

Guida per gli sviluppatori

AWSMobile SDK for Unity



AWSMobile SDK for Unity: Guida per gli sviluppatori

Copyright © 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

.....	vi
Che cos'è ?AWSMobile SDK for Unity?	1
Guide e argomenti correlati	1
Contenuti di riferimento archiviati	1
Compatibilità	2
Scarica il kit SDK mobile per Unity	2
Cosa è incluso nell'SDK mobile per Unity?	2
Configurazione dell'SDK AWS Mobile per Unity	3
Prerequisiti	3
Fase 1: Scarica l'SDK AWS Mobile per Unity	3
Fase 2: Configurazione dell'SDK AWS Mobile per Unity	4
Crea una scena	4
Impostazione della regione di servizio AWS predefinita	4
Imposta le informazioni di registrazione	4
Utilizzo del file link.xml	5
Fase 3: Ottieni l'ID del pool di identità utilizzando Amazon Cognito	6
Fasi successive	7
Nozioni di base su SDK AWS Mobile per Unity	8
Amazon Cognito Identity	8
Amazon Cognito Sync	8
Utilizzo diCognitoSyncEsempio di manager	8
Dynamo DB	9
Utilizzo dell'esempio DynamoDB	10
Mobile Analytics	10
Configurazione di Mobile Analytics	10
Utilizzo dell'esempio Mobile Analytics	11
Amazon S3	12
Configurazione della firma predefinita S3	12
Utilizzo del campione S3	12
Amazon Simple Notification Service	13
AWS Lambda	13
Amazon Cognito Identity	14
Che cos'è Amazon Cognito Identity?	14
Utilizzo di un provider pubblico per autenticare gli utenti	14

Utilizzo di identità non autenticate per gli sviluppatori	14
Amazon Cognito Sync	15
Amazon Mobile Analytics	16
Integrazione di Amazon Mobile Analytics	16
Crea un'app nella console di Mobile Analytics	16
Integra Mobile Analytics nella tua app	16
Registrare gli eventi di monetizzazione	17
Registrare eventi personalizzati	18
Sessioni di registrazione	18
Amazon Simple Storage Service (S3)	20
Creare e configurare un bucket S3	20
Creare un bucket S3	20
Impostazione delle autorizzazioni per S3	20
Carica file dalla console	21
(facoltativo) Configurare la versione Signature per le richieste S3	22
Creare 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
Creazione di una tabella DynamoDB	27
Creazione di un client DynamoDB	27
Descrizione di una tabella	28
Salva un oggetto	29
Creazione di un libro	29
Recupera un libro	30
Aggiornamento di un libro	31
Eliminazione di un libro	31
Amazon Simple Notification Service	33
Prerequisiti	3
Impostazione delle autorizzazioni per SNS	33
Prerequisiti iOS	34
Prerequisiti Android	34
Configurazione dell'app Unity Sample per iOS	34

Configurazione delle	35
Configurazione iOS	35
Configurazione SNS	36
Utilizzo di Xcode	37
Esempio Unity (iOS)	37
Configurazione dell'app Unity Sample per Android	38
Configurazione delle	38
Configurazione Android	39
Configurazione SNS	40
Esempio Unity (Android)	40
AWS Lambda	42
Autorizzazioni	42
Configurazione progetto	43
Autorizzazioni impostate per AWS Lambda	43
Creare un nuovo ruolo di esecuzione	43
Creazione di una funzione in AWS Lambda	43
Creare un client Lambda	44
Creare un oggetto di richiesta	44
Richiamare la funzione Lambda	44
Risoluzione dei problemi	46
Assicurati che il ruolo IAM abbia le autorizzazioni richieste	46
Utilizzo di un debugger proxy HTTP	47

LaAWSMobile SDK for Unity è ora incluso nel kit SDK MobileAWS SDK for .NET. Questa guida fa riferimento alla versione archiviata dell'SDK Mobile per Unity. Per ulteriori informazioni, consulta [Che cos'è ?AWSMobile 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.

Che cos'è ?AWSMobile SDK for Unity?

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

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

Guide e argomenti correlati

- Per lo sviluppo di app front-end e per dispositivi mobili, si consiglia di utilizzare[AWS Amplify](#).
- Per considerazioni speciali sull'utilizzo delAWS SDK for .NETper le tue app Unity, vedi[Considerazioni speciali per il supporto Unity](#)nellaAWS SDK for .NETGuida per gli sviluppatori.
- A scopo di riferimento, è possibile trovare la versione archiviata del[AWSMobile SDK for Unity](#)sulGitHub.

Contenuti di riferimento archiviati

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

supportateAWSi servizi includono:

- [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 \(Amazon S3\)](#)

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

Compatibilità

Mobile SDK for Unity v3 è compatibile con Unity versione 4.6 e successive.

L'ultima versione di Mobile SDK for Unity ha introdotto miglioramenti che potrebbero richiedere di modificare il codice quando si integra nel progetto. Per ulteriori informazioni su queste modifiche, consulta [Miglioramenti della AWS Mobile SDK for Unity](#) sul blog Front-End Web & Mobile.

Scarica il kit SDK mobile per Unity

È inoltre possibile scaricare Mobile SDK for Unity come file.zip [qui](#).

Cosa è incluso nell'SDK mobile per Unity?

Per l'elenco completo delle NuGet pacchetti, esempi e altri file nel Mobile SDK for Unity, vedere [AWS SDK for .NET](#) sul GitHub.

Configurazione dell'SDK AWS Mobile per Unity

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

Prerequisiti

Prima di poter utilizzare l'SDK AWS Mobile per Unity, è necessario avere a disposizione 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 eseguite su iOS a 64 bit)

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

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

Fase 1: Scarica l'SDK AWS Mobile per Unity

Prima di tutto [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, le ricette `aws-unity-sdkll` pacchetto `-dynamodb-2.1.0.0.unitypackage` viene utilizzato per chiamare il servizio AWS DynamoDB. È possibile 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. Seleziona `Asset > Pacchetto di importazione > Package Custom`.
3. Nella `ImportaPackage` (Pacchetto), passare a e selezionare i file `unitypackage` che si desidera utilizzare.
4. Nella `Importazione` di finestra di dialogo pacchetto, assicurati che tutti gli elementi siano selezionati e fai clic `Importa`.

Fase 2: Configurazione dell'SDK AWS Mobile per Unity

Crea una scena

Quando si lavora con AWS SDK for Unity, è possibile iniziare a includere la seguente riga di codice nella `StartoAwake` metodo della tua classe di comportamento mono:

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

Crea la tua scena scegliendo `Nuova scena` dagli `File` menu.

L'SDK AWS per Unity contiene classi client per ogni servizio AWS supportato. Questi client sono configurati utilizzando un file denominato `awsconfig.xml`. Nella sezione seguente vengono descritte le impostazioni più comunemente utilizzate nel `awsconfig.xml` file. Per ulteriori informazioni su queste impostazioni, consulta [il `Informazioni di riferimento delle API Unity SDK`](#).

Impostazione della regione di servizio AWS predefinita

Per configurare l'area predefinita per tutti i client di servizio:

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

Imposta l'area predefinita per tutti i client di servizio in Unity SDK. Questa impostazione può essere ignorata 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 accedi a `UnityLogger`, il framework stampa internamente l'output nei registri di debug. Se si desidera

registrare le risposte HTTP, impostare il flag `LogResponses`: i valori possono essere `Sempre`, `Mai` o `OnError`. È inoltre possibile registrare le metriche delle prestazioni per le richieste HTTP utilizzando la proprietà `LogMetrics`, il formato di log può essere specificato utilizzando `LogMetricsProprietàFormat`, i valori validi sono `JSON` o `standard`.

Nell'esempio seguente vengono mostrate le impostazioni più comunemente utilizzate nel file `awsconfig.xml`. Per ulteriori informazioni sulle impostazioni specifiche del servizio, vedere la sezione del servizio 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 componenti specifici della piattaforma. Se si utilizza il backend di scripting `IL2CPP`, `strip bytecode` è sempre abilitato su iOS, quindi è necessario avere a disposizione `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>
```

Fase 3: Ottieni 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 del pool di identità consente alla tua app di accedere ai servizi AWS senza dover incorporare le credenziali private nella tua applicazione. Ciò consente inoltre di impostare le autorizzazioni per controllare i servizi AWS a cui gli utenti hanno accesso.

Per iniziare a utilizzare 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 ha ruoli IAM configurabili che consentono di specificare a quali servizi AWS possono accedere agli utenti dell'applicazione. In genere, uno sviluppatore utilizza un pool di identità per applicazione. Per ulteriori informazioni sui pool di identità, consultare la sezione [Guida per sviluppatori di Amazon Cognito](#).

Per creare un pool di identità per la tua applicazione:

1. Accedi al [Console Amazon Cognito](#) e clicca su **Crea un nuovo pool di identità**.
2. Inserisci un nome per il pool di identità e seleziona la casella di controllo per abilitare l'accesso alle identità non autenticate. Fare clic su **Crea pool** per creare il pool di identità.
3. Fare clic su **Abilita** per creare i due ruoli predefiniti associati al pool di identità — uno per gli utenti non autenticati e uno per gli utenti autenticati. Questi ruoli predefiniti forniscono l'accesso del pool di identità a Cognito Sync e Mobile Analytics.

Nella pagina successiva viene visualizzato il codice che crea un provider di credenziali in modo da poter facilmente integrare Cognito Identity con l'applicazione Unity. Trasmetti l'oggetto provider di credenziali al costruttore del client di AWS in uso. Il codice sarà il seguente:

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

Fasi successive

- Inizia: Leggi [Introduzione a AWS Mobile SDK for Unity](#) per ottenere una panoramica più dettagliata dei servizi inclusi nell'SDK.
- Esecuzione delle demo: Visualizza il nostro [Applicazioni di esempio Unity](#) che dimostrano casi d'uso comuni. Per eseguire le app di esempio, configurare l'SDK for Unity come descritto sopra, quindi seguire le istruzioni contenute nei file README dei singoli campioni.
- Leggi la documentazione di riferimento delle API: Visualizzare il [Documentazione di riferimento API](#) per AWS Mobile SDK for Unity.
- Fai domande: Pubblica domande sul [Forum SDK AWS Mobile](#) o [apri un problema su Github](#).

Nozioni di base su SDK AWS Mobile per Unity

Questa pagina fornisce una panoramica di ciascun servizio AWS nell'SDK AWS Mobile per Unity e istruzioni su come configurare gli esempi di Unity. Devi compilare tutte le istruzioni riportate sul [Configurazione dell'SDK AWS Mobile per Unity](#) pagina prima di iniziare a utilizzare i servizi riportati di seguito.

Amazon Cognito Identity

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

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

Per le nozioni di base su Cognito Identity, consulta [Guida per sviluppatori di Amazon Cognito](#).

Amazon Cognito Sync

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

Per le nozioni di base su Cognito Sync, consulta [Guida per sviluppatori di Amazon Cognito](#).

Utilizzo di CognitoSync Esempio di manager

Nella `Progettore` riquadro, passa a `Asset/AWSSDK/esempi/CognitoSync`, e nella parte destra del riquadro seleziona `CognitoSync` scena per aprire la scena.

Per eseguire l'esempio fai clic sul pulsante Play (Riproduci) nella parte superiore della schermata dell'editor. Quando l'app viene eseguita, visualizza alcune caselle di testo e pulsanti che consentono di inserire alcune informazioni sul lettore. Di seguito, ci sono una serie di pulsanti che salvano le informazioni sul lettore localmente, sincronizzano le informazioni del giocatore locale con Cognito Cloud, aggiornano le informazioni sul lettore da Cognito Cloud ed eliminano le informazioni sul lettore locale. Premere ciascun pulsante per eseguire un'operazione. L'esempio mostra il feedback nella parte superiore della schermata di gioco.

Per configurare il `CognitoSyncEsempio di manager`, è necessario specificare un ID del pool di identità Cognito. Per specificare questo valore, nell'editor Unity, selezionare `SyncManager` nella Gerarchia riquadro, e inseriscilo nell'`IDENTITY_POOL_ID` casella di testo nella Riquadro controllo.

Note

La `CognitoSyncEsempio di Manager` contiene codice che illustra come utilizzare il provider di identità di Facebook, cercare la macro «`USE_FACEBOOK_LOGIN`». Ciò richiede l'uso dell'SDK di Facebook per Unity. Per ulteriori informazioni, consulta [SDK Facebook per 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 dello storage dei dati, mantenendo una bassa latenza e prestazioni prevedibili.

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

Per ulteriori informazioni su DynamoDB, consulta [Guida per gli sviluppatori di DynamoDB](#).

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

Utilizzo dell'esempio DynamoDB

NellaProgettoriquadro, passa aAsset/AWSSDK/ esempi/DynamoDB. Questo esempio è composto dalle seguenti scene:

- DynamoDBExample: la scena iniziale dell'app
- LowLevelDynamoDbEsempio: esempio che utilizza l'API DynamoDBD di basso livello
- TableQueryAndScanEsempio: esempio che mostra come eseguire query
- HighLevelEsempio: esempio che utilizza l'API DynamoDB di alto livello

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

Per eseguire l'esempio fai clic sul pulsante Play (Riproduci) nella parte superiore della schermata dell'editor. Quando l'app viene eseguita, visualizza una serie di pulsanti:

- Operazioni tabella di basso livello: illustra come creare, elencare, aggiornare, descrivere ed eliminare tabelle.
- Operazioni di query e scansione di livello medio: illustra come eseguire le query.
- Object Mapper ad alto livello: illustra come creare, aggiornare ed eliminare oggetti.

Mobile Analytics

Utilizzo di[Amazon Mobile Analytics](#), è possibile monitorare i comportamenti dei clienti, aggregare metriche, generare visualizzazioni di dati e identificare modelli significativi. L'SDK AWS per Unity fornisce l'integrazione con il servizio Amazon Mobile Analytics. Per informazioni su Mobile Analytics, consulta[Guida dell'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, vengono eliminati eventuali eventi aggiuntivi.
- `dbWarningThreshold`- Questo è il limite alle dimensioni del database che, una volta raggiunto, genererà registri di avviso.
- `maxRequestSize`- Questa è la dimensione massima della richiesta in byte che dovrebbe essere trasmessa in una richiesta HTTP al servizio di analisi mobile.
- `allowUseDataRete`: un valore booleano che specifica se gli eventi di sessione vengono inviati sulla rete dati.

Utilizzo dell'esempio Mobile Analytics

NellaProgettoriquadro, passa aAsset/AWSSDK/ esempi/Mobile Analytics, e nella parte destra del riquadro selezionaEsempio di Amazon Mobile Analyticsscena per aprire la scena. Per utilizzare il campione, è necessario aggiungere l'app utilizzando il[Console Amazon Mobile Analytics](#). Per ulteriori informazioni sull'utilizzo della console Mobile Analytics, consulta[Guida per l'utente di Amazon Mobile Analytics](#).

Attenersi alla seguente procedura per configurare l'esempio prima dell'esecuzione:

1. Selezionare ilAmazonMobileAnalyticsSampleoggetto game.
2. Specifica il tuo ID app (creato nel[Console Amazon Mobile Analytics](#)) nel campo «ID app».
3. Specificare l'ID del pool di identità Cognito (creato utilizzando il[Console Amazon Cognito presso](#)) 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 dei criteri ai ruoli IAM, vedere,[Gestione dei ruoli](#).

Quando si esegue l'applicazione di esempio, tenere presente che gli eventi potrebbero non essere trasmessi immediatamente al servizio di backend. Un thread in background eseguirà il buffer degli 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 abbiano un impatto negativo. A causa della complessa elaborazione che Amazon Mobile Analytics esegue sui dati, gli eventi inviati e i report corrispondenti potrebbero non essere visibili nella console AWS fino a 60 minuti dall'invio iniziale.

Per ulteriori informazioni sui report forniti da Amazon Mobile Analytics, consulta [Metriche di report e dispositivi mobili](#).

Amazon S3

Amazon Simple Storage Service (Amazon S3) fornisce a sviluppatori e 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 [Nozioni di base su S3](#).

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

Configurazione della firma predefinita S3

La firma S3 predefinita è configurata come segue:

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

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

Utilizzo del campione S3

Nella **Progettoria**, passa a **Asset/AWSSDK/esempi/S3**, e nella parte destra del riquadro seleziona **Esempio S3 scena** per aprire la scena. L'esempio illustra come elencare i bucket, elencare gli oggetti all'interno di un bucket, registrare oggetti in un bucket e scaricare oggetti da un bucket. Attenersi alla seguente procedura per configurare l'esempio prima dell'esecuzione:

1. Selezionare il **S3 oggetto** di gioco nel **Gerarchia** riquadro.
2. Nella **Inspector** riquadro immettere i valori per **S3 Bucket Name** e **Sample File Nome**. **S3 Bucket Name** è il nome del bucket utilizzato dal campione e **S3 Sample File** il nome è il nome del file che il campione verrà caricato nel bucket S3 specificato.

3. Assicurati che i tuoi ruoli autenticati e non autenticati dispongano delle autorizzazioni per accedere ai bucket S3 nel tuo account. Per ulteriori informazioni sull'applicazione dei criteri ai ruoli IAM, vedere, [Gestione dei ruoli](#).

Per eseguire l'esempio fai clic sul pulsante Play (Riproduci) nella parte superiore della schermata dell'editor. Quando l'app viene eseguita, visualizza una serie di pulsanti:

- Ottieni oggetti: ottiene un elenco di tutti gli oggetti in tutti i bucket nell'account AWS.
- Ottieni bucket: ottiene un elenco di tutti i bucket nell'account AWS.
- Oggetto post: carica un oggetto in un bucket S3 specifico.
- Elimina oggetto - Elimina tutto l'oggetto 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 notifiche push rapido, flessibile e completamente gestito che consente di inviare singoli messaggi o più messaggi a un numero elevato di destinatari. Amazon Simple Notification Service consente di inviare notifiche push a utenti di dispositivi mobili, destinatari di e-mail o addirittura altri servizi distribuiti in modo semplice e conveniente. Per le nozioni di base su Amazon Simple Notification Service, consul [Amazon Simple Notification Service](#).

AWS Lambda

AWS Lambda è un servizio di elaborazione che esegue codice in risposta a richieste o eventi e gestisce automaticamente le risorse di elaborazione, consentendo di creare con facilità applicazioni che rispondano rapidamente alle nuove informazioni. Le funzioni AWS Lambda possono essere richiamate direttamente da app mobili, IoT e Web e invia una risposta in modo sincrono, consentendo di creare back-end scalabili, sicuri e ad alta disponibilità per le app mobili senza dover eseguire provisioning o gestione dell'infrastruttura. Per ulteriori informazioni, consulta [AWS Lambda](#).

Amazon Cognito Identity

Che cos'è Amazon Cognito Identity?

Utilizzando Amazon Cognito Identity, puoi creare identità univoche per gli utenti e autenticare per accesso sicuro alle risorse AWS come Amazon S3 o Amazon DynamoDB. Amazon Cognito Identity supporta provider di identità pubbliche: Amazon, Facebook, Twitter/Digits, Google o qualsiasi provider compatibile con OpenID Connect, nonché identità non autenticate. Cognito supporta anche identità non autenticate per gli sviluppatori e ti consente di registrare e autenticare gli utenti utilizzando il tuo processo di autenticazione di back-end, mentre continui a utilizzare [Amazon Cognito Sync](#) per sincronizzare i dati utente e accedere alle risorse AWS.

Per ulteriori informazioni su Cognito Identity, consultare il [Guida per sviluppatori di Amazon Cognito](#).

Per informazioni sulla disponibilità di Cognito Authentication Region, consultare [Disponibilità di Amazon Cognito Identity](#).

Utilizzo di un provider pubblico per autenticare gli utenti

Per informazioni sull'utilizzo di provider di identità pubblica come Amazon, Facebook, Twitter/Digits o Google per autenticare gli utenti, consulta la sezione [Provider esterni](#) nella Guida per sviluppatori di Amazon Cognito.

Utilizzo di identità non autenticate per gli sviluppatori

Per informazioni sulle identità autenticate dagli sviluppatori, consulta la sezione [Identità autenticate tramite gli sviluppatori](#) nella Guida per sviluppatori di Amazon Cognito.

Amazon Cognito Sync

Cognito Sync è un servizio AWS e una libreria client che consente la sincronizzazione tra più dispositivi di dati dell'utente relativi all'applicazione. È possibile utilizzare l'API Cognito Sync per sincronizzare i dati utente tra i dispositivi. Per utilizzare Cognito Sync nella tua app, devi includere AWS Mobile SDK for Unity nel tuo progetto.

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

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, vedere [AWS Mobile Analytics](#).

Integrazione di Amazon Mobile Analytics

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

Crea un'app nella console di Mobile Analytics

Passa alla [Console Amazon Mobile Analytics](#) e crea un'app. Nota: appId valore, come è necessario in un secondo momento.

Note

Per ulteriori informazioni sull'utilizzo della console, consulta la [Guida per l'utente di Amazon Mobile Analytics](#).

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

Integra Mobile Analytics nella tua app

Per accedere a Mobile Analytics da Unity è necessario utilizzare le seguenti istruzioni:

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

La procedura consigliata è 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 riportate in [Amazon Cognito Identity](#).

Creare un'istanza di un `MobileAnalyticsIstanza di manager` con le seguenti informazioni:

- `cognitoIdentityPoolId` - ID del pool di identità Cognito per la tua app
- `cognitoRegion` - La regione per il tuo Cognito Identity Pool, ad esempio»`RegionEndpoint.UsaAST1`»
- `region` - La regione per il servizio Mobile Analytics, ad esempio»`RegionEndpoint.UsaAST1`»
- `appId` - Il valore generato da Mobile Analytics Console quando aggiungi un'app

Utilizzo dell'`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 app viene generato per te durante la procedura guidata di creazione dell'app. Entrambi questi valori devono corrispondere a quelli di Mobile Analytics Console.

La `appId` viene utilizzato per raggruppare i dati nella console di Mobile Analytics. Per trovare il tuo ID app dopo aver creato l'app nella console di Mobile Analytics, accedi a Mobile Analytics Console, fai clic sull'icona a forma di ingranaggio nell'angolo in alto a destra dello schermo. Viene visualizzata la pagina Gestione app che elenca tutte le app registrate e i relativi ID app.

Registrare gli eventi di monetizzazione

SDK for Unity fornisce il `MonetizationEvent` class, che consente di generare eventi di monetizzazione per monitorare gli acquisti effettuati all'interno di 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);
```

Registrazione eventi personalizzati

Mobile Analytics consente di definire eventi personalizzati. Gli eventi personalizzati sono definiti interamente da te e ti aiutano a monitorare le azioni degli utenti specifiche della tua app o del tuo gioco. Per ulteriori informazioni sugli eventi personalizzati, consulta [Eventi personalizzati](#). Per questo esempio, supponiamo che la tua app sia un gioco e desideri registrare un evento quando un utente completa un livello. Crea un «LevelComplete» evento creando un nuovo `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 lo stato attivo è possibile sospendere la sessione. Nello stato `OnApplicationFocus` controlla se l'app è in pausa. In tal caso,

chiama `PauseSession` altrimenti chiama `ResumeSession` come illustrato nel seguente frammento di codice:

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

Per impostazione predefinita, se l'utente passa lo stato attivo dall'app per meno di 5 secondi e torna all'app, la sessione verrà ripresa. Se l'utente allontana lo stato attivo 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 [Nozioni di base su SDK AWS Mobile per Unity](#).

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3) fornisce a sviluppatori e team IT uno storage degli oggetti sicuro, durevole e altamente scalabile. Gli sviluppatori Unity possono sfruttare S3 per caricare dinamicamente le risorse utilizzate dai loro giochi. Ciò può rendere i giochi inizialmente più veloci dagli app store.

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

Per informazioni sulla disponibilità delle regioni AWS S3, consulta [Disponibilità di una regione AWS Service](#).

Note

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

Creare e configurare un bucket S3

Amazon S3 memorizza le tue risorse nei bucket Amazon S3, ovvero contenitori di storage cloud che vivono in uno specifico [regione](#). Ogni bucket Amazon S3 deve avere un nome univoco globale. Puoi utilizzare il plugin [Console Amazon S3](#) Per creare un bucket.

Creare un bucket S3

1. Accedi alla [Console Amazon S3](#) e fare clic su **Creare bucket**.
2. Inserire un nome bucket, selezionare una regione e fare clic su **Create**.

Impostazione delle autorizzazioni per S3

La policy del ruolo IAM predefinita concede alla tua applicazione l'accesso ad Amazon Mobile Analytics e Amazon Cognito Sync. Affinché il pool di identità Cognito acceda ad Amazon S3, devi modificare i ruoli del pool di identità.

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

3. Fare clic sul ruolo degli utenti non autenticati (non verrà aggiunto l'autenticazione al nome del pool di identità).
4. Fare clic su **Creare policy** per i ruoli, selezionare **generatore di policy** e quindi fare clic su **Seleziona**.
5. Sul **Modifica delle autorizzazioni** inserisci le impostazioni mostrate nell'immagine seguente, sostituendo l'Amazon Resource Name (ARN) con il tuo. L'ARN di un bucket S3 sembra `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 daranno al tuo pool di identità pieno l'accesso 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. At the top, there are radio buttons for 'Effect', with 'Allow' selected and 'Deny' unselected. Below that is a dropdown menu for 'AWS Service' set to 'Amazon S3'. Underneath is a text box for 'Actions' containing 'All Actions Selected'. The 'Amazon Resource Name (ARN)' field contains 'arn:aws:s3:::examplebucket/*'. Below the ARN field is a link for 'Add Conditions (optional)'. At the bottom is a button labeled 'Add Statement'.

1. Fai clic sull'icona della barra degli strumenti **Aggiungi istruzione** e poi fai clic **Fase successiva**.
2. La procedura guidata mostrerà la configurazione generata. Fare clic su **Applica policy**.

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

Carica file dalla console

Per caricare un file di test nel bucket:

1. Nella console S3, nella vista bucket, fai clic su **Caricamento**.
2. Fare clic su **Aggiunta di file** e seleziona un file di prova da caricare. Per questo tutorial, supponiamo che tu stia caricando un'immagine chiamata `myImage.jpg`.
3. Con l'immagine di prova selezionata, fai clic sul **Inizia a caricare**.

(facoltativo) Configurare la versione Signature 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 Signature Version 4. Tuttavia, molte regioni precedenti supportano ancora le richieste Signature Version 4 e Signature Version 2.

Se il bucket si trova in una delle regioni che non supportano le richieste Signature Version 2 come elencato su [questa pagina](#), devi impostare il `AWSConfigsS3.UseSignatureProprietà Version4` su «true».

Per ulteriori informazioni sulle versioni di AWS Signature, consulta [Autenticazione delle richieste \(AWS Signature Version 4\)](#).

Creare il client Amazon S3

Per utilizzare Amazon S3, dobbiamo innanzitutto creare un'istanza `AmazonS3Client` che richieda un riferimento a `CognitoAWSCredentials` istanza creata in precedenza:

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

La `AmazonS3Client` class è il punto di ingresso all'API S3 di alto livello.

Creazione di un elenco di bucket

Per elencare i bucket in un account AWS chiama il `AmazonS3Client.ListBucketsAsync` come illustrato 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, chiamare `IAmazonS3Client.ListObjectsAsync` come illustrato 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, crea un `GetObjectRichiesta`, specificando il nome del bucket e la chiave e passa l'oggetto a una chiamata al `Client.GetObjectasincrono`:

```
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 del `GetObjectRichiesta`, richiamata e un'istanza di `AsyncOptions`. Il callback deve essere di tipo: `AmazonServiceCallback<GetObjectRequest, GetObjectResponse>`. L'istanza di `AsyncOptions` è facoltativa. Se specificato, determina se il callback verrà eseguito sul thread principale.

Caricamento di un oggetto

Per caricare un oggetto, scrivi il tuo oggetto in uno stream, creane un nuovo `PostObjectRichiedere` e specificare la chiave, il nome del bucket e i dati del flusso.

AWS SDK for Unity utilizza il client HTTP WWW che non supporta l'operazione HTTP PUT. Per caricare un oggetto nel bucket S3, è necessario 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 dello storage dei dati, mantenendo una bassa latenza e prestazioni prevedibili. Per informazioni su DynamoDB, consulta [Amazon DynamoDB](#).

AWS Mobile SDK for Unity fornisce una libreria di alto livello per lavorare con DynamoDB. È inoltre possibile effettuare richieste direttamente contro l'API DynamoDB di basso livello, ma per la maggior parte dei casi d'uso è consigliata la libreria di alto livello. `LaAmazonDynamoDBClient` è una parte particolarmente utile della libreria di alto livello. Utilizzando questa classe, è possibile eseguire diverse operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) ed eseguire query.

Note

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

Integrazione di Amazon DynamoDB

Per utilizzare DynamoDB in un'applicazione Unity, è necessario aggiungere Unity SDK al progetto. Se non l'hai già fatto, [scarica l'SDK for Unity](#) e segui le istruzioni in [Configurazione dell'SDK AWS Mobile per Unity](#). Consigliamo di utilizzare Amazon Cognito Identity per fornire credenziali AWS temporanee per le applicazioni. Queste credenziali consentono alla tua app di accedere ai servizi e alle risorse AWS.

Per utilizzare DynamoDB in un'applicazione, è necessario impostare le autorizzazioni corrette. Il seguente criterio 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 policy deve essere applicata ai ruoli assegnati al pool di identità Cognito, sostituendo la **Resource** valore con l'ARN corretto per la tua tabella DynamoDB. Cognito crea automaticamente un ruolo per il tuo nuovo pool di identità e puoi applicare le policy a questo ruolo nella [Console IAM](#).

È necessario 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 [Utilizzo di IAM per controllare l'accesso alle risorse DynamoDB](#).

Creazione di una tabella DynamoDB

Ora che abbiamo configurato le autorizzazioni e le credenziali, creiamo una tabella DynamoDB per la nostra applicazione. Per creare una tabella, vai alla [La console DynamoDB](#) e procedi come descritto:

1. Fare clic su Create Table (Crea tabella).
2. InvioBookstore come nome della tabella.
3. Seleziona Hash come tipo di chiave primaria.
4. Seleziona Numero inserisci id per il nome dell'attributo hash. Fai clic su Continue (Continua).
5. Fare clic su Continua di nuovo per saltare l'aggiunta di indici.
6. Imposta la capacità di lettura su 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 database.

Creazione di un client DynamoDB

Affinché la nostra app interagisca con una tabella DynamoDB, abbiamo bisogno di un client. Possiamo creare un client DynamoDB predefinito come segue:

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

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

La classe `DbClient` 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 il modello di persistenza degli oggetti.

Descrizione di una tabella

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

```
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 oggetto `DescribeTableRequest`, assegniamo il nome della nostra tabella al **TableName** e quindi passiamo l'oggetto della richiesta al metodo `DescribeTableAsync` asincrono sull'oggetto `DbClient`. `DescribeTableAsync` accetta anche un delegato che verrà chiamato al termine dell'operazione asincrona.

Note

Tutti i metodi asincroni sul `AmazonDynamoDBClient` prende i delegati che vengono chiamati al termine dell'operazione asincrona.

Salva un oggetto

Per salvare un oggetto in DynamoDB, utilizzare il `SaveAsync<T>` metodo del `AmazonDynamoDBClient`, dove `T` è il tipo di oggetto che si sta salvando.

Abbiamo chiamato il nostro database «Bookstore» e, in linea con quel tema, implementeremo un modello di dati che registra gli attributi relativi al libro. 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 di libreria avremmo bisogno di campi aggiuntivi per cose come autore e prezzo. La classe `Book` è decorata con l'attributo `[DynamoDBTable]`, che definisce gli oggetti della tabella di database di tipo `Book` verranno scritti. La chiave per ogni istanza della classe `Book` è identificata utilizzando `[DynamoDBHashKey]` attributo. Le proprietà sono identificate con l'attributo `[DynamoDBProperty]`, specificando la colonna nella tabella del database in cui verrà scritta la proprietà. Con il modello in posizione, possiamo scrivere alcuni metodi per creare, recuperare, aggiornare ed eliminare gli oggetti `Book`.

Creazione di 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);
}
```

Aggiornamento di 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");
            });
        }
    });
}
```

Eliminazione di 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");
    });
}
});
}
```

Amazon Simple Notification Service

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

Questo argomento ti guiderà nella configurazione dell'app di esempio AWS SDK for Unity, SNSExample.unity, per ricevere notifiche push mobili tramite Amazon SNS.

È possibile creare app iOS e Android utilizzando l'esempio SNSExample.unity. I passaggi di configurazione sono diversi tra iOS e Android, leggi la sezione appropriata di seguito per la piattaforma che stai prendendo di mira.

Prerequisiti

Per utilizzare questa soluzione sono necessari i prerequisiti riportati di seguito.

Impostazione delle autorizzazioni per SNS

Quando crei un Cognito Identity Pool vengono generati due ruoli IAM:

- Cognito/_ <Identity-Pool-Name>Autore_DefaultRole- il ruolo IAM predefinito per gli utenti autenticati
- Cognito/_ <Identity-Pool-Name>Non autore_DefaultRole- il ruolo IAM predefinito per utenti autenticati

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

1. Passare alla barra degli strumenti [Console IAM](#) e seleziona il ruolo IAM da configurare.
2. Fare clic su [Collegamento della policy](#), seleziona Amazon SNSFullAccesspolicy e clicca [Collegamento della policy](#).

Note

Utilizzo di Amazon SNSFullAccessNon è raccomandato in un ambiente di produzione, lo usiamo qui per consentirti di funzionare rapidamente. Per ulteriori informazioni sulla specifica delle autorizzazioni per un ruolo IAM, consulta [Panoramica delle autorizzazioni del ruolo IAM](#).

Prerequisiti iOS

- Iscrizione al programma Apple iOS Developer
- Genera un'identità di firma
- Creare un profilo di provisioning configurato per le notifiche push

Dovrai eseguire l'app su un dispositivo fisico per ricevere notifiche push. Per eseguire la tua app su un dispositivo devi avere un abbonamento al [iscrizione al programma per sviluppatori Apple iOS](#). Una volta ottenuto l'iscrizione, è possibile utilizzare Xcode per generare un'identità di firma. Per ulteriori informazioni, consulta Apple [Quick Start della distribuzione app](#) documentazione. Successivamente avrai bisogno di un profilo di provisioning configurato per le notifiche push per ulteriori informazioni, consulta Apple [Configurazione delle notifiche push](#) documentazione.

Prerequisiti Android

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

Configurazione dell'app Unity Sample per iOS

Apri l'editor Unity e creane uno nuovo. Importa il pacchetto AWS SDK for Unity selezionando Asset/Import Package/Pacchetto personalizzato e selezione aws-unity-sdk- pacchetto sns-2.0.0.1.unity. Assicurati che tutti gli articoli presentilimportazione di pacchettilla finestra di dialogo è selezionata e fai clic Importa.

Configurazione delle

Esegui la procedura seguente per configurare il progetto Unity:

1. NellaProgettoriquadro, passa aAsset/AWSSDK/esempie apri la scena SNSEample.
2. NellaHIERARCHYriquadro, selezionare SNSEample.
3. NellaInspectoril riquadro specifica l'ID del pool di identità Cognito.
4. Si noti che c'è una casella di testo perARN dell'applicazione della piattaforma iOS, genererai queste informazioni in un secondo momento.
5. SelezionaFile/Creazione delle impostazioni, nelCreazione delle impostazionifinestra di dialogo, fai clic sulAggiungi correntepulsante sotto la barra degli strumentiCreazione di scenecasella di riepilogo per aggiungere la scena corrente alla build.
6. UNDERPiattafomaselezionareiIOSe clickImpostazioni lettore...pulsante, nelRiquadro controllodell'editor Unity, fai clic sull'icona dell'iPhone e scorri verso il basso fino all'identificazionesezione e specifica aBundle Identifier (Identificatore bundle).

Configurazione iOS

Esegui la procedura seguente per configurare l'esempio per configurare le impostazioni specifiche per iOS:

1. In un browser Web, passare alla[Centro membri Apple Developer](#), clickCertificati, identificatori e profili.
2. Fare clic sulidentificatori disottoApp iOS, fare clic sul pulsante più nell'angolo in alto a destra della pagina Web per aggiungere un nuovo ID app iOS e inserire una descrizione dell'ID app.
3. Scorrere fino alla barra degli strumentiAggiungi suffisso IDsezione e selezionaID dell'app esplicitoe inserisci il tuo identificatore del bundle.
4. Scorrere fino alla barra degli strumentiApp Servicessezione e selezionaNotifiche push.
5. Fai clic sull'icona della barra degli strumentiContinuaPulsante.
6. Fai clic sull'icona della barra degli strumentiInviaPulsante.
7. Fai clic sull'icona della barra degli strumentiFattoPulsante.
8. Seleziona l'ID app appena creato e fai clic sulModificarePulsante.
9. Scorrere fino alla barra degli strumentiNotifiche pushsezione.

10. Fai clic sull'icona della barra degli strumenti Creazione di certificato pulsante UNDER 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 nella barra degli strumenti Certificati, identificatori e profili pagina web, fare clic su All (Tutti) sotto Profili di provisioning.
13. Fare clic sul pulsante più nell'angolo in alto a destra per aggiungere un nuovo profilo di provisioning.
14. Seleziona Sviluppo di app iOS e click Continua Pulsante.
15. Seleziona il tuo ID app e fai clic sul Continua Pulsante.
16. Seleziona il tuo certificato sviluppatore e fai clic su Continua Pulsante.
17. Seleziona il tuo dispositivo e fai clic su Continua Pulsante.
18. Inserisci il nome di un profilo e fai clic sul Genera Pulsante.
19. Scarica e fai doppio clic sul file di provisioning per installare il profilo di provisioning.

Dopo averne aggiunto uno nuovo, potrebbe essere necessario aggiornare i profili di provisioning in Xcode. In Xcode:

1. Seleziona il pulsante Xcode/Preferenze voce di menu.
2. Seleziona il pulsante Accounts scheda, seleziona il tuo ID Apple e fai clic su Consulta i dettagli.
3. Fare clic sul pulsante di aggiornamento nell'angolo in basso a sinistra della finestra di dialogo per aggiornare i profili di provisioning e assicurarsi che il nuovo profilo sia visualizzato.

Configurazione SNS

1. Esegui `KeyChain` accedi all'app, seleziona i miei certificati nella parte inferiore sinistra dello schermo, fare clic con il pulsante destro del mouse sul certificato SSL generato per connettersi a APNS e selezionare Esportazione, 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 passare alla [Console SNS](#) e click Applicazioni sul lato sinistro dello schermo.
3. Fare clic su Crea applicazione piattaforma per creare una nuova applicazione piattaforma SNS.
4. Specificare un Nome applicazione.

5. Seleziona **Apple Push Notification Service Sandbox (APNS_SANDBOX)** per Piattaforma delle notifiche push.
6. Fare clic su **Scelta file** e seleziona il file P12 creato quando hai esportato il certificato SSL.
7. Immettere la password specificata all'esportazione del certificato SSL e fare clic su **Carica credenziali da file**.
8. Fare clic su **Crea applicazione piattaforma**.
9. Seleziona l'applicazione Platform appena creata e copia l'ARN dell'applicazione.
10. Torna al tuo progetto in Unity Editor, seleziona **Esempio SNS nella HIERARCHY** riquadro, nel **Inspector** riquadro e incolla l'ARN applicazione piattaforma nella casella di testo con etichetta **ARN** dell'applicazione della piattaforma iOS.
11. Seleziona **File/Creazione delle impostazioni** e click **Build** pulsante questo creerà un progetto Xcode.

Utilizzo di Xcode

1. Aprire il progetto Xcode e selezionare il progetto in Project Navigator.
2. Verificare che l'identificatore del bundle sia impostato correttamente
3. Verifica che il tuo account Apple Developer sia specificato nella **Team**- questa operazione è necessaria per rendere effettiva l'effetto del tuo profilo di provisioning.
4. Costruisci il progetto ed eseguilo sul tuo dispositivo.
5. Tocca il pulsante **Registrati** per la notifica, tap **OK** per consentire le notifiche, l'app visualizzerà il token del dispositivo

Nella [Console SNS](#), click **Applicazioni**, seleziona l'applicazione della piattaforma e fai clic su **Creazione di endpoint** platforme inserisci il token del dispositivo visualizzato dall'app.

A questo punto la tua app, APNS e NSN sono completamente configurati. È possibile selezionare l'applicazione della piattaforma, selezionare l'endpoint e fare clic su **Pubblicazione su endpoint** per inviare una notifica push al dispositivo.

Esempio Unity (iOS)

Il campione crea un **CognitoAWSCredentials** stanza per generare credenziali temporanee a portata limitata che consentono all'app di chiamare i servizi AWS. Crea anche un'istanza di **AmazonSimpleNotificationServiceClient** per comunicare con SNS. L'app mostra due pulsanti etichettati **Registrati** per la notifica e **Annulla** la registrazione.

Quando Registrati per le notifiche il pulsante è toccato, il `RegisterDevice()` viene chiamato il metodo `RegisterDevice()` chiama `UnityEngine.iOS.NotificationServices.RegisterForNotifica` che specifica quali tipi di notifica (avviso, suono o badge) verranno utilizzati. Effettua anche una chiamata asincrona ad APNS per ottenere un token del dispositivo. Poiché non è definito alcun callback, `CheckForDeviceToken` viene chiamato ripetutamente (fino a 10 volte) per verificare la presenza del token del dispositivo.

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

Il campione è ora configurato per ricevere notifiche push. È possibile navigare verso il [Console SNS](#), click Applicazioni sul lato sinistro della pagina, seleziona l'applicazione piattaforma, seleziona un endpoint e fai clic su Pubblicazione su endpoint. Seleziona l'endpoint da usare e fai clic su Pubblicazione su endpoint. Digita un messaggio di testo nella casella di testo e fai clic su Pubblica un messaggio Per pubblicare un messaggio.

Configurazione dell'app Unity Sample per Android

Apri l'editor Unity e creane uno nuovo. Importa il pacchetto AWS SDK for Unity selezionando Asset/Import Package/Pacchetto personalizzato e seleziona `aws-unity-sdk-pacchetto sns-2.0.0.1.unity`. Assicurati che tutti gli articoli presenti l'importazione di pacchetto la finestra di dialogo è selezionata e fai clic Importa.

Configurazione delle

Esegui la procedura seguente per configurare il progetto Unity:

1. Nella **Progettore** riquadro, passa a `Asset/AWSSDK/esempie` apri la scena `SNSEample`.
2. Nella **HIERARCHY** riquadro, selezionare `SNSEample`.
3. Nella **Inspector** riquadro specifica l'ID del pool di identità Cognito.
4. Si noti che c'è una casella di testo per Applicazione piattaforma Android ARNeID progetto Google Console, genererai queste informazioni in un secondo momento.
5. Seleziona `File/Creazione delle impostazioni`, nel `Creazione delle impostazioni` finestra di dialogo, fai clic sul `Aggiungi corrente` pulsante sotto la barra degli strumenti `Creazione di scena` casella di riepilogo per aggiungere la scena corrente alla build.

6. **UNDER**PiattaformaselezionareAndroidFare clic sulla barra degli strumentiImpostazioni lettore...pulsante, nelRiquadro controllodell'editor Unity, fai clic sull'icona Android e scorri verso il basso fino allIdentificazione sezione e specifica aBundle Identifier (Identificatore bundle).
7. Copy (Copia)android-support-v4.jar egoogle-play-services.jar nelAsset/Plug-in/Androiddirectory nella barra degli strumentiProgettoRiquadro.

Per ulteriori informazioni su dove trovareandroid-support-v4.jar, vedere[Installazione della libreria di Support Android](#). Per ulteriori informazioni su come trovaregoogle-play-services.jar, vedere[API di Google per Android](#).

Configurazione Android

Per prima cosa aggiungi un nuovo progetto Google API:

1. In un browser Web, passare alla[Google Developer Console](#), clickCreazione di un progetto.
2. NellaNuovo progettocasella, inserisci il nome di un progetto, prendi nota del numero del progetto (ne avrai bisogno in seguito) e fai clic suCreate.

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

1. Nella Google Developers Console, il nuovo progetto dovrebbe già essere selezionato, in caso contrario, selezionarlo nel menu a discesa nella parte superiore della pagina.
2. SelezionaAPI e autenticazione dalla barra laterale sul lato sinistro della pagina.
3. Nella casella di ricerca, digita «Google Cloud Messaging per Android» e fai clic sulGoogle Cloud Messaging per Androidlink qui di seguito.
4. Fare clic suAbilitazione dell'API.

Finalmente ottieni una chiave API:

1. Nella Google Developers Console, selezionaAPI e autenticazione >Credenziali.
2. **UNDER**Accesso alle API pubbliche, clickCreazione di una nuova chiave.
3. NellaCreazione di una nuova chiavefinestra di dialogo, fai clicChiave server.
4. Nella finestra di dialogo risultante, fai clic suCreatee copia la chiave API visualizzata.

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

Configurazione SNS

1. In un browser Web passare alla [Console SNS](#) e click **Applicazioni** sul lato sinistro dello schermo.
2. Fare clic su **Crea applicazione piattaforma** per creare una nuova applicazione piattaforma SNS.
3. Specificare un **Nome** applicazione
4. Seleziona **Google Cloud Messaging (GCM)** per **Piattaforma** delle notifiche push
5. Incolla la chiave API nella casella di testo con etichetta **Chiave API**.
6. Fare clic su **Crea applicazione piattaforma**
7. Seleziona l'applicazione Platform appena creata e copia l'ARN dell'applicazione.
8. Torna al tuo progetto in Unity Editor, seleziona **Esempio SNS** nella **HIERARCHY** riquadro, nel **Inspector** riquadro e incolla l'ARN applicazione piattaforma nella casella di testo con etichetta **Applicazione piattaforma Android ARN** e il numero del progetto nella casella di testo con etichetta **ID progetto Google Console**.
9. **Connect** del dispositivo Android al computer, seleziona **File/Creazione delle impostazioni** Fare clic sulla barra degli strumenti **Creazione ed esecuzione**.

Esempio Unity (Android)

Il campione crea un **CognitoAWSCredentials** istanza per generare credenziali temporanee a portata limitata che consentono all'app di chiamare i servizi AWS. Crea anche un'istanza di **AmazonSimpleNotificationServiceClient** per comunicare con SNS.

L'app mostra due pulsanti etichettati **Registrati** per la notifica e **Annulla la registrazione**. Quando **Registrati** per le notifiche il pulsante è toccato, il `RegisterDevice()` viene chiamato il metodo `RegisterDevice()` chiama `GCM.Register`, che registra 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 `callbackAmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync` viene chiamato per creare un endpoint della piattaforma per ricevere messaggi SNS.

Il campione è ora configurato per ricevere notifiche push. È possibile navigare verso il [Console SNS](#), click **Applicazioni** sul lato sinistro della pagina, seleziona l'applicazione piattaforma, seleziona un endpoint e fai clic su **Pubblicazione su endpoint**. Seleziona l'endpoint da usare e fai clic

su Pubblicazione su endpoint. Digita un messaggio di testo nella casella di testo e fai clic su Pubblica un messaggio Per pubblicare un messaggio.

AWS Lambda

AWS Lambda è un servizio di elaborazione che esegue codice in risposta a richieste o eventi e gestisce automaticamente le risorse di elaborazione, consentendo di creare con facilità applicazioni che rispondano rapidamente alle nuove informazioni. Le funzioni AWS Lambda possono essere chiamate direttamente dalle app mobili, IoT e Web e invia una risposta in modo sincrono, consentendo di creare back-end dimensionabili, sicuri e ad ampia disponibilità per le proprie app mobili senza incorrere nel provisioning o nella gestione dell'infrastruttura.

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

- 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 connessi).
- 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. Grazie a queste funzionalità, puoi utilizzare Lambda per creare facilmente trigger per servizi AWS, ad esempio Amazon S3 e Amazon DynamoDB, elaborare dati di streaming archiviati in Amazon Kinesis o creare sistemi di back-end propri che funzionino con le stesse scalabilità, prestazioni e sicurezza di AWS.

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

Autorizzazioni

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. Queste autorizzazioni vengono concesse creando un ruolo IAM, noto come ruolo di esecuzione.
- Autorizzazioni di chiamata— Le autorizzazioni di cui l'origine dell'evento ha bisogno per comunicare con la funzione Lambda. A seconda del modello di invocazione (modello push o pull), è possibile concedere queste autorizzazioni utilizzando il ruolo di esecuzione o i criteri delle risorse (il criterio di accesso associato alla funzione Lambda).

Configurazione progetto

Autorizzazioni impostate per AWS Lambda

1. Apertura della [Console AWS IAM](#).
2. Allega questa policy personalizzata ai tuoi ruoli, che consente alla tua applicazione di effettuare chiamate ad AWS Lambda.

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

Creare un nuovo ruolo di esecuzione

Questo ruolo si applica alla funzione Lambda che creerai nel passaggio successivo e determina a quali risorse AWS a cui è possibile accedere.

1. Apertura della [Console AWS IAM](#).
2. Fare clic su Ruoli.
3. Fare clic su Crea nuovi ruoli.
4. Segui le istruzioni visualizzate sullo schermo per selezionare i servizi e i criteri corrispondenti a cui la funzione Lambda avrà bisogno di accedere. Ad esempio, se desideri che la tua funzione Lambda crei un bucket S3, la tua policy avrà bisogno dell'accesso in scrittura a S3.
5. Fare clic su crea ruolo.

Creazione di una funzione in AWS Lambda

1. Apertura della [Console AWS Lambda](#).

2. Fare clic su **Creare una funzione Lambda**.
3. Fare clic su **Saltare** per ignorare la creazione di un progetto.
4. Configurare la propria funzione nella schermata successiva. Immettere il nome della funzione, una descrizione e scegliere il runtime. Seguire le istruzioni visualizzate sullo schermo in base al runtime scelto. Specificare le autorizzazioni di esecuzione assegnando il ruolo di esecuzione appena creato alla funzione.
5. Al termine, fare clic su **Successivo**.
6. Fare clic su **crea funzione**.

Creare un client Lambda

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

Creare un oggetto di richiesta

Creare un oggetto 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  
};
```

Richiamare la funzione Lambda

Richiamo chiamata, passando l'oggetto 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.www` utilizzata dall'SDK AWS per Unity, i messaggi di errore dettagliati non vengono restituiti quando si verifica un problema durante la chiamata a un servizio AWS. In questo argomento vengono descritte alcune idee per la risoluzione di tali problemi.

Assicurati che il ruolo IAM abbia le autorizzazioni richieste

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

Il seguente file policy illustra 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 criteri 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 su come specificare le policy, consulta [Policy IAM](#).

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 proxy HTTP/HTTPS per visualizzare le richieste e le risposte per ottenere maggiori informazioni su ciò che sta accadendo. Sono disponibili diversi debugger proxy HTTP, come:

- [Carlo](#)- un proxy di debug web per OSX
- [Violinista](#)- un proxyfidd di debug web per Windows

Important

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

Sia Charles che Fiddler richiedono una certa configurazione per poter visualizzare il traffico crittografato SSL, leggi la documentazione di questi strumenti per ulteriori informazioni. Se si utilizza un proxy di debug Web che non può essere configurato per visualizzare il traffico crittografato, aprire il file `aws_endpoints_json` (situato in `AWSUnitySDK/AWSCore/Resources`) e imposta il tag HTTP per il servizio AWS necessario per eseguire il debug su `true`