il client Amazon S3

Per utilizzare Amazon S3, dobbiamo prima creare un'istanza `AmazonS3Client` che faccia riferimento all'istanza Cognito creata in precedenza: `AWSCredentials`

```
AmazonS3Client S3Client = new AmazonS3Client (credentials);
```

La `AmazonS3Client` classe è il punto di accesso all'API S3 di alto livello.

Creazione di un elenco di bucket

Per elencare i bucket in un account AWS, chiama il `AmazonS3Client.ListBucketsAsync` metodo come mostrato nel seguente codice di esempio:

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

Elenco di oggetti

Per elencare tutti gli oggetti in un bucket, chiama il `AmazonS3Client.ListObjectsAsync` metodo come mostrato nel seguente codice di esempio:

```
// 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 di un oggetto

Per scaricare un oggetto, create un file `GetObjectRequest`, specificando il nome e la chiave del bucket, e passate l'oggetto a una chiamata a `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` prende un'istanza di `GetObjectRequest`, una callback e un' `AsyncOptions` istanza. Il callback deve essere di tipo: `AmazonServiceCallback<GetObjectRequest, GetObjectResponse>` L' `AsyncOptions` istanza è facoltativa. Se specificato, determina se il callback verrà eseguito sul thread principale.

Caricamento di un oggetto

Per caricare un oggetto, scrivi l'oggetto in uno stream, creane uno nuovo `PostObjectRequest` e specifica la chiave, il nome del bucket e i dati dello stream.

L'SDK AWS for Unity utilizza il client HTTP WWW che non supporta l'operazione HTTP PUT. Per caricare un oggetto nel tuo bucket S3, devi utilizzare il Browser Post di S3, come mostrato di seguito.

```
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

[Amazon DynamoDB](#) è un servizio di database non relazionale, conveniente, veloce e altamente scalabile e disponibile. DynamoDB rimuove le tradizionali limitazioni di scalabilità sullo storage dei dati mantenendo una bassa latenza e prestazioni prevedibili. [Per informazioni su DynamoDB, consulta Amazon DynamoDB.](#)

L'SDK AWS Mobile per Unity fornisce una libreria di alto livello per lavorare con DynamoDB. Puoi anche effettuare richieste direttamente all'API DynamoDB di basso livello, ma per la maggior parte dei casi d'uso è consigliata la libreria di alto livello. AmazonDynamoDBClient È una parte particolarmente utile della libreria di alto livello. Utilizzando questa classe, è possibile eseguire varie operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) ed eseguire interrogazioni.

Note

Alcuni esempi di questo documento presuppongono l'uso di una variabile di casella di testo chiamata ResultText per visualizzare l'output della traccia.

Integrazione di Amazon DynamoDB

Per utilizzare DynamoDB in un'applicazione Unity, devi aggiungere l'SDK Unity al tuo progetto. Se non l'hai già fatto, [scarica l'SDK per Unity](#) e segui le istruzioni [in Configurare l'SDK AWS Mobile per Unity](#). Ti consigliamo di utilizzare Amazon Cognito Identity per fornire credenziali AWS temporanee per le tue applicazioni. Queste credenziali consentono all'app di accedere ai servizi e alle risorse AWS.

Per utilizzare DynamoDB in un'applicazione, è necessario impostare le autorizzazioni corrette. [La seguente policy IAM consente all'utente di eliminare, ottenere, inserire, scansionare e aggiornare gli elementi in una tabella DynamoDB specifica, identificata da 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"
}]
}
```

Questa politica deve essere applicata ai ruoli assegnati al pool di identità di Cognito, ma sarà necessario sostituire il **Resource** valore con l'ARN corretto per la tabella DynamoDB. [Cognito crea automaticamente un ruolo per il tuo nuovo pool di identità e puoi applicare policy a questo ruolo nella console IAM.](#)

Dovresti aggiungere o rimuovere le azioni consentite in base alle esigenze della tua app. Per ulteriori informazioni sulle policy IAM, consulta la sezione relativa all'[utilizzo di IAM](#). Per ulteriori informazioni sulle policy specifiche di DynamoDB, consulta [Using IAM to Control Access to DynamoDB Resources](#).

Creare una tabella DynamoDB

Ora che abbiamo configurato le autorizzazioni e le credenziali, creiamo una tabella DynamoDB per la nostra applicazione. Per creare una tabella, accedi alla console [DynamoDB e segui questi](#) passaggi:

1. Fare clic su Create Table (Crea tabella).
2. Immettete Bookstore come nome della tabella.
3. Selezionate Hash come tipo di chiave principale.
4. Seleziona Numero e inserisci il nome id dell'attributo hash. Fai clic su Continue (Continua).
5. Fate nuovamente clic su Continua per saltare l'aggiunta di indici.
6. Imposta la capacità di lettura 10 e la capacità di scrittura su. 5 Fai clic su Continue (Continua).
7. Inserisci un'e-mail di notifica e fai clic su Continua per creare allarmi di throughput.
8. Fai clic su Create (Crea). DynamoDB creerà il tuo database.

Creare un client DynamoDB

Affinché la nostra app interagisca con una tabella DynamoDB, abbiamo bisogno di un client. Possiamo creare un client DynamodDB predefinito nel modo seguente:

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

```
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials);
DynamoDBContext Context = new DynamoDBContext(client);
```

La `AmazonDynamoDBClient` classe è il punto di ingresso per l'API `DynamoDB`. La classe fornisce metodi di istanza per creare, descrivere, aggiornare ed eliminare tabelle, tra le altre operazioni. `Context` aggiunge un ulteriore livello di astrazione sul client e consente di utilizzare funzionalità aggiuntive come l'`Object Persistence Model`.

Descrivi una tabella

Per ottenere una descrizione della nostra tabella `DynamoDB`, possiamo usare il codice seguente:

```
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);
}
```

In questo esempio, creiamo un client e un `DescribeTableRequest` oggetto, assegniamo il nome della nostra tabella alla **`TableName`** proprietà e quindi passiamo l'oggetto della richiesta al `DescribeTableAsync` metodo sull'oggetto. `AmazonDynamoDBClient` `DescribeTableAsync` accetta anche un delegato che verrà chiamato al termine dell'operazione asincrona.

Note

Tutti i metodi asincroni di AmazonDynamo DBClient take delegati che vengono chiamati al termine dell'operazione asincrona.

Salvare un oggetto

Per salvare un oggetto in DynamoDB, utilizzate `SaveAsync <T>` il metodo dell'oggetto, dove `T` è `AmazonDynamo DBClient` il tipo di oggetto che state salvando.

Abbiamo chiamato il nostro database «Bookstore» e, in linea con questo tema, implementeremo un modello di dati che registra gli attributi relativi ai libri. Ecco le classi che definiscono il nostro modello di dati.

```
[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; }
}
```

Naturalmente, per una vera applicazione da libreria avremmo bisogno di campi aggiuntivi per cose come autore e prezzo. La classe `Book` è decorata con l'attributo `[DynamoDBTable]`, che definisce la tabella del database su cui verranno scritti gli oggetti di tipo `Book`. La chiave per ogni istanza della classe `Book` viene identificata utilizzando l'attributo `[Dynamo DBHash Key]`. Le proprietà sono identificate con l'attributo `[DynamoDBProperty]`, che specifica la colonna nella tabella del database in cui verrà scritta la proprietà. Una volta installato il modello, possiamo scrivere alcuni metodi per creare, recuperare, aggiornare ed eliminare oggetti `Book`.

Crea un libro

```
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";
    });
}

```

Recupera un libro

```

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);
}
```

Aggiorna un libro

```
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");
            });
        }
    });
}
```

Eliminare un libro

```
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");
    });
}
});
}
```

Servizio di notifica Amazon Simple

Utilizzando Amazon Simple Notification Service (SNS) e Unity SDK, puoi scrivere app iOS e Android in grado di ricevere notifiche push per dispositivi mobili. Per informazioni su SNS, consulta [Amazon Simple Notification Service](#).

Questo argomento ti illustrerà come configurare l'app di esempio AWS SDK for Unity SNSExample, .unity, per ricevere notifiche push mobili tramite Amazon SNS.

Puoi creare app iOS e Android utilizzando l' SNSExampleesempio.unity. I passaggi di configurazione sono diversi tra iOS e Android, leggi la sezione appropriata di seguito per la piattaforma a cui ti rivolgi.

Prerequisiti

Per utilizzare questa soluzione sono necessari i seguenti prerequisiti.

Imposta le autorizzazioni per SNS

Quando crei un pool di identità di Cognito, vengono generati due ruoli IAM:

- Cognito/_ <Identity-Pool-Name>Auth_DefaultRole : il ruolo IAM predefinito per gli utenti autenticati
- Cognito/_ <Identity-Pool-Name>Unauth_DefaultRole : il ruolo IAM predefinito per gli utenti non autenticati

È necessario aggiungere le autorizzazioni per accedere al servizio Amazon SNS a questi ruoli. Per farlo:

1. Accedi alla [console IAM](#) e seleziona il ruolo IAM da configurare.
2. Fai clic su **Allega politica**, seleziona la politica di Amazon SNSFull Access e fai clic su **Allega politica**.

Note

L'uso di Amazon SNSFull Access non è consigliato in un ambiente di produzione, lo utilizziamo qui per consentirti di essere subito operativo. Per ulteriori informazioni sulla

specificazione delle autorizzazioni per un ruolo IAM, consulta [Panoramica delle autorizzazioni dei ruoli IAM](#).

Prerequisiti iOS

- Iscrizione all'Apple iOS Developer Program
- Genera un'identità di firma
- Crea un profilo di provisioning configurato per le notifiche push

Dovrai eseguire l'app su un dispositivo fisico per ricevere le notifiche push. Per eseguire la tua app su un dispositivo devi essere iscritto all'[Apple iOS Developer Program Membership](#). Una volta sottoscritta l'iscrizione, puoi usare Xcode per generare un'identità di firma. Per ulteriori informazioni, consulta la documentazione [Quick Start di App Distribution](#) di Apple. Successivamente avrai bisogno di un profilo di provisioning configurato per le notifiche push. Per ulteriori informazioni, consulta la documentazione sulla [configurazione delle notifiche push](#) di Apple.

Prerequisiti per Android

- Installa l'SDK per Android
- Installa il JDK
- android-support-v4.jar
- google-play-services.jar

Configurazione dell'app Unity Sample per iOS

Apri l'editor Unity e crea un nuovo progetto. Importa il pacchetto AWS SDK for Unity selezionando Assets/Import Package/Custom Package e aws-unity-sdk-sns selezionando -2.0.0.1.unitypackage. Assicurati che tutti gli elementi nella finestra di dialogo Importing Package siano selezionati e fai clic su Importa.

Configurazione Unity

Esegui i seguenti passaggi per configurare il progetto Unity:

1. Nel riquadro Progetto, accedete ad Assets/AWSSDK/examples e aprite la SNSExample scena.

2. Nel riquadro Gerarchia, selezionate SNSExample.
3. Nel riquadro Inspector, specifica il tuo ID Cognito Identity Pool.
4. Nota che c'è una casella di testo per l'ARN dell'applicazione della piattaforma iOS, genererai tali informazioni in seguito.
5. Seleziona File/Build Settings, nella finestra di dialogo Build Settings, fai clic sul pulsante Aggiungi corrente sotto la casella di riepilogo Scenes in Build per aggiungere la scena corrente alla build.
6. In Piattaforma seleziona iOS e fai clic sul pulsante Impostazioni giocatore..., nel pannello Inspector dell'editor Unity, fai clic sull'icona iPhone e scorri verso il basso fino alla sezione Identificazione e specifica un identificatore del pacchetto.

Configurazione iOS

Effettuate le seguenti operazioni per configurare l'esempio per configurare impostazioni specifiche per iOS:

1. In un browser web, vai all'[Apple Developer Member Center](#) e fai clic su Certificati, identificatori e profili.
2. Fai clic su Identificatori in App iOS, fai clic sul pulsante più nell'angolo in alto a destra della pagina Web per aggiungere un nuovo ID app iOS e inserisci una descrizione dell'ID app.
3. Scorri verso il basso fino alla sezione Aggiungi suffisso ID e seleziona Explicit App ID e inserisci l'identificatore del pacchetto.
4. Scorri verso il basso fino alla sezione Servizi app e seleziona Notifiche push.
5. Fai clic sul pulsante Continua.
6. Fai clic sul pulsante Invia.
7. Fai clic sul pulsante Fine.
8. Seleziona l'ID app che hai appena creato, quindi fai clic sul pulsante Modifica.
9. Scorri verso il basso fino alla sezione Notifiche push.
10. Fai clic sul pulsante Crea certificato sotto Certificato SSL di sviluppo.
11. Segui le istruzioni per creare una richiesta di firma del certificato (CSR), carica la richiesta e scarica un certificato SSL che verrà utilizzato per comunicare con Apple Notification Service (APNS).
12. Torna alla pagina web Certificati, identificatori e profili, fai clic su Tutti nella sezione Provisioning Profiles.

13. Fai clic sul pulsante più nell'angolo in alto a destra per aggiungere un nuovo profilo di provisioning.
14. Seleziona Sviluppo app iOS e fai clic sul pulsante Continua.
15. Seleziona l'ID dell'app e fai clic sul pulsante Continua.
16. Seleziona il tuo certificato di sviluppatore e fai clic sul pulsante Continua.
17. Seleziona il tuo dispositivo e fai clic sul pulsante Continua.
18. Inserisci un nome di profilo e fai clic sul pulsante Genera.
19. Scaricate e fate doppio clic sul file di provisioning per installare il profilo di provisioning.

Potrebbe essere necessario aggiornare i profili di provisioning in Xcode dopo averne aggiunto uno nuovo. In Xcode:

1. Seleziona la voce di menu Xcode/Preferenze.
2. Seleziona la scheda Account, seleziona il tuo ID Apple e fai clic su Visualizza dettagli.
3. Fai clic sul pulsante di aggiornamento nell'angolo inferiore sinistro della finestra di dialogo per aggiornare i profili di provisioning e assicurarti che venga visualizzato il nuovo profilo.

Configurazione SNS

1. Esegui l'app di KeyChain accesso, seleziona I miei certificati nella parte inferiore sinistra dello schermo, fai clic con il pulsante destro del mouse sul certificato SSL che hai generato per connetterti ad APNS e seleziona Esporta. Ti verrà richiesto di specificare un nome per il file e una password per proteggere il certificato. Il certificato verrà salvato in un file P12.
2. In un browser Web, vai alla [console SNS](#) e fai clic su Applicazioni sul lato sinistro dello schermo.
3. Fai clic su Crea applicazione di piattaforma per creare una nuova applicazione di piattaforma SNS.
4. Inserisci un nome per l'applicazione.
5. Seleziona Apple Push Notification Service Sandbox (APNS_SANDBOX) per la piattaforma di notifica push.
6. Fai clic su Scegli file e seleziona il file P12 che hai creato quando hai esportato il certificato SSL.
7. Inserisci la password che hai specificato quando hai esportato il certificato SSL e fai clic su Carica credenziali dal file.
8. Fai clic su Crea applicazione di piattaforma.
9. Seleziona l'applicazione della piattaforma che hai appena creato e copia l'ARN dell'applicazione.

10. Torna al tuo progetto in Unity Editor, seleziona `SNSExample` nel riquadro Hierarchy, nel riquadro Inspector e incolla l'ARN dell'applicazione della piattaforma nella casella di testo denominata iOS Platform Application ARN.
11. Seleziona File/Build Settings e fai clic sul pulsante Build per creare un progetto Xcode.

Usare Xcode

1. Apri il progetto Xcode e seleziona il progetto nel Project Navigator.
2. Verifica che l'identificatore del pacchetto sia impostato correttamente
3. Verifica che il tuo account Apple Developer sia specificato nel Team: è necessario affinché il tuo Provisioning Profile abbia effetto.
4. Crea il progetto ed esegilo sul tuo dispositivo.
5. Tocca Registra per la notifica, tocca OK per consentire le notifiche, l'app mostrerà il token del tuo dispositivo

Nella [console SNS](#), fai clic su Applicazioni, seleziona l'applicazione della piattaforma e fai clic su Crea endpoint della piattaforma, quindi inserisci il token del dispositivo visualizzato dall'app.

A questo punto l'app, l'APNS e l'NSN sono completamente configurati. Puoi selezionare l'applicazione della piattaforma, selezionare l'endpoint e fare clic su Pubblica sull'endpoint per inviare una notifica push al dispositivo.

Esempio di Unity (iOS)

L'esempio crea un'AWS Credentials istanza Cognito per generare credenziali temporanee ad ambito limitato che consentono all'app di chiamare i servizi AWS. Crea anche un'istanza di AmazonSimpleNotificationServiceClient per comunicare con SNS. L'app mostra due pulsanti denominati Register for Notification e Annulla registrazione.

Quando si tocca il pulsante Registra notifiche, viene richiamato il `RegisterDevice()` metodo. `RegisterDevice()` chiama `UnityEngine.iOS.NotificationServices.RegisterForNotification` che specifica quali tipi di notifica (avviso, suono o badge) verranno utilizzati. Effettua anche una chiamata asincrona all'APNS per ottenere un token del dispositivo. Poiché non è stata definita alcuna callback, `CheckForDeviceToken` viene chiamata ripetutamente (fino a 10 volte) per verificare la presenza del token del dispositivo.

Quando viene recuperato un token

`AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync()` viene chiamato per creare un endpoint per l'applicazione della piattaforma SNS.

L'esempio è ora configurato per ricevere notifiche push. Puoi accedere alla [console SNS](#), fare clic su Applicazioni sul lato sinistro della pagina, selezionare l'applicazione della piattaforma, selezionare un endpoint e fare clic su Pubblica sull'endpoint. Seleziona l'endpoint da utilizzare e fai clic su Pubblica su endpoint. Digita un messaggio di testo nella casella di testo e fai clic su Pubblica messaggio per pubblicare un messaggio.

Configurazione dell'app Unity Sample per Android

Apri l'editor Unity e crea un nuovo progetto. Importa il pacchetto AWS SDK for Unity selezionando Assets/Import Package/Custom Package e `aws-unity-sdk-sns` selezionando `-2.0.0.1.unitypackage`. Assicurati che tutti gli elementi nella finestra di dialogo Importing Package siano selezionati e fai clic su Importa.

Configurazione Unity

Esegui i seguenti passaggi per configurare il progetto Unity:

1. Nel riquadro Progetto, accedete ad Assets/AWSSDK/examples e aprite la SNSExample scena.
2. Nel riquadro Gerarchia, selezionate SNSExample.
3. Nel riquadro Inspector, specifica il tuo ID Cognito Identity Pool.
4. Nota che c'è una casella di testo per l'ARN dell'applicazione della piattaforma Android e l'ID del progetto Google Console, genererai tali informazioni in seguito.
5. Seleziona File/Build Settings, nella finestra di dialogo Build Settings, fai clic sul pulsante Aggiungi corrente sotto la casella di riepilogo Scenes in Build per aggiungere la scena corrente alla build.
6. In Piattaforma seleziona Android e fai clic sul pulsante Impostazioni giocatore..., nel pannello Inspector dell'editor Unity, fai clic sull'icona Android e scorri verso il basso fino alla sezione Identificazione e specifica un identificatore del pacchetto.
7. Copia `android-support-v 4.jar` `google-play-services e.jar` nella directory Assets/Plugins/Android nel pannello Progetto.

Per ulteriori informazioni su dove trovare android-support-v 4.jar, consulta [Android Support Library Setup](#). Per ulteriori informazioni su come google-play-services trovare.jar, consulta [Configurazione di Google APIs per Android](#).

Configurazione Android

Per prima cosa aggiungi un nuovo progetto API di Google:

1. In un browser web, vai alla [Google Developers Console](#) e fai clic su Crea progetto.
2. Nella casella Nuovo progetto, inserisci il nome del progetto, prendi nota del numero del progetto (ti servirà in seguito) e fai clic su Crea.

Quindi, abilita il servizio Google Cloud Messaging (GCM) per il tuo progetto:

1. Nella Google Developers Console, il tuo nuovo progetto dovrebbe essere già selezionato, in caso contrario, selezionalo nel menu a discesa nella parte superiore della pagina.
2. Seleziona APIs & auth dalla barra laterale sul lato sinistro della pagina.
3. Nella casella di ricerca, digita «Google Cloud Messaging per Android» e fai clic sul link Google Cloud Messaging per Android di seguito.
4. Fai clic su Abilita API.

Finalmente ottieni una chiave API:

1. Nella Google Developers Console, seleziona APIs & auth > Credentials.
2. In Accesso pubblico all'API, fai clic su Crea nuova chiave.
3. Nella finestra di dialogo Crea una nuova chiave, fai clic su Chiave server.
4. Nella finestra di dialogo visualizzata, fai clic su Crea e copia la chiave API visualizzata.

Utilizzerai la chiave API per eseguire l'autenticazione in un secondo momento.

Configurazione SNS

1. In un browser Web, vai alla [console SNS](#) e fai clic su Applicazioni sul lato sinistro dello schermo.
2. Fai clic su Crea applicazione di piattaforma per creare una nuova applicazione di piattaforma SNS.
3. Inserisci il nome di un'applicazione

4. Seleziona Google Cloud Messaging (GCM) per la piattaforma di notifica push
5. Incolla la chiave API nella casella di testo denominata Chiave API.
6. Fai clic su Crea applicazione di piattaforma
7. Seleziona l'applicazione della piattaforma che hai appena creato e copia l'ARN dell'applicazione.
8. Torna al tuo progetto in Unity Editor, seleziona SNSExample nel riquadro Hierarchy, nel riquadro Inspector e incolla l'ARN dell'applicazione della piattaforma nella casella di testo denominata Android Platform Application ARN e il numero del progetto nella casella di testo Google Console Project ID.
9. Collega il tuo dispositivo Android al computer, seleziona File/Build Settings e fai clic su Build and Run.

Esempio di Unity (Android)

L'esempio crea un'AWS Credentials istanza Cognito per generare credenziali temporanee ad ambito limitato che consentono all'app di chiamare i servizi AWS. Crea anche un'istanza di AmazonSimpleNotificationServiceClient per comunicare con SNS.

L'app mostra due pulsanti denominati Register for Notification e Annulla registrazione. Quando si tocca il pulsante Registra notifiche, viene richiamato il RegisterDevice() metodo. RegisterDevice() chiama GCM.Register, che registrano l'app con GCM. GCM è una classe definita all'interno del codice di esempio. Effettua una chiamata asincrona per registrare l'app con GCM.

Quando viene chiamato il callback, AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync viene chiamato per creare un endpoint della piattaforma per ricevere messaggi SNS.

L'esempio è ora configurato per ricevere notifiche push. Puoi accedere alla [console SNS](#), fare clic su Applicazioni sul lato sinistro della pagina, selezionare l'applicazione della piattaforma, selezionare un endpoint e fare clic su Pubblica sull'endpoint. Seleziona l'endpoint da utilizzare e fai clic su Pubblica su endpoint. Digita un messaggio di testo nella casella di testo e fai clic su Pubblica messaggio per pubblicare un messaggio.

AWS Lambda

AWS Lambda è un servizio di elaborazione che esegue il codice in risposta a richieste o eventi e gestisce automaticamente le risorse di calcolo per te, semplificando la creazione di applicazioni che rispondono rapidamente a nuove informazioni. Le funzioni di AWS Lambda possono essere richiamate direttamente da app mobili, IoT e Web e inviano una risposta in modo sincrono, semplificando la creazione di backend scalabili, sicuri e altamente disponibili per le app mobili senza la necessità di fornire o gestire l'infrastruttura.

AWS Lambda può eseguire le tue funzioni Lambda in risposta a una delle seguenti situazioni:

- Eventi, come aggiornamenti discreti (ad esempio, eventi creati da oggetti in Amazon S3 o CloudWatch avvisi) o aggiornamenti in streaming (ad esempio, clickstream di siti Web o output da dispositivi collegati).
- Input JSON o comandi HTTPS dalle tue applicazioni personalizzate.

AWS Lambda esegue il codice solo quando è necessario e si dimensiona automaticamente, da poche richieste al giorno a migliaia al secondo. Con queste funzionalità, puoi usare Lambda per creare facilmente trigger per servizi AWS come Amazon S3 e Amazon DynamoDB, elaborare dati di streaming archiviati in Amazon Kinesis o creare il tuo back-end che opera su scala, prestazioni e sicurezza AWS.

Per ulteriori informazioni su come funziona AWS Lambda, consulta [AWS Lambda: How It Works](#).

Permissions

Esistono due tipi di autorizzazioni relative alle funzioni Lambda:

- Autorizzazioni di esecuzione: le autorizzazioni necessarie alla tua funzione Lambda per accedere ad altre risorse AWS nel tuo account. Concedi queste autorizzazioni creando un ruolo IAM, noto come ruolo di esecuzione.
- Autorizzazioni di chiamata: le autorizzazioni necessarie all'origine dell'evento per comunicare con la funzione Lambda. A seconda del modello di invocazione (modello push o pull), puoi concedere queste autorizzazioni utilizzando il ruolo di esecuzione o le policy delle risorse (la politica di accesso associata alla tua funzione Lambda).

Configurazione del progetto

Imposta le autorizzazioni per AWS Lambda

1. Apri la [console AWS IAM](#).
2. Allega questa policy personalizzata ai tuoi ruoli, che consente all'applicazione di effettuare chiamate ad AWS Lambda.

JSON

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

Crea un nuovo ruolo di esecuzione

Questo ruolo si applica alla funzione Lambda che creerai nella fase successiva e determina a quali risorse AWS può accedere quella funzione.

1. Apri la [console AWS IAM](#).
2. Fai clic su Ruoli.
3. Fai clic su Crea nuovi ruoli.
4. Segui le istruzioni sullo schermo per selezionare i servizi e le politiche corrispondenti a cui la tua funzione Lambda dovrà accedere. Ad esempio, se desideri che la funzione Lambda crei un bucket S3, la tua policy richiederà l'accesso in scrittura a S3.
5. Fai clic su Crea ruolo.

Creazione di una funzione in AWS Lambda

1. Apri la console [AWS Lambda](#).
2. Fate clic su Crea una funzione Lambda.
3. Fai clic su Ignora per saltare la creazione di un blueprint.
4. Configura la tua funzione nella schermata successiva. Inserisci il nome della funzione, una descrizione e scegli il tuo runtime. Segui le istruzioni sullo schermo in base al runtime scelto. Specificate i permessi di esecuzione assegnando il ruolo di esecuzione appena creato alla funzione.
5. Al termine, fai clic su Avanti.
6. Fai clic su Crea funzione.

Creare un client Lambda

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

Crea un oggetto di richiesta

Crea un oggetto di richiesta per specificare il tipo di invocazione e il nome della funzione:

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

Invoca la tua funzione Lambda

Chiama invoke, passando l'oggetto della richiesta:

```
Client.InvokeAsync(request, (result) =>  
{  
    if (result.Exception == null)  
    {
```

```
        Debug.Log(Encoding.ASCII.GetString(result.Response.Payload.ToArray()));
    }
    else
    {
        Debug.LogError(result.Exception);
    }
});
```

Risoluzione dei problemi

A causa delle limitazioni della classe `Unity.wwww` utilizzata dall'SDK AWS per Unity, non vengono restituiti messaggi di errore dettagliati quando si verifica un problema durante la chiamata a un servizio AWS. Questo argomento descrive alcune idee per la risoluzione di questi problemi.

Assicurati che IAM Role disponga delle autorizzazioni richieste

Quando chiama i servizi AWS, l'app utilizza un'identità proveniente da un pool di identità di Cognito. Ogni identità nel pool è associata a un ruolo IAM (Identity and Access Management). Al ruolo sono associati uno o più file di policy che specificano a quali risorse AWS hanno accesso gli utenti assegnati al ruolo. Per impostazione predefinita, vengono creati due ruoli, uno per gli utenti autenticati e uno per gli utenti non autenticati. Dovrai modificare il file di policy esistente o associare un nuovo file di policy alle autorizzazioni richieste dall'app. Se la tua app consente sia utenti autenticati che non autenticati, a entrambi i ruoli devono essere concesse le autorizzazioni per accedere alle risorse AWS di cui l'app ha bisogno.

Il seguente file di policy mostra come concedere l'accesso a un bucket S3:

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

Il seguente file di policy mostra come concedere l'accesso a un database 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"
}]
}
```

[Per ulteriori informazioni sulla specificazione delle policy, consulta IAM Policies.](#)

Utilizzo di un debugger proxy HTTP

Se il servizio AWS che la tua app sta chiamando ha un endpoint HTTP o HTTPS, puoi utilizzare un debugger HTTP/HTTPS proxy per visualizzare le richieste e le risposte e ottenere maggiori informazioni su ciò che sta accadendo. Sono disponibili diversi debugger proxy HTTP, come:

- [Charles](#): un proxy di debug web per OSX
- [Fiddler](#): un proxy di debug web per Windows

Important

Esiste un problema noto con il provider di credenziali Cognito durante l'esecuzione del proxy di debug web Charles che impedisce al provider di credenziali di funzionare correttamente.

Sia Charles che Fiddler richiedono alcune configurazioni per poter visualizzare il traffico crittografato SSL. Per ulteriori informazioni, leggi la documentazione di questi strumenti. Se utilizzi un proxy di debug web che non può essere configurato per visualizzare traffico crittografato, apri il file `aws_endpoints_json` (che si trova in) `AWSUnity SDK/AWSCore/Resources` e imposta il tag HTTP per il servizio AWS di cui devi eseguire il debug su `true`