
SDK para móviles de AWS

Guía para desarrolladores de Xamarin

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no sean propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

.....	vii
¿Qué es el SDK de AWS Mobile para .NET y Xamarin?	1
Guías y temas relacionados	1
Contenido de referencia archivado	1
¿Qué se incluye en AWS Mobile SDK for .NET and Xamarin?	1
Compatibilidad	2
¿Cómo puedo obtener AWS Mobile SDK for .NET and Xamarin?	2
Acerca de los servicios móviles de AWS	2
Configuración del SDK de AWS Mobile para .NET y Xamarin	4
Requisitos previos	4
Paso 1: Obtener las credenciales de AWS	4
Paso 2: Establecer permisos	5
Paso 3: Crear un nuevo proyecto de	6
Windows	6
OS X	6
Paso 4: Instale el SDK de AWS Mobile para .NET y Xamarin	6
Windows	6
Mac (OS X)	7
Paso 5: Configurar el SDK de AWS Mobile para .NET y Xamarin	7
Definición del registro	7
Definición del punto de enlace de la región	8
Configuración de los ajustes del proxy HTTP	8
Corrección del sesgo del reloj	8
Pasos siguientes	9
Introducción al SDK de AWS Mobile para .NET y Xamarin	10
Almacenar y recuperar archivos en Amazon S3	10
Configuración del proyecto	10
Inicializar el S3TransferUtilityCliente	12
Cargar un archivo en Amazon S3	12
Descargar un archivo de Amazon S3	12
Sincronización de datos de usuario con Cognito Sync	12
Configuración del proyecto	10
Inicializar elCognitoSyncGerente	13
Sincronización de los datos de usuario	13
Almacenar y recuperar datos con DynamoDB	14
Configuración del proyecto	10
InicializarAmazonDynamoDBClient	16
Crear una clase	16
Almacenamiento de un elemento	16
Recuperación de un elemento	17
Actualización de un elemento	17
Eliminación de un elemento	17
Seguimiento de los datos de uso de las aplicaciones con Amazon Mobile Analytics	17
Configuración del proyecto	10
InicializarMobileAnalyticsGerente	19
Seguimiento de eventos de la sesión	19
Recibir notificaciones de inserción usando SNS (Xamarin iOS)	19
Configuración del proyecto	10
Creación de un cliente de SNS	21
Registro de una aplicación para recibir notificaciones remotas	21
Envío de un mensaje desde la consola de SNS a un punto de enlace	22
Recepción de notificaciones de inserción con SNS (Xamarin Android)	22
Configuración del proyecto	10
Creación de un cliente de SNS	21

Registro de una aplicación para recibir notificaciones remotas	21
Envío de un mensaje desde la consola de SNS a un punto de enlace	22
Identidad de Amazon Cognito	27
¿Qué es Amazon Cognito Identity?	27
Uso de un proveedor público para autenticar a los usuarios	27
Uso de identidades autenticadas por el desarrollador	27
Amazon Cognito Sync	28
¿Qué es Amazon Cognito Sync?	28
Amazon Mobile Analytics	29
Conceptos clave	29
Tipos de informes	29
Configuración del proyecto	10
Requisitos previos	4
Configuración de Mobile Analytics	18
Integración de Mobile Analytics con su aplicación	30
Creación de una aplicación en la consola de Mobile Analytics	18
Creación de unMobileAnalyticsCliente manager	31
Registro de eventos de monetización	31
Registro de eventos personalizados	31
Registro de sesiones	32
Amazon Simple Storage Service (S3)	33
¿Qué es S3?	33
Conceptos clave	29
Bucket	33
Objetos	33
Metadatos de objetos	34
Configuración del proyecto	10
Requisitos previos	4
Creación de S3 Bucket	34
Definición de permisos de S3	10
(opcional) Configuración de la versión de Signature para las solicitudes de S3	11
Integración de S3 en una aplicación	35
Utilizar S3 Transfer Utility	35
Inicializar elTransferUtility	36
Configurar elTransferUtility	36
Descarga de un archivo	36
Cargar un archivo	37
Uso de las API de nivel de servicio de S3	37
Inicialización del cliente de Amazon S3	37
Descarga de un archivo	36
Cargar un archivo	37
Eliminación de un elemento	17
Eliminación de varios elementos	38
Lista de buckets	39
Lista de objetos	39
Obtención de la región de un bucket	40
Obtención de la política de un bucket	40
Amazon DynamoDB	41
¿Qué es Amazon DynamoDB?	41
Conceptos clave	29
Tablas	41
Elementos y atributos	41
Tipos de datos	41
Clave principal	42
Índices secundarios	42
Consulta y examen	42
Configuración del proyecto	10

Requisitos previos	4
Creación de una tabla de DynamoDB	14
Definición de permisos para DynamoDB	15
Integración de DynamoDB en una aplicación	44
Uso del modelo de documento	45
Creación de un cliente de DynamoDB	45
Operaciones de CRUD	45
Uso del modelo de persistencia de objetos	47
Información general	47
Tipos de datos admitidos	48
Creación de un cliente de DynamoDB	45
Operaciones de CRUD	45
Consulta y examen	42
Uso de las API de nivel de servicio de DynamoDB	50
Creación de un cliente de DynamoDB	45
Operaciones de CRUD	45
Consulta y examen	42
Amazon Simple Notification Service (SNS)	54
Conceptos clave	29
Temas	54
Suscripciones	54
Publicación	54
Configuración del proyecto	10
Requisitos previos	4
Integrar SNS en la aplicación	55
Enviar notificaciones de inserción (Xamarin para Android)	55
Configuración del proyecto	10
Creación de un cliente de SNS	21
Registro de una aplicación para recibir notificaciones remotas	21
Envío de un mensaje desde la consola de SNS a un punto de enlace	22
Envío de notificaciones de inserción (Xamarin iOS)	59
Configuración del proyecto	10
Creación de un cliente de SNS	21
Registro de una aplicación para recibir notificaciones remotas	21
Envío de un mensaje desde la consola de SNS a un punto de enlace	22
Enviar y recibir notificaciones SMS.	62
Creación de un tema	62
Suscribirse a un tema utilizando el protocolo SMS	63
Publicar un mensaje	63
Envío de mensajes a puntos de enlace HTTP/HTTPS	64
Configuración de un punto de enlace HTTP/HTTPS para recibir mensajes de Amazon SNS	64
Suscripción del punto de enlace HTTP/HTTPS a un tema de Amazon SNS	64
Confirmación de la suscripción	65
Envío de mensajes al punto de enlace HTTP/HTTPS	65
Solución de problemas de SNS	65
Usar Delivery Status en la consola de Amazon SNS	65
Prácticas recomendadas para utilizar el SDK de AWS Mobile para.NET y Xamarin	66
Biblioteca de documentación del servicio de AWS	66
Amazon Cognito Identity	27
Amazon Cognito Sync	3
Amazon Mobile Analytics	29
Simple Storage Service (Amazon S3)	67
Amazon DynamoDB	67
Amazon Simple Notification Service (SNS)	67
Otros enlaces útiles	67
Solución de problemas	68
Comprobar que un rol de IAM tiene los permisos requeridos	68

Uso de un depurador de proxy HTTP	69
Historial de revisión	70

La AWS EI SDK móvil para Xamarin se incluye ahora en el AWS SDK for .NET. En esta guía se hace referencia a la versión archivada del SDK para móviles para Xamarin.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.

¿Qué es el SDK de AWS Mobile para .NET y Xamarin?

La AWS Mobile SDK para Xamarin está incluido en el AWS SDK for .NET. Para obtener más información, consulte [AWS SDK for .NET Developer Guide](#).

Esta guía ya no se actualiza; hace referencia a la versión archivada del SDK móvil para Xamarin.

Guías y temas relacionados

- Para el desarrollo de aplicaciones front-end y móviles, recomendamos utilizar [AWS Amplify](#).
- Para consideraciones especiales sobre el uso de la AWS SDK for .NET para las aplicaciones de Xamarin, consulte [Consideraciones especiales para el soporte de Xamarin](#) en la AWS SDK for .NET Guía para desarrolladores.
- Para fines de referencia, puede encontrar la versión archivada del [AWS SDK para móviles para Xamarin](#) en GitHub.

Contenido de referencia archivado

El SDK de AWS Mobile para .NET y Xamarin contiene un conjunto de bibliotecas .NET, muestras de código y documentación para ayudar a los desarrolladores a crear aplicaciones móviles conectadas para:

- Xamarin iOS
- Xamarin Android
- Windows Phone Silverlight
- Windows RT 8.1
- Windows Phone 8.1

Las aplicaciones móviles escritas con AWS Mobile SDK for .NET and Xamarin llaman a API nativas de la plataforma de manera que tienen el aspecto y el funcionamiento de las aplicaciones nativas. Las bibliotecas .NET del SDK proporcionan contenedores C# para las API REST de AWS.

¿Qué se incluye en AWS Mobile SDK for .NET and Xamarin?

En la actualidad, los servicios de AWS compatibles son, entre otros:

- [Amazon Cognito](#)
- [Amazon S3](#)
- [Amazon DynamoDB](#)
- [Amazon Mobile Analytics](#)

- [Amazon Simple Notification Service](#)

Estos servicios le permiten autenticar a los usuarios, guardar datos del juego y de los jugadores, guardar objetos en la nube, recibir notificaciones de inserción y recopilar y analizar datos de uso.

AWS Mobile SDK for .NET and Xamarin también le permite utilizar la mayoría de los servicios de AWS compatibles con AWS SDK for .NET. Los servicios de AWS específicos del desarrollo de aplicaciones móviles se explican en esta guía para desarrolladores. Para obtener más información acerca de AWS SDK para .NET, consulte:

- [Guía de introducción a AWS SDK para .NET](#)
- [Guía para desarrolladores de AWS SDK para .NET](#)
- [Referencia de la API del SDK de AWS Mobile para .NET](#)

Compatibilidad

AWS Mobile SDK for .NET and Xamarin se suministra como una biblioteca de clases portable (PCL). La Support PCL se añadió en Xamarin.Android 4.10.1 y Xamarin.iOS 7.0.4. Los proyectos de Portable Library están integrados en Visual Studio.

IDE

Para obtener más información sobre el uso de IDE con la versión archivada del SDK de Xamarin, consulte [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#).

¿Cómo puedo obtener AWS Mobile SDK for .NET and Xamarin?

Para obtener AWS Mobile SDK for .NET and Xamarin, consulte [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#). El SDK de AWS Mobile para .NET y Xamarin se distribuye como NuGet paquetes. Puede encontrar una lista completa de los paquetes de servicios de AWS en [Packages de AWS SDK en NuGet](#) en AWS SDK for .NET [GitHub Repositorio](#).

Acerca de los servicios móviles de AWS

Amazon Cognito Identity

Todas llamadas efectuadas a AWS requieren credenciales de AWS. En vez de codificar de forma rígida las credenciales en sus aplicaciones, le recomendamos usar [Amazon Cognito Identity](#) para proporcionar las credenciales de AWS a su aplicación. Siga las instrucciones de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) para obtener las credenciales de AWS a través de Amazon Cognito.

Cognito también le permite autenticar usuarios mediante proveedores de inicio de sesión públicos como Amazon, Facebook, Twitter y Google, así como proveedores que admiten [OpenID Connect](#). Cognito también funciona con usuarios sin autenticar. Cognito proporciona credenciales temporales con derechos de acceso limitados que se especifican con un rol de [Identity and Access Management \(IAM\)](#). Cognito se configura mediante la creación de un grupo de identidades que está asociado a un rol de IAM. El rol de IAM especifica los recursos y servicios a los que puede obtener acceso la aplicación.

Para ver una introducción a Cognito Identity, consulte [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Para obtener más información acerca de Cognito Identity, consulte [Amazon Cognito Identity \(p. 27\)](#).

Amazon Cognito Sync

Cognito Sync es un servicio de AWS y una biblioteca de cliente que permite la sincronización entre dispositivos de los datos de usuarios relacionados con la aplicación. Puede utilizar la API de Cognito Sync para sincronizar los datos del perfil del usuario entre diversos dispositivos y proveedores de inicio de sesión, como Amazon, Facebook, Google y su propio proveedor de identidades personalizado.

Para ver una introducción a Cognito Sync, consulte [Sync User Data with Cognito Sync \(p. 12\)](#).

Para obtener más información acerca de Cognito Sync, consulte [Amazon Cognito Sync \(p. 28\)](#).

Mobile Analytics

Amazon Mobile Analytics le permite recopilar, visualizar y comprender el uso de sus aplicaciones móviles. Hay informes sobre métricas de usuarios activos, sesiones y retención, ingresos dentro de cada aplicación y eventos personalizados, y se pueden filtrar por plataforma y por rango de fechas. Amazon Mobile Analytics se ha diseñado para crecer con su negocio y puede recopilar y procesar miles de millones de eventos procedentes de muchos millones de puntos de enlace.

Para comenzar a utilizar Mobile Analytics, consulte [Tracking App Usage Data with Amazon Mobile Analytics \(p. 17\)](#).

Para obtener más información acerca de Mobile Analytics, consulte [Amazon Mobile Analytics \(p. 29\)](#).

Dynamo DB

Amazon DynamoDB es un servicio de base de datos no relacional rentable y rápido, de alta disponibilidad y de alta escalabilidad. DynamoDB elimina las limitaciones tradicionales de escalabilidad del almacenamiento de datos y, al mismo tiempo, mantiene una baja latencia y un desempeño previsible.

Para comenzar a utilizar DynamoDB, consulte [Almacenar y recuperar datos con DynamoDB \(p. 14\)](#).

Para obtener más información acerca de DynamoDB, consulte [Amazon DynamoDB \(p. 41\)](#).

Amazon Simple Notification Service

Amazon Simple Notification Service (SNS) es un servicio de notificaciones de inserción rápido, flexible y totalmente administrado que le permite enviar mensajes individuales o distribuir mensajes a un gran número de destinatarios. Amazon Simple Notification Service hace que enviar notificaciones de inserción a usuarios de dispositivos móviles o destinatarios de correo electrónico, o incluso enviar mensajes a otros servicios distribuidos, resulte sencillo y rentable.

Para comenzar a utilizar SNS con Xamarin iOS, consulte [Recibir notificaciones de inserción usando SNS \(Xamarin iOS\) \(p. 19\)](#).

Para comenzar a utilizar SNS con Xamarin Android, consulte [Recepción de notificaciones de inserción usando SNS \(Xamarin Android\) \(p. 22\)](#).

Para obtener más información acerca de SNS, consulte [Amazon Simple Notification Service \(SNS\) \(p. 54\)](#).

Configuración del SDK de AWS Mobile para .NET y Xamarin

Puede configurar el SDK de AWS Mobile para .NET y Xamarin y comenzar a crear un nuevo proyecto o puede integrar el SDK en un proyecto existente. También puede clonar y ejecutar las [muestras](#) para hacerse una idea del funcionamiento del SDK. Siga estos pasos para configurar y comenzar a usarlo AWS Mobile SDK for .NET and Xamarin.

Requisitos previos

Para poder utilizar el SDK de AWS Mobile para .NET y Xamarin, debe hacer lo siguiente:

- Crear una [cuenta de AWS](#).
- Instalar [Xamarin](#).

Una vez completados los requisitos previos:

1. Obtenga las credenciales de AWS a través de Amazon Cognito.
2. Defina los permisos necesarios para cada servicio de AWS que utilice en la aplicación.
3. Cree un nuevo proyecto en su IDE.
4. Instale el SDK de AWS Mobile para .NET y Xamarin.
5. Configurar el SDK de AWS Mobile para .NET y Xamarin.

Paso 1: Obtener las credenciales de AWS

Para realizar llamadas a AWS en su aplicación, primero debe obtener las credenciales de AWS. Para ello, utilizará Amazon Cognito, un servicio de AWS que permite a su aplicación obtener acceso a los servicios del SDK sin tener que integrar las credenciales de AWS privadas en la aplicación.

Para comenzar a utilizar Amazon Cognito, debe crear un grupo de identidades. Un grupo de identidades es un almacén de información específica de su cuenta y se identifica por medio de un ID de grupo de identidades único con un aspecto similar al siguiente:

```
"us-east-1:00000000-0000-0000-0000-000000000000"
```

1. Inicie sesión en la [consola de Amazon Cognito](#), elija Manage Federated Identities (Administrar identidades federadas) y después elija Create new identity pool (Crear nuevo grupo de identidades).
2. Escriba un nombre para el grupo de identidades y active la casilla de verificación para habilitar el acceso a las identidades sin autenticar. Elija Create Pool (Crear grupo) para crear un grupo de identidades.
3. Elija Allow (Permitir) para crear los dos roles predeterminados asociados a su grupo de identidades, uno para los usuarios sin autenticar y otro para los usuarios autenticados. Con estos roles predeterminados, se proporciona al grupo de identidades acceso a Amazon Cognito Sync y Amazon Mobile Analytics.

Normalmente se usa solo un grupo de identidades por aplicación.

Después de crear su grupo de identidades, obtendrá las credenciales de AWS mediante la creación de un objeto `CognitoAWSCredentials` (transfiriéndole su ID de grupo de identidades) y, a continuación, transfiriéndolo al constructor de un cliente de AWS como se indica a continuación:

```
CognitoAWSCredentials credentials = new CognitoAWSCredentials (
    "us-east-1:00000000-0000-0000-0000-000000000000", // Your identity pool ID
    RegionEndpoint.USEast1 // Region
);

// Example for |MA|
analyticsManager = MobileAnalyticsManager.GetOrCreateInstance(
    credentials,
    RegionEndpoint.USEast1, // Region
    APP_ID // app id
);
```

Paso 2: Establecer permisos

Debe definir los permisos de todos los servicios de AWS que desee utilizar en su aplicación. En primer lugar, debe comprender cómo ve AWS a los usuarios de su aplicación.

Cuando un usuario utiliza la aplicación y realiza llamadas a AWS, AWS le asigna una identidad. El grupo de identidades que creó en el paso 1 es la ubicación en la que AWS almacena estas identidades. Hay dos tipos de identidades: autenticadas y sin autenticar. Las identidades autenticadas pertenecen a los usuarios que se autentican con un proveedor de inicio de sesión público (por ejemplo, Facebook, Amazon, Google). Las identidades sin autenticar corresponden a los usuarios invitados.

Cada identidad está asociada a un rol de AWS Identity and Access Management. En el paso 1, creó dos roles de IAM, uno para los usuarios autenticados y otro para los usuarios sin autenticar. Cada rol de IAM tiene una o varias políticas adjuntas que especifican a qué servicios de AWS pueden obtener acceso las identidades asignadas a ese rol. Por ejemplo, la siguiente política de muestra concede acceso a un bucket de Amazon S3. :

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

Para configurar los permisos de los servicios de AWS que desea utilizar en su aplicación, modifique la política adjunta a los roles de.

1. Vaya a la [consola de IAM y elija Roles](#). Escriba el nombre del grupo de identidades en el campo de búsqueda. Elija el rol de IAM que desee configurar. Si su aplicación admite usuarios autenticados y no autenticados, debe conceder permisos a los dos roles.
2. Haga clic en **Attach Policy (Asociar política)**, seleccione la política que desee y haga clic en **Attach Policy (Asociar política)**. Las políticas predeterminadas de los roles de IAM que ha creado proporcionan acceso a Amazon Cognito Sync y Mobile Analytics.

Para obtener más información acerca de la creación de políticas o realizar una elección en una lista de políticas existentes, consulte [Políticas de IAM](#).

Paso 3: Crear un nuevo proyecto de

Windows

Puede utilizar Visual Studio para desarrollar su aplicación.

OS X

Puede utilizar Visual Studio para desarrollar sus aplicaciones. El desarrollo con Xamarin en iOS requiere acceso a un equipo Mac para ejecutar la aplicación. Para obtener más información, consulte [Installing Xamarin.iOS on Windows](#).

Note

El IDE comercial multiplataforma [Jinete](#) desde [JetBrains](#) incluye compatibilidad con Xamarin en plataformas Windows y Mac.

Paso 4: Instale el SDK de AWS Mobile para .NET y Xamarin

Windows

Opción 1: Instalación a través de la Consola de Package Manager

AWS Mobile SDK for .NET and Xamarin se compone de varios conjuntos de .NET. Para instalar el SDK de AWS Mobile para .NET y Xamarin, ejecute el comando del paquete de instalación para cada paquete en la consola de Package Manager. Por ejemplo, para instalar Cognito Identity, ejecute lo siguiente:

```
Install-Package AWSSDK.CognitoIdentity
```

Los paquetes de AWS Core Runtime y Amazon Cognito Identity son necesarios para todos los proyectos. A continuación se ofrece una lista completa de los nombres de los paquetes de cada servicio.

Servicio	Package name
AWS Core Runtime	AWSSDK.Core
Amazon Cognito Sync	AWSSDK.CognitoSync
Amazon Cognito Identity	AWSSDK.CognitoIdentity
Amazon DynamoDB	AWSSDK.DynamoDBv2
Amazon Mobile Analytics	AWSSDK.MobileAnalytics

Servicio	Package name
Simple Storage Service (Amazon S3)	AWSSDK.S3
Amazon SNS	AWSSDK.SimpleNotificationService (Servicio)

Para incluir un paquete preliminar, inserte el argumento de línea de comandos `-Pre` al instalar el paquete como se indica a continuación:

```
Install-Package AWSSDK.CognitoSync -Pre
```

Puede encontrar una lista completa de los paquetes de servicios de AWS en [Paquetes de SDK de AWS enNuGet](#) en el [AWS SDK para .NET GitHub Repositorio](#).

Opción 2: Instalar a través del IDE

En Visual Studio

1. Haga clic derecho en el proyecto y, a continuación, haga clic en `ManejarNuGetPaquetes`.
2. Busque el nombre del paquete que desee añadir a su proyecto. Para incluir el prelanzamiento `NuGetpaquetes`, elija `Incluir prelanzamiento`. Puede encontrar una lista completa de los paquetes de servicios de AWS en [Paquetes de SDK de AWS enNuGet](#).
3. Elija el paquete y, a continuación, haga clic en `Install (Instalar)`.

Mac (OS X)

En Visual Studio

1. Haga clic con el botón derecho en la carpeta de paquetes y elija `Add Packages`.
2. Busque el nombre del paquete que desee añadir a su proyecto. Para incluir el prelanzamiento `NuGetpaquetes`, elija `Mostrar paquetes de prelanzamiento`. Puede encontrar una lista completa de los paquetes de servicios de AWS en [Paquetes de SDK de AWS enNuGet](#).
3. Active la casilla de verificación situada junto al paquete que desea y elija `Add Package`.

Important

Si está desarrollando utilizando una biblioteca de clases portátil, también debe añadir el `AWSSDK.CoreNuGetpaquete` para todos los proyectos derivados de la biblioteca de clases portátil.

Paso 5: Configurar el SDK de AWS Mobile para .NET y Xamarin

Definición del registro

Para configurar los ajustes de registro, use la clase `Amazon.AWSConfigs` y la clase `Amazon.Util.LoggingConfig`. Puede encontrarlas en el conjunto `AWSSdk.Core`, disponible a través de Nuget Package Manager en Visual Studio. Puede insertar el código de configuración de registro en el método `OnCreate` del archivo `MainActivity.cs` para las aplicaciones de Android o el archivo

`AppDelegate.cs` para las aplicaciones de iOS. También debe añadir las declaraciones `using Amazon` y `using Amazon.Util` a los archivos `.cs`.

Configure los ajustes de registro como se indica a continuación:

```
var loggingConfig = AWSConfigs.LoggingConfig;
loggingConfig.LogMetrics = true;
loggingConfig.LogResponses = ResponseLoggingOption.Always;
loggingConfig.LogMetricsFormat = LogMetricsFormatOption.JSON;
loggingConfig.LogTo = LoggingOptions.SystemDiagnostics;
```

Cuando inicias sesión en `SystemDiagnostics`, el marco de trabajo imprime internamente la salida en `System.Console`. Si desea registrar las respuestas de HTTP, defina la marca `LogResponses`. Los valores pueden ser `Always`, `Never` o `OnError`.

También puede registrar las métricas de desempeño de las solicitudes HTTP mediante la propiedad `LogMetrics`. El formato de registro se puede especificar mediante la utilización de la propiedad `LogMetricsFormat`. Los valores válidos son `JSON` o `standard`.

Definición del punto de enlace de la región

Configure la región predeterminada de todos los clientes de servicios como se indica a continuación:

```
AWSConfigs.AWSRegion="us-east-1";
```

De este modo se establece la región predeterminada para todos los clientes de servicios en el SDK. Puede invalidar este ajuste especificando explícitamente la región en el momento de crear una instancia del cliente de servicio, como se indica a continuación:

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

Configuración de los ajustes del proxy HTTP

Si su red está detrás de un proxy, puede configurar el proxy para las solicitudes HTTP como se indica a continuación.

```
var proxyConfig = AWSConfigs.ProxyConfig;
proxyConfig.Host = "localhost";
proxyConfig.Port = 80;
proxyConfig.Username = "<username>";
proxyConfig.Password = "<password>";
```

Corrección del sesgo del reloj

Esta propiedad determina si el SDK debe corregir el sesgo del reloj del cliente determinando la hora correcta del servidor y reenviando la solicitud con la hora correcta.

```
AWSConfigs.CorrectForClockSkew = true;
```

Este campo se define si una llamada a un servicio ha generado una excepción y el SDK ha determinado que existe una diferencia entre la hora del servidor y la local.

```
var offset = AWSConfigs.ClockOffset;
```

Para obtener más información acerca del sesgo del reloj, consulte [Clock-skew Correction](#) en el Blog de AWS.

Pasos siguientes

Ahora que ha configurado el SDK de AWS Mobile para .NET y Xamarin, puede:

- Introducción. Lectura [Introducción al SDK de AWS Mobile para .NET y Xamarin \(p. 10\)](#) para obtener instrucciones de inicio rápido acerca de la manera de utilizar y configurar los servicios en AWS Mobile SDK for .NET and Xamarin.
- Explorar los temas del servicio. Obtenga más información acerca de cada servicio y de cómo funciona en el SDK de AWS Mobile para .NET y Xamarin.
- Ejecutar las demostraciones. Vea nuestras [aplicaciones de muestra de Xamarin](#), que ejemplifican casos de uso comunes. Para ejecutar las aplicaciones de muestra, configure el SDK de AWS Mobile para .NET y Xamarin como se ha descrito anteriormente y siga las instrucciones incluidas en los archivos README de cada muestra.
- Familiarizarse con las API. Ver el [sdk-xamarin-ref](#).
- Pregunte: Publicar preguntas en el [Foros de AWS Mobile SDK](#) [abrir un problema en GitHub](#).

Introducción al SDK de AWS Mobile para .NET y Xamarin

AWS Mobile SDK for .NET and Xamarin proporciona las bibliotecas, las muestras y la documentación que se necesitan para llamar a los servicios de AWS desde las aplicaciones de Xamarin.

Debe completar todas las instrucciones de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar a usar los servicios siguientes.

En estos temas de introducción se describe lo siguiente:

Temas

- [Almacenar y recuperar archivos en Amazon S3 \(p. 10\)](#)
- [Sincronización de datos de usuario con Cognito Sync \(p. 12\)](#)
- [Almacenar y recuperar datos con DynamoDB \(p. 14\)](#)
- [Seguimiento de los datos de uso de las aplicaciones con Amazon Mobile Analytics \(p. 17\)](#)
- [Recibir notificaciones de inserción usando SNS \(Xamarin iOS\) \(p. 19\)](#)
- [Recepción de notificaciones de inserción con SNS \(Xamarin Android\) \(p. 22\)](#)

Para obtener información acerca de otros SDK de AWS Mobile, consulte [SDK de AWS Mobile](#).

Almacenar y recuperar archivos en Amazon S3

Amazon Simple Storage Service (Amazon S3) proporciona a los desarrolladores de móviles un almacenamiento de objetos seguro, duradero y altamente escalable. Amazon S3 es fácil de usar e incluye una sencilla interfaz de servicios web que le permite almacenar y recuperar la cantidad de datos que desee desde cualquier ubicación de la Web.

En el siguiente tutorial se explica cómo integrar el S3TransferUtility, una utilidad de alto nivel para usar S3 con su aplicación. Para obtener más información sobre el uso de S3 en aplicaciones Xamarin, consulte [Amazon Simple Storage Service \(S3\) \(p. 33\)](#).

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

En este tutorial también se asume que ya ha creado un bucket de S3. Para crear un bucket de S3, consulte [S3 AWS Console](#).

Definición de permisos de S3

La política predeterminada de roles de IAM da a su aplicación acceso a Amazon Mobile Analytics y Amazon Cognito Sync. Para que el grupo de identidades de Cognito obtenga acceso a Amazon S3, debe modificar los roles del grupo de identidades.

1. Vaya a la [consola de Identity and Access Management](#) y haga clic en Roles (Roles) en el panel izquierdo.

2. Escriba el nombre del grupo de identidades en el campo de búsqueda. Se muestran dos roles: uno para los usuarios no autenticados y otro para los usuarios autenticados.
3. Haga clic en el rol de los usuarios no autenticados (el nombre del grupo de identidades tendrá añadido el texto "unauth").
4. Haga clic en Create Role Policy (Crear política de roles), seleccione Policy Generator (Generador de políticas) y haga clic en Select (Seleccionar).
5. En la página Edit Permissions (Editar permisos), escriba la configuración que se muestra en la siguiente imagen y reemplace el nombre de recurso de Amazon (ARN) por el suyo. El ARN de un bucket de S3 es parecido a `arn:aws:s3:::examplebucket/*` y está integrado por la región en la que se encuentra el bucket y por el nombre del bucket. La configuración que se muestra a continuación dará a su grupo de identidades pleno acceso a todas las acciones para el bucket especificado.

Edit Permissions

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

Effect: Allow ☒ Deny ☐

AWS Service: Amazon S3

Actions: All Actions Selected

Amazon Resource Name (ARN): arn:aws:s3:::examplebucket/*

[Add Conditions \(optional\)](#)

Add Statement

1. Haga clic en el botón Add Statement (Agregar instrucción) y, a continuación, haga clic en Next Step (Paso siguiente).
2. El asistente le mostrará la configuración que ha generado. Haga clic en Apply Policy (Aplicar política).

Para obtener más información acerca del procedimiento para conceder acceso a S3, consulte [Granting Access to an Amazon S3 Bucket](#).

AñadirNuGetPaquete para S3 para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para añadir el S3NuGetpaquete para su proyecto.

(opcional) Configuración de la versión de Signature para las solicitudes de S3

Toda interacción con Amazon S3 es o autenticada o anónima. AWS utiliza los algoritmos de Signature Version 4 o Signature Version 2 para autenticar las llamadas al servicio.

Las nuevas regiones de AWS creadas después de enero de 2014 admiten únicamente Signature Version 4. Sin embargo, muchas de las regiones más antiguas todavía admiten solicitudes de Signature Version 4 y Signature Version 2.

Si su bucket se encuentra en una región que no admite solicitudes de Signature Version 2, como las incluidas en la lista de [esta página](#), debe configurar el `AWSConfigsS3.UseSignaturePropiedad` Version4 a «true» así:

```
AWSConfigsS3.UseSignatureVersion4 = true;
```

Para obtener más información acerca de las versiones de AWS Signature, consulte [Authenticating Requests \(AWS Signature Version 4\)](#).

Inicializar el S3TransferUtilityCliente

Cree un cliente de S3 pasándole las credenciales del objeto de AWS y después pase el cliente de S3 a Transfer Utility como en el ejemplo:

```
var s3Client = new AmazonS3Client(credentials, region);  
var transferUtility = new TransferUtility(s3Client);
```

Cargar un archivo en Amazon S3

Para cargar un archivo en S3, llame a Upload en el objeto TransferUtility, pasando los siguientes parámetros:

- `file`: nombre de la cadena del archivo que desea cargar
- `bucketName`: nombre del bucket de S3 en el que se almacenará el archivo

```
transferUtility.Upload(  
    Path.Combine(Environment.SpecialFolder.ApplicationData, "file"),  
    "bucketName"  
);
```

En el código anterior se asume que existe un archivo en el directorio `Environment.SpecialFolder.ApplicationData`. Las cargas utilizan automáticamente la funcionalidad de carga multiparte de S3 en archivos grandes para mejorar el desempeño.

Descargar un archivo de Amazon S3

Para descargar un archivo de S3, llame a Download en el objeto TransferUtility, pasando los siguientes parámetros:

- `file`: nombre de la cadena del archivo que desea descargar
- `bucketName`: nombre de cadena del bucket de S3 del que quiere descargar el archivo
- `key`: cadena que representa el nombre del objeto S3 (un archivo en este caso) que se descargará

```
transferUtility.Download(  
    Path.Combine(Environment.SpecialFolder.ApplicationData, "file"),  
    "bucketName",  
    "key"  
);
```

Para obtener más información acerca del acceso a Amazon S3 desde aplicaciones Xamarin, consulte [Amazon Simple Storage Service \(S3\) \(p. 33\)](#).

Sincronización de datos de usuario con Cognito Sync

Amazon Cognito Sync facilita el almacenamiento de datos de usuarios móviles, tales como las preferencias de aplicaciones o el estado de los videojuegos, en AWS Cloud sin tener que escribir código backend ni

administrar infraestructuras. Puede guardar los datos en los dispositivos de los usuarios para que las aplicaciones puedan trabajar incluso cuando los dispositivos estén desconectados. Puede sincronizar los datos de los diferentes dispositivos de un usuario para que su experiencia con la aplicación sea homogénea, sea cual sea el dispositivo que use.

El siguiente tutorial explica cómo integrar Sync con su aplicación.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Permitir acceso a los recursos de Cognito Sync

La política asociada por defecto a los roles autenticados y no autenticados que creó durante la configuración permite que su aplicación pueda acceder Cognito Sync. No se necesitan más configuraciones.

AñadirNuGetPaquete para Cognito Sync para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para añadir el CognitoSyncManager NuGetpaquete para su proyecto.

Inicializar elCognitoSyncGerente

Pase el proveedor de credenciales de Amazon Cognito inicializado al constructor CognitoSyncManager:

```
CognitoSyncManager syncManager = new CognitoSyncManager (
    credentials,
    new AmazonCognitoSyncConfig {
        RegionEndpoint = RegionEndpoint.USEast1 // Region
    }
);
```

Sincronización de los datos de usuario

Para sincronizar datos de usuario sin autenticar:

1. Cree un conjunto de datos.
2. Añada los datos de usuario al conjunto de datos.
3. Sincronice el conjunto de datos con la nube.

Crear un conjunto de datos

Cree una instancia de Dataset. LaopenOrCreateEl método de conjunto de datos se utiliza para crear un nuevo conjunto de datos o abrir una instancia de un conjunto existente almacenado localmente en el dispositivo:

```
Dataset dataset = syncManager.OpenOrCreateDataset("myDataset");
```

Añadir los datos de usuario al conjunto de datos

Los datos de usuario se añaden en forma de pares clave/valor:

```
dataset.OnSyncSuccess += SyncSuccessCallback;  
dataset.Put("myKey", "myValue");
```

Los conjuntos de datos de Cognito funcionan como diccionarios, con valores accesibles por clave:

```
string myValue = dataset.Get("myKey");
```

Sincronizar conjuntos de datos

Para sincronizar un conjunto de datos, llame a su método de sincronización:

```
dataset.SynchronizeAsync();  
  
void SyncSuccessCallback(object sender, SyncSuccessEventArgs e) {  
    // Your handler code here  
}
```

Los datos escritos en conjuntos de datos se almacenarán localmente hasta que el conjunto de datos se haya sincronizado. En el código de esta sección se asume que usted está utilizando una identidad de Cognito sin autenticar, por lo que, cuando los datos de usuario se sincronicen con la nube, estos se almacenarán por dispositivo. El dispositivo tiene asociado un ID. Cuando los datos de usuario se sincronicen en la nube se asociarán con el ID de ese dispositivo.

Para obtener más información acerca de Cognito Sync, consulte [Amazon Cognito Sync \(p. 28\)](#).

Almacenar y recuperar datos con DynamoDB

[Amazon DynamoDB](#) es un servicio de base de datos no relacional rentable y rápido, de alta disponibilidad y de alta escalabilidad. DynamoDB elimina las limitaciones tradicionales de escalabilidad del almacenamiento de datos y, al mismo tiempo, mantiene una baja latencia y un desempeño previsible.

En el tutorial siguiente se explica cómo integrar el modelo de persistencia de objetos de DynamoDB con su aplicación, que almacena objetos en DynamoDB.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Creación de una tabla de DynamoDB

Antes de leer y escribir datos en una base de datos de DynamoDB, debe crear una tabla. Al crear una tabla, debe especificar la clave principal. La clave principal se compone de un atributo de hash y un atributo de rango opcional. Para obtener más información sobre cómo se usan los atributos principales y de rango, consulte [Uso de las tablas](#).

1. Vaya a la [consola de DynamoDB](#) y, a continuación, haga clic en Create Table (Crear tabla). Aparece el asistente para crear tablas.

2. Especifique el nombre de la tabla, el tipo de clave principal (Hash) y el nombre de atributo de ("Id"), tal como se muestra más abajo, y, a continuación, haga clic en Continue (Continuar):

Create Table [Cancel]

PRIMARY KEY | ADD INDEXES (optional) | PROVISIONED THROUGHPUT CAPACITY | ADDITIONAL OPTIONS (optional) | SUMMARY

Table Name: Books
Table will be created in us-east-1 region

Primary Key:
DynamoDB is a schema-less database. You only need to tell us your primary key attribute(s).

Primary Key Type: ☐ Hash and Range ☒ Hash

Hash Attribute Name: ☐ String ☐ Number ☐ Binary
Id

Warning: Choose a hash attribute that ensures that your workload is evenly distributed across hash keys.
For example, "Customer ID" is a good hash key, while "Game ID" would be a bad choice if most of your traffic relates to a few popular games.
[Learn more about choosing your primary key](#)

[Cancel] [Continue] [Help]

3. Deje vacíos los campos de edición en la siguiente pantalla y haga clic en Continue (Continuar).
4. Acepte los valores predeterminados de Read Capacity Units (Unidades de capacidad de lectura) y Write Capacity Units (Unidades de capacidad de escritura). Haga clic en Continue (Continuar).
5. En la pantalla siguiente escriba su dirección de correo electrónico en el cuadro de texto Send notification to: (Enviar notificación a:) y haga clic en Continue (Continuar). Aparece la pantalla de revisión.
6. Haga clic en Create (Crear). La tabla puede tardar unos minutos en crearse.

Definición de permisos para DynamoDB

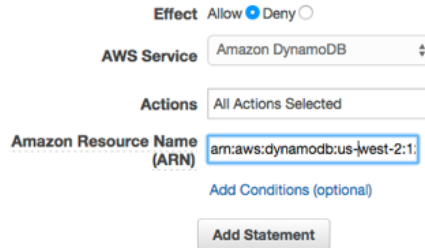
Para que el grupo de identidades obtenga acceso a Amazon DynamoDB, debe modificar los roles del grupo de identidades.

1. Vaya a la [consola de Identity and Access Management](#) y haga clic en Roles en el panel izquierdo. Busque el nombre del grupo de identidades; se mostrarán dos roles: uno para los usuarios autenticados y otro para los no autenticados.
2. Haga clic en el rol de los usuarios no autenticados (el nombre del grupo de identidades tendrá añadido el texto "unauth") y haga clic en Create Role Policy (Crear política de rol).
3. Seleccione Policy Generator (Generador de políticas) y haga clic en Select (Seleccionar).
4. En la página Edit Permissions (Editar permisos), escriba la configuración que se muestra en la imagen siguiente. El nombre de recurso de Amazon (ARN) de una tabla de DynamoDB es parecido a `arn:aws:dynamodb:us-west-2:123456789012:table/Books` y se compone de la región en la que se encuentra la tabla, el número de cuenta de AWS del propietario y el nombre de la tabla con

el formato `table/Books`. Para obtener más información sobre la especificación de ARN, consulte [Nombres de recurso de Amazon para DynamoDB](#).

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.



5. Haga clic en Add Statement (Añadir declaración) y, a continuación, en Next Step (Siguiendo paso). El asistente le mostrará la configuración generada.
6. Haga clic en Apply Policy (Aplicar política).

AñadirNuGetpaquete para DynamoDB para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para agregar el DynamoDBNuGetel paquete para su proyecto.

InicializarAmazonDynamoDBClient

Pase el proveedor de credenciales de Amazon Cognito inicializado y la región al constructor `AmazonDynamoDB`; a continuación, envíe el cliente a `DynamoDBContext`:

```
var client = new AmazonDynamoDBClient(credentials, region);
DynamoDBContext context = new DynamoDBContext(client);
```

Crear una clase

Para escribir una fila en la tabla, defina una clase que contenga los datos de fila. La clase también debe contener propiedades que contengan los datos de atributo de la fila y se mapeará a la tabla de DynamoDB creada en la consola. La siguiente declaración de clase ilustra una clase de este tipo:

```
[DynamoDBTable("Books")]
public class Book
{
    [DynamoDBHashKey] // Hash key.
    public int Id { get; set; }
    public string Title { get; set; }
    public string ISBN { get; set; }
    public int Price { get; set; }
    public string PageCount { get; set; }
    public string Author { get; set; }
}
```

Almacenamiento de un elemento

Para guardar un elemento, primero debe crear un objeto:

```
Book songOfIceAndFire = new Book()  
{  
    Id=1,  
    Title="Game Of Thrones",  
    ISBN="978-0553593716",  
    Price=4,  
    PageCount="819",  
    Author="GRRM"  
};
```

A continuación, guárdelo:

```
context.Save(songOfIceAndFire);
```

Para actualizar una fila, modifique la instancia de la clase `DDTableRow` y llame a `AWSDynamoObjectMapper.save()`, tal como se muestra más arriba.

Recuperación de un elemento

Recuperar un elemento con una clave principal:

```
Book retrievedBook = context.Load<Book>(1);
```

Actualización de un elemento

Para actualizar un elemento:

```
Book retrievedBook = context.Load<Book>(1);  
retrievedBook.ISBN = "978-0553593716";  
context.Save(retrievedBook);
```

Eliminación de un elemento

Para eliminar un elemento:

```
Book retrievedBook = context.Load<Book>(1);  
context.Delete(retrievedBook);
```

Para obtener más información sobre el acceso a DynamoDB desde una aplicación de Xamarin, consulte [Amazon DynamoDB \(p. 41\)](#).

Seguimiento de los datos de uso de las aplicaciones con Amazon Mobile Analytics

Amazon Mobile Analytics permite medir el uso y los ingresos de las aplicaciones. Mediante el seguimiento de tendencias clave como el número de usuarios nuevos y recurrentes, los ingresos de la aplicación, la retención de usuarios y los eventos personalizados de comportamiento dentro de la aplicación, puede tomar decisiones basadas en los datos para aumentar el interés y la monetización de su aplicación.

En el tutorial siguiente se explica el procedimiento para integrar Mobile Analytics con una aplicación.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Creación de una aplicación en la consola de Mobile Analytics

Vaya a la [consola de Amazon Mobile Analytics](#) y cree una aplicación. Tome nota del valor de `appId`, ya que lo necesitará posteriormente. Al crear una aplicación en la consola de Mobile Analytics, tendrá que especificar el ID de su grupo de identidades. Para obtener instrucciones sobre cómo crear un grupo de identidades consulte [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Para obtener más información acerca del trabajo en la consola, consulte [Amazon Mobile Analytics User Guide](#).

Definición de permisos para Mobile Analytics

La política predeterminada asociada a los roles que creó durante la configuración dan a su aplicación acceso a Mobile Analytics. No se necesitan más configuraciones.

AñadirNuGetPaquete para Mobile Analytics para un proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para agregar Mobile AnalyticsNuGetpaquete para un proyecto.

Configuración de Mobile Analytics

Mobile Analytics define algunos ajustes que se pueden configurar en el archivo `awsconfig.xml`:

```
var config = new MobileAnalyticsManagerConfig();
config.AllowUseDataNetwork = true;
config.DBWarningThreshold = 0.9f;
config.MaxDBSize = 5242880;
config.MaxRequestSize = 102400;
config.SessionTimeout = 5;
```

- `AllowUseDataNetwork`: valor booleano que especifica si los eventos de sesión se envían por la red de datos.
- `DBWarningThreshold`- límite del tamaño de la base de datos que, una vez alcanzado, generará registros de advertencia.
- `MaxDBSize`: tamaño de la base de datos de SQLite. Cuando la base de datos alcanza el tamaño máximo, se eliminan los eventos adicionales.
- `MaxRequestTamaño`: tamaño máximo de la solicitud en bytes que deben transmitirse en una solicitud HTTP al servicio Mobile Analytics.
- `SessionTimeout`: intervalo de tiempo transcurrido el cual una aplicación cambia a segundo plano y la sesión se puede terminar.

La configuración mostrada incluye los valores predeterminados para cada elemento de configuración.

InicializarMobileAnalyticsGerente

Para inicializarMobileAnalyticsGerente, llameGetOrCreateInstanceen las comidasMobileAnalyticsManager, transfiera las credenciales de AWS, su región, su ID de aplicación de Mobile Analytics y el objeto de configuración opcional:

```
var manager = MobileAnalyticsManager.GetOrCreateInstance(  
    "APP_ID",  
    "Credentials",  
    "RegionEndPoint",  
    config  
);
```

Seguimiento de eventos de la sesión

Xamarin Android

Invalide los métodos OnPause() y OnResume() de la actividad para registrar los eventos de la sesión.

```
protected override void OnResume()  
{  
    manager.ResumeSession();  
    base.OnResume();  
}  
  
protected override void OnPause()  
{  
    manager.PauseSession();  
    base.OnPause();  
}
```

Esta acción se debe implementar para cada actividad de la aplicación.

Xamarin iOS

En las comidasAppDelegate.cs:

```
public override void DidEnterBackground(UIApplication application)  
{  
    manager.PauseSession();  
}  
  
public override void WillEnterForeground(UIApplication application)  
{  
    manager.ResumeSession();  
}
```

Para obtener más información acerca de Mobile Analytics, consulte [Amazon Mobile Analytics \(p. 29\)](#).

Recibir notificaciones de inserción usando SNS (Xamarin iOS)

En este documento se explica cómo enviar notificaciones de inserción a una aplicación de Xamarin iOS con Amazon Simple Notification Service (SNS) y AWS Mobile SDK for .NET and Xamarin.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Establecer permisos de SNS

Siga el paso 2 de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) para adjuntar la política que se menciona a continuación a los roles de su aplicación. Esto otorgará a su aplicación los permisos correctos para obtener acceso a SNS:

1. Vaya a la [consola de IAM](#) y seleccione el rol de IAM que desea configurar.
2. ClicAdjuntar política, seleccione AmazonSNSFullAccesspolítica y haga clic enAdjuntar política.

Warning

UsarUse AmazonSNSFullAccessno se recomienda en un entorno de producción. Aquí lo utilizamos para que pueda comenzar a trabajar rápidamente. Para obtener más información acerca de la especificación de los permisos de un rol de IAM, consulte la [información general sobre los permisos de los roles de IAM](#).

Obtención de una suscripción a Apple iOS Developer Program

Tendrá que ejecutar su aplicación en un dispositivo físico para recibir notificaciones de inserción. Para ejecutar la aplicación en un dispositivo, debe estar suscrito a [Apple iOS Developer Program](#). Una vez que se haya suscrito, podrá utilizar Xcode para generar una identidad de firma. Para obtener más información, consulte la documentación [App Distribution Quick Start](#) de Apple.

Creación de un certificado de iOS

En primer lugar, debe crear un certificado de iOS. A continuación, debe crear un perfil de aprovisionamiento configurado para las notificaciones de inserción. Para ello:

1. Vaya a [Apple Developer Member Center](#) y haga clic en Certificates, Identifiers & Profiles.
2. Haga clic en Identifiers (Identificadores) en iOS Apps (Aplicaciones iOS), haga clic en el botón más situado en la esquina superior derecha de la página web para añadir un nuevo ID de aplicación de iOS y escriba una descripción de ID de aplicación.
3. Desplácese hasta la sección Add ID Suffix (Agregar sufijo de ID), seleccione Explicit App ID (ID de aplicación explícito) y escriba el identificador de su paquete.
4. Desplácese hacia abajo hasta la sección App Services (Servicios de aplicación) y seleccione Push Notifications (Notificaciones de inserción).
5. Haga clic en Continue (Continuar).
6. Haga clic en Submit.
7. Haga clic en Done (Hecho).
8. Seleccione el ID de aplicación que acaba de crear y, a continuación, haga clic en Edit.
9. Desplácese hasta la sección Push Notifications (Notificaciones de inserción). Haga clic en Create Certificate en Development SSL Certificate.
- 10Siga las instrucciones para crear una solicitud de firma del certificado (CSR), cargar la solicitud y descargar un certificado SSL que se utilizará para la comunicación con Apple Push Notification Service (APNS).
- 11.Vuelva a la página Certificates, Identifiers & Profiles. Haga clic en All en Provisioning Profiles.

- 12 Haga clic en el botón más situado en la esquina superior derecha para añadir un nuevo perfil de aprovisionamiento.
- 13 Seleccione iOS App Development y haga clic en Continue.
- 14 Seleccione su ID de aplicación y haga clic en Continue.
- 15 Seleccione su certificado de desarrollador y haga clic en Continue.
- 16 Seleccione su dispositivo y haga clic en Continue.
- 17 Escriba un nombre de perfil y haga clic en Generate.
- 18 Para instalar el perfil de aprovisionamiento, descargue el archivo de aprovisionamiento y haga doble clic en él.

Para obtener más información sobre el aprovisionamiento de un perfil configurado para notificaciones de inserción, consulte la documentación de Apple [Configuring Push Notifications](#).

Uso del certificado para crear un ARN de plataforma en la consola de SNS

1. Ejecute laKeyChainapp de acceso, seleccioneMy Certificatesen la parte inferior izquierda de la pantalla y, a continuación, haga clic con el botón derecho del ratón en el certificado SSL que ha generado para conectarse a APNS y seleccioneExportar. Se le pedirá que especifique un nombre para el archivo y una contraseña para proteger el certificado. El certificado se guardará en un archivo P12.
2. Vaya a la [consola de SNS](#) y haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.
3. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
4. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
5. Seleccione Apple Development (Desarrollo Apple) en Push notification platform (Pataforma de notificación de inserción).
6. Haga clic en Choose File (Elegir archivo) y seleccione el archivo P12 que creó al exportar el certificado SSL.
7. Escriba la contraseña que especificó cuando exportó el certificado SSL y haga clic en Load Credentials From File (Cargar credenciales desde archivo).
8. Haga clic en Create platform application (Crear aplicación de plataforma).
9. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación. Lo necesitará en los próximos pasos.

AñadirNuGetPaquete para SNS para su proyecto

Siga el paso 4 de las instrucciones de[Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#)para agregar Amazon Simple Notification ServiceNuGetpaquete para su proyecto.

Creación de un cliente de SNS

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

Registro de una aplicación para recibir notificaciones remotas

Para registrar una solicitud, llameRegisterForRemoteNotificationsen el objeto UIApplication, como se muestra a continuación. Coloque el siguiente código enAppDelegate.cs, insertando el ARN de la aplicación de plataforma donde se le indique:

```
public override bool FinishedLaunching(UIApplication app, NSDictionary options) {  
    // do something  
    var pushSettings = UIUserNotificationSettings.GetSettingsForTypes (  
        UIUserNotificationType.Alert |  
        UIUserNotificationType.Badge |  
        UIUserNotificationType.Sound,  
        null  
    );  
    app.RegisterUserNotifications(pushSettings);  
    app.RegisterForRemoteNotifications();  
    // do something  
    return true;  
}  
  
public override void RegisteredForRemoteNotifications(UIApplication application, NSData  
token) {  
    var deviceToken = token.Description.Replace("<", "").Replace(">", "").Replace(" ", "");  
    if (!string.IsNullOrEmpty(deviceToken)) {  
        //register with SNS to create an endpoint ARN  
        var response = await SnsClient.CreatePlatformEndpointAsync(  
            new CreatePlatformEndpointRequest {  
                Token = deviceToken,  
                PlatformApplicationArn = "YourPlatformArn" /* insert your platform application ARN  
here */  
            });  
    }  
}
```

Envío de un mensaje desde la consola de SNS a un punto de enlace

1. Vaya a la [consola de SNS](#) y haga clic en [Applications \(Aplicaciones\)](#).
2. Seleccione su aplicación de plataforma, seleccione un punto de enlace y haga clic en Publish to endpoint (Publicar en punto de enlace).
3. Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

Recepción de notificaciones de inserción con SNS (Xamarin Android)

En este tutorial se explica cómo enviar notificaciones de inserción a una aplicación de Xamarin Android con Amazon Simple Notification Service (SNS) y AWS Mobile SDK for .NET and Xamarin.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Establecer permisos de SNS

Siga el paso 2 de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) para adjuntar la política que se menciona a continuación a los roles de su aplicación. Esto otorgará a su aplicación los permisos correctos para obtener acceso a SNS:

1. Vaya a la [consola de IAM](#) y seleccione el rol de IAM que desea configurar.
2. ClicAdjuntar política, seleccione AmazonSNSFullAccesspolítica y haga clic enAdjuntar política.

Warning

UsarUse AmazonSNSFullAccessno es recomendable en un entorno de producción. Aquí lo utilizamos para que pueda comenzar a trabajar rápidamente. Para obtener más información acerca de la especificación de los permisos de un rol de IAM, consulte la [información general sobre los permisos de los roles de IAM](#).

Habilitar notificaciones de inserción en Google Cloud

En primer lugar, añada un nuevo proyecto de API de Google:

1. Vaya a la [Consola para desarrolladores de Google](#).
2. Haga clic en Create Project (Crear proyecto).
3. En el cuadro New Project (Nuevo proyecto), escriba un nombre de proyecto, tome nota del ID de proyecto (lo necesitará más adelante) y haga clic en Create (Crear).

A continuación, habilite el servicio Google Cloud Messaging (GCM) para el proyecto:

1. En la [consola para desarrolladores de Google](#), el nuevo proyecto ya debería estar seleccionado. En caso contrario, selecciónela en el menú desplegable situado en la parte superior de la página.
2. Seleccione APIs & auth (API y autenticación) en la barra lateral situada en la parte izquierda de la página.
3. En el cuadro de búsqueda, escriba "Google Cloud Messaging for Android" y haga clic en el enlace Google Cloud Messaging for Android (Google Cloud Messaging para Android).
4. Haga clic en Enable API (Habilitar API).

Por último, obtenga una clave de API:

1. En la consola de Google para desarrolladores, seleccione APIs & auth (API y autenticación) > Credentials (Credenciales).
2. En Public API access (Acceso a API pública), haga clic en Create new key (Crear nueva clave).
3. En el cuadro de diálogo Create a new key (Crear nueva clave), haga clic en Server key (Clave de servidor).
4. En el cuadro de diálogo resultante, haga clic en Create (Crear) y copie la clave de API que se muestra. Utilizará esta clave de API para realizar la autenticación más adelante.

Uso del ID de proyecto para crear un ARN de plataforma en la consola de SNS

1. Vaya a la [consola de SNS](#).
2. Haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.

3. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
4. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
5. Seleccione Google Cloud Messaging (GCM) en Push notification platform (Plataforma de notificación de inserción).
6. Pegue la clave de API en el cuadro de texto API key (Clave de API).
7. Haga clic en Create platform application (Crear aplicación de plataforma).
8. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación.

AñadirNuGetPaquete de SNS para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para agregar Amazon Simple Notification ServiceNuGetpaquete para su proyecto.

Creación de un cliente de SNS

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

Registro de una aplicación para recibir notificaciones remotas

Si desea registrarse para recibir notificaciones remotas en Android, tendrá que crear unBroadcastReceiverque puede recibir mensajes de Google Cloud. Cambie el nombre del paquete cuando se le pida a continuación:

```
[BroadcastReceiver(Permission = "com.google.android.c2dm.permission.SEND")]
[IntentFilter(new string[] {
    "com.google.android.c2dm.intent.RECEIVE"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
[IntentFilter(new string[] {
    "com.google.android.c2dm.intent.REGISTRATION"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
[IntentFilter(new string[] {
    "com.google.android.gcm.intent.RETRY"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
public class GCMBroadcastReceiver: BroadcastReceiver {
    const string TAG = "PushHandlerBroadcastReceiver";
    public override void OnReceive(Context context, Intent intent) {
        GCMIntentService.RunIntentInService(context, intent);
        SetResult(Result.Ok, null, null);
    }
}

[BroadcastReceiver]
[IntentFilter(new[] {
    Android.Content.Intent.ActionBootCompleted
})]
public class GCMBootReceiver: BroadcastReceiver {
    public override void OnReceive(Context context, Intent intent) {
```

SDK para móviles de AWS Guía
para desarrolladores de Xamarin
Registro de una aplicación para
recibir notificaciones remotas

```
        GCMIntentService.RunIntentInService(context, intent);  
        SetResult(Result.Ok, null, null);  
    }  
}
```

A continuación se puede ver el servicio que recibe la notificación de inserción del `BroadcastReceiver` y muestra la notificación en la barra de notificaciones del dispositivo:

```
[Service]  
public class GCMIntentService: IntentService {  
    static PowerManager.WakeLock sWakeLock;  
    static object LOCK = new object();  
  
    public static void RunIntentInService(Context context, Intent intent) {  
        lock(LOCK) {  
            if (sWakeLock == null) {  
                // This is called from BroadcastReceiver, there is no init.  
                var pm = PowerManager.FromContext(context);  
                sWakeLock = pm.NewWakeLock(  
                    WakeLockFlags.Partial, "My WakeLock Tag");  
            }  
        }  
  
        sWakeLock.Acquire();  
        intent.SetClass(context, typeof(GCMIntentService));  
        context.StartService(intent);  
    }  
  
    protected override void OnHandleIntent(Intent intent) {  
        try {  
            Context context = this.ApplicationContext;  
            string action = intent.Action;  
  
            if (action.Equals("com.google.android.c2dm.intent.REGISTRATION")) {  
                HandleRegistration(intent);  
            } else if (action.Equals("com.google.android.c2dm.intent.RECEIVE")) {  
                HandleMessage(intent);  
            }  
        } finally {  
            lock(LOCK) {  
                //Sanity check for null as this is a public method  
                if (sWakeLock != null) sWakeLock.Release();  
            }  
        }  
    }  
  
    private void HandleRegistration(Intent intent) {  
        string registrationId = intent.GetStringExtra("registration_id");  
        string error = intent.GetStringExtra("error");  
        string unregistration = intent.GetStringExtra("unregistered");  
  
        if (string.IsNullOrEmpty(error)) {  
            var response = await SnsClient.CreatePlatformEndpointAsync(new  
                CreatePlatformEndpointRequest {  
                    Token = registrationId,  
                    PlatformApplicationArn = "YourPlatformArn" /* insert your platform application ARN  
here */  
                });  
        }  
    }  
  
    private void HandleMessage(Intent intent) {  
        string message = string.Empty;  
        Bundle extras = intent.Extras;
```


SDK para móviles de AWS Guía
para desarrolladores de Xamarin
Envío de un mensaje desde la
consola de SNS a un punto de enlace

```
if (!string.IsNullOrEmpty(extras.GetString("message"))) {  
    message = extras.GetString("message");  
} else {  
    message = extras.GetString("default");  
}  
  
Log.Info("Messages", "message received = " + message);  
ShowNotification(this, "SNS Push", message);  
//show the message  
  
}  
  
public void ShowNotification(string contentTitle,  
string contentText) {  
    // Intent  
    Notification.Builder builder = new Notification.Builder(this)  
        .SetContentTitle(contentTitle)  
        .SetContentText(contentText)  
        .SetDefaults(NotificationDefaults.Sound | NotificationDefaults.Vibrate)  
        .SetSmallIcon(Resource.Drawable.Icon)  
        .SetSound(RingtoneManager.GetDefaultUri(RingtoneType.Notification));  
  
    // Get the notification manager:  
    NotificationManager notificationManager =  
this.GetSystemService(Context.NotificationService) as NotificationManager;  
  
    notificationManager.Notify(1001, builder.Build());  
}  
}
```

Envío de un mensaje desde la consola de SNS a un punto de enlace

1. Vaya a la [consola de SNS](#) y haga clic en [Applications \(Aplicaciones\)](#).
2. Seleccione su aplicación de plataforma, seleccione un punto de enlace y haga clic en Publish to endpoint (Publicar en punto de enlace).
3. Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

Identidad de Amazon Cognito

¿Qué es Amazon Cognito Identity?

Amazon Cognito Identity le permite crear identidades exclusivas para sus usuarios y autenticarlos mediante proveedores de identidades. Con una identidad, puede obtener credenciales de AWS temporales con privilegios limitados para sincronizar datos con Amazon Cognito Sync u obtener acceso directo a otros servicios de AWS. Amazon Cognito Identity es compatible con proveedores de identidades públicos (Amazon, Facebook y Google) así como con identidades sin autenticar. También es compatible con las identidades autenticadas de desarrollador, que permiten registrar y autenticar a los usuarios mediante su propio proceso de autenticación backend.

Para obtener más información sobre Cognito Identity, consulte la [Guía para desarrolladores de Amazon Cognito](#).

Para obtener información acerca de la disponibilidad regional de Cognito Authentication, consulte la [disponibilidad regional de los servicios de AWS](#).

Uso de un proveedor público para autenticar a los usuarios

Con Amazon Cognito Identity puede crear identidades exclusivas para los usuarios y autenticarlos para un acceso seguro a los recursos de AWS, como Amazon S3 o Amazon DynamoDB. Amazon Cognito Identity es compatible con proveedores de identidades públicos (Amazon, Facebook, Twitter/Digits, Google o cualquier proveedor compatible con OpenID Connect) así como con identidades sin autenticar.

Para obtener información sobre el uso de proveedores de identidades públicos como Amazon, Facebook, Twitter/Digits o Google para autenticar usuarios, consulte [Proveedores externos](#) en la Guía para desarrolladores de Amazon Cognito.

Uso de identidades autenticadas por el desarrollador

Amazon Cognito es compatible con las identidades autenticadas por el desarrollador, además de ser compatible con la identidad federada web mediante Facebook, Google y Amazon. Con las identidades autenticadas por el desarrollador, sigue pudiendo registrar y autenticar usuarios mediante su propio proceso de autenticación, sin dejar de usar [Amazon Cognito Sync \(p. 28\)](#) para sincronizar los datos de los usuarios y obtener acceso a los recursos de AWS. El uso de las identidades autenticadas por el desarrollador implica una interacción entre el dispositivo del usuario final, el backend para la autenticación y Amazon Cognito.

Para obtener información sobre las identidades autenticadas, consulte [Identidades autenticadas por el desarrollador](#) en la Guía para desarrolladores de Amazon Cognito.

Amazon Cognito Sync

¿Qué es Amazon Cognito Sync?

Cognito Sync es un servicio de AWS y una biblioteca de cliente que permite la sincronización entre varios dispositivos de los datos del usuario (por ejemplo, las puntuaciones de los juegos, las preferencias del usuario o el estado de los juegos). Puede utilizar la API de Cognito Sync para sincronizar los datos del usuario entre los distintos dispositivos. Para utilizar Cognito Sync en su aplicación, debe incluir | en su proyecto.

Para obtener más información acerca de la integración de Amazon Cognito Sync en su aplicación, consulte [Amazon Cognito Sync Developer Guide](#).

Amazon Mobile Analytics

[Amazon Mobile Analytics](#) es un servicio para recopilar, visualizar, comprender y extraer los datos de uso de las aplicaciones a gran escala. Mobile Analytics captura fácilmente tanto los datos de dispositivos estándar como los eventos personalizados y genera informes automáticamente. Además de los informes agregados que se indican a continuación, también puede establecer que los datos se exporten automáticamente a Redshift y S3 para seguir analizándolos.

Amazon Mobile Analytics le permite realizar un seguimiento del comportamiento de los clientes, agrupar métricas, generar visualizaciones de datos e identificar patrones significativos.

Conceptos clave

Tipos de informes

A diferencia del resto, Mobile Analytics proporciona los siguientes informes en la consola de Mobile Analytics:

- Daily Active Users (DAU), Monthly Active Users (MAU) y New Users
- Sticky Factor (DAU dividido por MAU)
- Session Count y Average Sessions per Daily Active User
- Average Revenue per PAU (ARPPDAU) y Average Revenue per Paid DAU (ARPPDAU)
- Day 1, Day 3 y Day 7 Retention y Week 1, Week 2 y Week 3 Retention
- Custom Events

Estos informes se proporcionan a través de seis pestañas de informes en la consola:

- Resumen— Realizar un seguimiento de nueve informes preseleccionados en un simple-to-review panel de control para obtener una idea rápida de la participación: MAU, DAU, New Users, Daily Sessions, Sticky Factor, 1-Day Retention, ARPPDAU, Daily Sessions, Sticky Factor, DAU.
- Active Users: realice un seguimiento de la cantidad de usuarios conectados a la aplicación diariamente y monitoree el factor de adhesión para calibrar el compromiso, el atractivo y la monetización.
- Sessions: realice un seguimiento de la frecuencia de uso de la aplicación en un determinado día y de la frecuencia con que cada usuario la abre en un día.
- Retention: realice un seguimiento de la velocidad a la que los clientes vuelven a la aplicación a diario y semanalmente.
- Revenue: realice un seguimiento de las tendencias de ingresos de la aplicación para identificar áreas de mejora en cuanto a monetización.
- Eventos personalizados: realice un seguimiento de acciones personalizadas definidas por el usuario específicas de su aplicación.

Para obtener más información acerca de los informes de Mobile Analytics y trabajar en la consola de Mobile Analytics, consulte [Mobile Analytics Console Reports Overview](#) en la Guía para desarrolladores de Mobile Analytics.

Configuración del proyecto

Requisitos previos

Para utilizar Mobile Analytics en su aplicación, tendrá que añadir el SDK al proyecto. Para ello, siga las instrucciones que se describen en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Configuración de Mobile Analytics

Mobile Analytics define algunos ajustes que se pueden configurar en el archivo `awsconfig.xml`:

```
var config = new MobileAnalyticsManagerConfig();
config.AllowUseDataNetwork = true;
config.DBWarningThreshold = 0.9f;
config.MaxDBSize = 5242880;
config.MaxRequestSize = 102400;
config.SessionTimeout = 5;
```

- **SessionTimeout**- Si la aplicación permanece en segundo plano durante un tiempo superior al `SessionTimeout` continuación, el cliente de Mobile Analytics termina la sesión actual y crea otra cuando la aplicación vuelve a primer plano. Le recomendamos que utilice valores entre 5 y 10. El valor predeterminado es 5.
- **MaxDBSize**: tamaño máximo de la base de datos (en bytes) utilizados para el almacenamiento local de eventos. Si el tamaño de la base de datos supera este valor, los eventos adicionales se ignorarán. Le recomendamos que utilice valores entre 1 MB y 10 MB. El valor predeterminado es 5242880 (5 MB).
- **DBWarningThreshold**- El umbral de advertencia. Los valores válidos oscilan entre 0 y 1. Si los valores superan el umbral de advertencia se generarán nuevos registros. El valor predeterminado es 0.9.
- **MaxRequestSize (Tamaño)**- Tamaño máximo de una solicitud HTTP realizada al servicio Mobile Analytics. El valor se especifica en bytes y puede oscilar entre 1-512 KB. El valor predeterminado es 102400 (100 KB). No utilice valores mayores de 512 KB, porque podría provocar que el servicio rechazara la solicitud HTTP.
- **AllowUseDataNetwork**- Valor que indica si se permiten llamadas de servicio a través de una red móvil. Utilice esta opción con precaución ya que puede aumentar el uso de los datos del cliente.

La configuración mostrada incluye los valores predeterminados para cada elemento de configuración.

Integración de Mobile Analytics con su aplicación

En las secciones siguientes se explica el procedimiento para integrar Mobile Analytics con una aplicación.

Creación de una aplicación en la consola de Mobile Analytics

Vaya a la [consola de Amazon Mobile Analytics](#) y cree una aplicación. Tome nota del valor de `appId`, ya que lo necesitará posteriormente. Al crear una aplicación en la consola de Mobile Analytics, tendrá que especificar el ID de su grupo de identidades. Para obtener instrucciones sobre cómo crear un grupo de identidades consulte [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Para obtener más información sobre el funcionamiento de la consola de Mobile Analytics, consulte [Mobile Analytics Console Reports Overview](#) en la Guía para desarrolladores de Mobile Analytics.

Creación de unMobileAnalyticsCliente manager

Para inicializarMobileAnalyticsGerente, llameGetOrCreateInstanceen las comidasmobileAnalyticsManager, transfiera las credenciales de AWS, su región, su ID de aplicación de Mobile Analytics y el objeto de configuración opcional:

```
// Initialize the MobileAnalyticsManager
analyticsManager = MobileAnalyticsManager.GetInstance(
    cognitoCredentials,
    RegionEndpoint.USEast1,
    APP_ID,
    config
);
```

El APP_ID se genera durante la ejecución del asistente de creación de aplicaciones. Ambos valores deben coincidir con los de la consola de Mobile Analytics. El valor de APP_ID se utiliza para agrupar los datos en la consola de Mobile Analytics. Para encontrar el ID de la aplicación después de crearla en la consola de Mobile Analytics, vaya a la consola de Mobile Analytics y haga clic en el icono del engranaje situado en la esquina superior derecha de la pantalla. Se mostrará la página App Management, que incluye una lista de todas las aplicaciones registradas y los ID de aplicación correspondientes.

Registro de eventos de monetización

El SDK de AWS Mobile para .NET y Xamarin proporciona elMonetizationEvent, que permite generar eventos de monetización para realizar un seguimiento de las compras realizadas en aplicaciones móviles. El siguiente fragmento de código muestra cómo crear un evento de monetización:

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

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

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

Registro de eventos personalizados

Mobile Analytics le permite definir eventos personalizados. Los eventos personalizados están íntegramente definidos por usted y le ayudan a realizar un seguimiento de las acciones de los usuarios específicas de su aplicación o su juego. Para obtener más información sobre eventos personalizados, consulte [Eventos personalizados](#).

En este ejemplo, vamos a suponer que la aplicación es un juego y que queremos registrar un evento cuando un usuario complete un nivel. Crea un«LevelComplete» evento mediante la creación de un nuevoAmazonMobileAnalyticsEventinstancia:

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

// Add attributes
customEvent.AddAttribute("LevelName", "Level1");
```

```
customEvent.AddAttribute("CharacterClass", "Warrior");  
customEvent.AddAttribute("Successful", "True");  
  
// Add metrics  
customEvent.AddMetric("Score", 12345);  
customEvent.AddMetric("TimeInLevel", 64);  
  
// Record the event  
analyticsManager.RecordEvent(customEvent);
```

Registro de sesiones

Xamarin iOS

Cuando la aplicación deja de ser el elemento activo, puede hacer una pausa en la sesión. Para aplicaciones iOS, en `AppDelegate` archivo.cs, anular `DidEnterBackground` y `WillEnterForeground` para llamar `AMobileAnalyticsManager.PauseSession` y `AMobileAnalyticsManager.ResumeSession` y como se muestra en el siguiente fragmento de código:

```
public override void DidEnterBackground(UIApplication application)  
{  
    // ...  
    _manager.PauseSession();  
    // ...  
}  
  
public override void WillEnterForeground(UIApplication application)  
{  
    // ...  
    _manager.ResumeSession();  
    // ...  
}
```

Xamarin Android

Para aplicaciones Android llame `AMobileAnalyticsManager.PauseSession` en `OnPause()` y `AMobileAnalyticsManager.ResumeSession` en `OnResume()`, como se muestra en el siguiente fragmento de código:

```
protected override void OnResume()  
{  
    _manager.ResumeSession();  
    base.OnResume();  
}  
  
protected override void OnPause()  
{  
    _manager.PauseSession();  
    base.OnPause();  
}
```

De forma predeterminada, si el usuario cambia el foco a otro elemento distinto de la aplicación durante menos de 5 segundos y vuelve a la aplicación, la sesión se reanuda. Si el usuario cambia el foco a un elemento distinto de la aplicación durante 5 segundos o más, se crea una nueva sesión. Se puede configurar este ajuste en el archivo de configuración `aws_mobile_analytics.json` estableciendo la propiedad `"SESSION_DELTA"` en el número de segundos que se debe esperar para crear una sesión.

Amazon Simple Storage Service (S3)

¿Qué es S3?

[Amazon Simple Storage Service \(Amazon S3\)](#) proporciona a los desarrolladores un almacenamiento de objetos seguro, duradero y con un alto grado de escalabilidad. Amazon S3 es fácil de usar e incluye una sencilla interfaz de servicios web que le permite almacenar y recuperar la cantidad de datos que desee desde cualquier ubicación de la Web. En Amazon S3, solo se paga el espacio de almacenamiento que realmente se usa. No hay cuota mínima ni costos de configuración.

Amazon S3 proporciona almacenamiento de objetos económico para una amplia variedad de casos de uso, entre los que se incluyen las aplicaciones en la nube, la distribución de contenido, las operaciones de backup y archivado, la recuperación de desastres y el análisis de big data.

Para obtener información acerca de la disponibilidad regional de AWS S3, consulte la [disponibilidad regional de los servicios de AWS](#).

Conceptos clave

Bucket

Todos los objetos almacenados en Amazon S3 residen en un bucket. Puede utilizar los buckets para agrupar objetos relacionados del mismo modo en que usa un directorio para agrupar archivos en un sistema de archivos. Los buckets tienen propiedades, como, por ejemplo, permisos de acceso y estado de control de versiones, y se puede especificar la región en la que deben residir.

Para obtener más información acerca de los buckets de S3, consulte [Working with Buckets](#) en la Guía para desarrolladores de S3.

Objetos

Los objetos son los datos que se almacenan en Amazon S3. Todos los objetos residen en un bucket que se crea en cada región de AWS.

Los objetos almacenados en una región nunca la abandonan, a menos que se transfieran expresamente a otra región. Por ejemplo, los objetos almacenados en la región UE (Irlanda) nunca salen de ella. Los objetos almacenados en una región de Amazon S3 permanecen físicamente en esa región. Amazon S3 no almacena copias ni las mueve a ninguna otra región. Sin embargo, puede obtener acceso a los objetos desde cualquier lugar, siempre y cuando disponga de los permisos necesarios.

Los objetos pueden tener cualquier tipo de archivo: imágenes, datos de backup, películas, etc. Un objeto puede tener hasta 5 TB. Puede haber un número ilimitado de objetos en un bucket.

Para poder cargar un objeto en Amazon S3, debe disponer de permisos de escritura en un bucket. Para obtener más información acerca de la definición de los permisos de un bucket, consulte [Editing Bucket Permissions](#) en la Guía para desarrolladores de S3.

Para obtener más información acerca de los objetos de S3, consulte [Working with Objects](#) en la Guía para desarrolladores de S3.

Metadatos de objetos

Cada objeto de Amazon S3 tiene un conjunto de pares de clave-valor que representa sus metadatos. Existen dos tipos de metadatos:

- Metadatos del sistema: procesados en algunos casos por Amazon S3, por ejemplo, Content-Type y Content-Length.
- Metadatos del usuario: nunca se procesan en Amazon S3. Los metadatos del usuario se almacenan con el objeto y se devuelven con este. El tamaño máximo de los metadatos del usuario es de 2 KB, y tanto las claves como sus valores deben cumplir con los estándares ASCII de EE. UU.

Para obtener más información acerca de los metadatos de objetos de S3, consulte [Editing Object Metadata](#).

Configuración del proyecto

Requisitos previos

Para utilizar Amazon S3 en su aplicación, tendrá que añadir el SDK al proyecto. Para ello, siga las instrucciones que se describen en [Setting Up the AWS Mobile SDK for .NET and Xamarin](#) (p. 4).

Creación de S3 Bucket

Amazon S3 almacena los recursos de su aplicación en los buckets de Amazon S3, unos contenedores de almacenamiento en la nube que residen en una [región](#) concreta. Cada bucket de Amazon S3 debe tener un nombre único. Puede utilizar la [consola de Amazon S3](#) para crear un bucket.

1. Inicie sesión en la [consola de Amazon S3](#) y haga clic en Create Bucket (Crear bucket).
2. Escriba un nombre de bucket, seleccione una región y haga clic en Create (Crear).

Definición de permisos de S3

La política predeterminada de roles de IAM da a su aplicación acceso a Amazon Mobile Analytics y Amazon Cognito Sync. Para que el grupo de identidades de Cognito obtenga acceso a Amazon S3, debe modificar los roles del grupo de identidades.

1. Vaya a la [consola de Identity and Access Management](#) y haga clic en Roles (Roles) en el panel izquierdo.
2. Escriba el nombre del grupo de identidades en el campo de búsqueda. Se muestran dos roles: uno para los usuarios no autenticados y otro para los usuarios autenticados.
3. Haga clic en el rol de los usuarios no autenticados (el nombre del grupo de identidades tendrá añadido el texto "unauth").
4. Haga clic en Create Role Policy (Crear política de roles), seleccione Policy Generator (Generador de políticas) y haga clic en Select (Seleccionar).
5. En la página Edit Permissions (Editar permisos), escriba la configuración que se muestra en la siguiente imagen y reemplace el nombre de recurso de Amazon (ARN) por el suyo. El ARN de un bucket de S3 es parecido a `arn:aws:s3:::examplebucket/*` y está integrado por la región en la que se encuentra el bucket y por el nombre del bucket. La configuración que se muestra a continuación dará a su grupo de identidades pleno acceso a todas las acciones para el bucket especificado.

Edit Permissions

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

Effect ☒ Allow ☐ Deny

AWS Service

Actions

Amazon Resource Name (ARN)

[Add Conditions \(optional\)](#)

1. Haga clic en el botón Add Statement (Agregar instrucción) y, a continuación, haga clic en Next Step (Paso siguiente).
2. El asistente le mostrará la configuración que ha generado. Haga clic en Apply Policy (Aplicar política).

Para obtener más información acerca del procedimiento para conceder acceso a S3, consulte [Granting Access to an Amazon S3 Bucket](#).

(opcional) Configuración de la versión de Signature para las solicitudes de S3

Toda interacción con Amazon S3 es o autenticada o anónima. AWS utiliza los algoritmos de Signature Version 4 o Signature Version 2 para autenticar las llamadas al servicio.

Las nuevas regiones de AWS creadas después de enero de 2014 admiten únicamente Signature Version 4. Sin embargo, muchas de las regiones más antiguas todavía admiten solicitudes de Signature Version 4 y Signature Version 2.

Si su bucket se encuentra en una región que no es compatible con solicitudes de Signature Version 2 según la lista de [esta página](#), debe establecer el `AWSSignatureVersion4` a «true» así:

```
AWSSignatureVersion4 = true;
```

Para obtener más información acerca de las versiones de AWS Signature, consulte [Authenticating Requests \(AWS Signature Version 4\)](#).

Integración de S3 en una aplicación

Hay dos formas de interactuar con S3 en una aplicación de Xamarin. Los dos métodos se exploran de manera detallada en los siguientes temas:

Utilizar S3 Transfer Utility

S3 Transfer Utility facilita la carga y descarga de archivos a S3 desde la aplicación Xamarin.

Inicializar elTransferUtility

Cree un cliente de S3 pasándole las credenciales del objeto de AWS y después pase el cliente de S3 a Transfer Utility como en el ejemplo:

```
var s3Client = new AmazonS3Client(credentials,region);  
var transferUtility = new TransferUtility(s3Client);
```

Configurar elTransferUtility

Existen tres propiedades opcionales que puede configurar:

- **ConcurrentServiceSolicitudes:** determina la cantidad de subprocesos activos o el número de solicitudes web asíncronas simultáneas que se utilizarán para cargar/descargar el archivo. El valor predeterminado es 10.
- **MinSizeBeforePartCargar:** obtiene o establece el tamaño mínimo de parte para cargar partes en bytes. El valor predeterminado es 16 MB. La reducción del tamaño mínimo de parte hace que las cargas multiparte se dividan en un mayor número de partes más pequeñas. Establecer un valor demasiado bajo repercute negativamente en la velocidad de transferencia, lo que provoca una latencia y una comunicación adicionales para cada parte.
- **NumberOfUploadThreads:** obtiene o establece el número de subprocesos que se ejecutarán. Esta propiedad determina la cantidad de subprocesos activos que se utilizarán para cargar el archivo. El valor de predeterminado es de 10 subprocesos.

Para configurar el S3TransferUtilitycliente, cree un objeto de configuración, defina las propiedades, y pase el objeto a suTransferUtilityconstructor así:

```
var config = new TransferUtilityConfig();  
  
config.ConcurrentServiceRequests = 10;  
config.MinSizeBeforePartUpload=16*1024*1024;  
config.NumberOfUploadThreads=10;  
  
var s3Client = new AmazonS3Client(credentials);  
var utility = new TransferUtility(s3Client,config);
```

Descarga de un archivo

Para descargar un archivo de S3, llame a `Download` en el objeto `TransferUtility`, pasando los siguientes parámetros:

- **file:** nombre de la cadena del archivo que desea descargar
- **bucketName:** nombre de cadena del bucket de S3 del que quiere descargar el archivo
- **key:** cadena que representa el nombre del objeto S3 (un archivo en este caso) que se descargará

```
transferUtility.Download(  
    Path.Combine(Environment.SpecialFolder.ApplicationData,"file"),  
    "bucketName",  
    "key"  
);
```

Cargar un archivo

Para cargar un archivo en S3, llame a `Upload` en el objeto `TransferUtility`, pasando los siguientes parámetros:

- `file`: nombre de la cadena del archivo que desea cargar
- `bucketName`: nombre del bucket de S3 en el que se almacenará el archivo

```
transferUtility.Upload(  
    Path.Combine(Environment.SpecialFolder.ApplicationData, "file"),  
    "bucketName"  
);
```

En el código anterior se asume que existe un archivo en el directorio `Environment.SpecialFolder.ApplicationData`. Las cargas utilizan automáticamente la funcionalidad de carga multiparte de S3 en archivos grandes para mejorar el desempeño.

Uso de las API de nivel de servicio de S3

Además de utilizar el `S3TransferUtility`, puede interactuar con S3 utilizando las API de S3 de bajo nivel.

Inicialización del cliente de Amazon S3

Para utilizar Amazon S3, primero debe crear una instancia de `AmazonS3Client` que incluya una referencia a `CognitoAWSCredentials` instancia que creaste anteriormente y tu región:

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

Descarga de un archivo

Para descargar un archivo desde S3:

```
// Create a GetObject request  
GetObjectRequest request = new GetObjectRequest  
{  
    BucketName = "SampleBucket",  
    Key = "Item1"  
};  
  
// Issue request and remember to dispose of the response  
using (GetObjectResponse response = client.GetObject(request))  
{  
    using (StreamReader reader = new StreamReader(response.ResponseStream))  
    {  
        string contents = reader.ReadToEnd();  
        Console.WriteLine("Object - " + response.Key);  
        Console.WriteLine(" Version Id - " + response.VersionId);  
        Console.WriteLine(" Contents - " + contents);  
    }  
}
```

Cargar un archivo

Para cargar un archivo en S3:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Create a PutObject request
PutObjectRequest request = new PutObjectRequest
{
    BucketName = "SampleBucket",
    Key = "Item1",
    FilePath = "contents.txt"
};

// Put object
PutObjectResponse response = client.PutObject(request);
```

Eliminación de un elemento

Para eliminar un elemento en S3:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Create a DeleteObject request
DeleteObjectRequest request = new DeleteObjectRequest
{
    BucketName = "SampleBucket",
    Key = "Item1"
};

// Issue request
client.DeleteObject(request);
```

Eliminación de varios elementos

Para eliminar varios objetos de un bucket con una única solicitud HTTP:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Create a DeleteObject request
DeleteObjectsRequest request = new DeleteObjectsRequest
{
    BucketName = "SampleBucket",
    Objects = new List<KeyVersion>
    {
        new KeyVersion() {Key = "Item1"},
        // Versioned item
        new KeyVersion() { Key = "Item2", VersionId =
            "Rej8CiBxcZKVK81cLr39j27Y5FVXghDK", },
        // Item in subdirectory
        new KeyVersion() { Key = "Logs/error.txt"}
    }
};

try
{
    // Issue request
    DeleteObjectsResponse response = client.DeleteObjects(request);
}
catch (DeleteObjectsException doe)
{
}
```

```
// Catch error and list error details
DeleteObjectsResponse errorResponse = doe.Response;

foreach (DeletedObject deletedObject in errorResponse.DeletedObjects)
{
    Console.WriteLine("Deleted item " + deletedObject.Key);
}
foreach (DeleteError deleteError in errorResponse.DeleteErrors)
{
    Console.WriteLine("Error deleting item " + deleteError.Key);
    Console.WriteLine(" Code - " + deleteError.Code);
    Console.WriteLine(" Message - " + deleteError.Message);
}
}
```

Puede especificar hasta 1 000 claves.

Lista de buckets

Para devolver una lista de todos los buckets pertenecientes al remitente autenticado de la solicitud:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Issue call
ListBucketsResponse response = client.ListBuckets();

// View response data
Console.WriteLine("Buckets owner - {0}", response.Owner.DisplayName);
foreach (S3Bucket bucket in response.Buckets)
{
    Console.WriteLine("Bucket {0}, Created on {1}", bucket.BucketName,
        bucket.CreationDate);
}
```

Lista de objetos

Puede devolver algunos de los objetos almacenados en su bucket de S3 o todos ellos (hasta 1 000). Para hacerlo, debe disponer de acceso de lectura en el bucket.

```
// Create a GetObject request
GetObjectRequest request = new GetObjectRequest
{
    BucketName = "SampleBucket",
    Key = "Item1"
};

// Issue request and remember to dispose of the response
using (GetObjectResponse response = client.GetObject(request))
{
    using (StreamReader reader = new StreamReader(response.ResponseStream))
    {
        string contents = reader.ReadToEnd();
        Console.WriteLine("Object - " + response.Key);
        Console.WriteLine(" Version Id - " + response.VersionId);
        Console.WriteLine(" Contents - " + contents);
    }
}
```

Obtención de la región de un bucket

Para obtener la región en la que reside un bucket:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Construct request
GetBucketLocationRequest request = new GetBucketLocationRequest
{
    BucketName = "SampleBucket"
};

// Issue call
GetBucketLocationResponse response = client.GetBucketLocation(request);

// View response data
Console.WriteLine("Bucket location - {0}", response.Location);
```

Obtención de la política de un bucket

Para obtener la política de un bucket:

```
// Create a client
AmazonS3Client client = new AmazonS3Client();

// Construct request
GetBucketPolicyRequest getRequest = new GetBucketPolicyRequest
{
    BucketName = "SampleBucket"
};

string policy = client.GetBucketPolicy(getRequest).Policy;

Console.WriteLine(policy);
Debug.Assert(policy.Contains("BasicPerms"));
```

Amazon DynamoDB

¿Qué es Amazon DynamoDB?

[Amazon DynamoDB](#) es un servicio de bases de datos no relacionales rápido y con un alto grado de escalabilidad. DynamoDB elimina las limitaciones tradicionales de escalabilidad del almacenamiento de datos y, al mismo tiempo, mantiene una baja latencia y un desempeño previsible.

Conceptos clave

El modelo de datos de DynamoDB incluye conceptos como las tablas, los elementos y los atributos.

Tablas

En Amazon DynamoDB, una base de datos es una colección de tablas. Una tabla es una colección de elementos y cada elemento es una colección de atributos.

En una base de datos relacional, una tabla tiene un esquema predefinido como, por ejemplo, el nombre de la tabla, la clave principal, la lista de los nombres de columnas y los tipos de datos de estas. Todos los registros almacenados en la tabla deben tener el mismo conjunto de columnas. En cambio, DynamoDB solo requiere que una tabla tenga una clave principal, pero no es necesario que se definan todos los nombres de atributos y los tipos de datos de antemano.

Para obtener más información acerca del procedimiento para trabajar con tablas, consulte [Uso de tablas en DynamoDB](#).

Elementos y atributos

Cada elemento de una tabla de DynamoDB puede tener cualquier número de atributos, aunque existe un límite de 400 KB para el tamaño del elemento. El tamaño de un elemento es la suma de las longitudes de los nombres y los valores de sus atributos (longitudes binarias y UTF-8).

Cada atributo de un elemento es un par nombre-valor. Un atributo puede tener un valor único o un conjunto de varios valores (atributo multivalor). Por ejemplo, un elemento de libro puede tener atributos de título y autores. Cada libro tiene un título, pero puede tener muchos autores. El atributo multivalor es un conjunto. No se admiten los valores duplicados.

Por ejemplo, imagine que desea almacenar un catálogo de productos en DynamoDB. Puede crear una tabla, `ProductCatalog`, con el atributo `ID` como clave principal. La clave principal identifica de forma única a cada elemento, de forma que en la tabla no puede haber dos productos con el mismo `ID`.

Para obtener más información acerca del trabajo con elementos, consulte [Uso de elementos en DynamoDB](#).

Tipos de datos

Amazon DynamoDB admite los siguientes tipos de datos:

- Tipos escalares: número, cadena, binario, booleano y nulo.
- Tipos multivalor: conjunto de cadenas, conjunto de números y conjunto binario.
- Tipos de documentos: lista y mapa.

Para obtener más información acerca de los tipos de datos escalares, los tipos de datos multivalor y los tipos de datos de documentos, consulte [Tipos de datos de DynamoDB](#).

Clave principal

Al crear una tabla, además de asignarle un nombre, debe especificar su clave principal. La clave principal identifica de forma única a cada elemento de la tabla, de manera que no puede haber dos elementos con la misma clave. DynamoDB admite los dos tipos siguientes de claves principales:

- Clave hash: La clave principal se compone de un atributo hash. DynamoDB genera un índice hash desordenado basado en este atributo de la clave principal. Cada elemento de la tabla se identifica de forma única por medio de su valor de clave hash.
- Clave hash y rango: La clave principal se compone de dos atributos. El primer atributo es el atributo hash y el segundo es el atributo de rango. DynamoDB crea un índice hash sin ordenar basado en el atributo hash de la clave principal y un índice de rango ordenado basado en el atributo de rango de la clave principal. Cada elemento de la tabla se identifica de forma única por medio de la combinación de sus valores de clave hash y de rango. Dos elementos pueden tener el mismo valor de clave hash, pero deben tener diferentes valores de clave de rango.

Índices secundarios

Cuando cree una tabla con una clave hash y de rango, puede definir uno o varios índices secundarios en dicha tabla. Un índice secundario le permite consultar los datos de la tabla usando una clave alternativa, además de realizar consultas basadas en la clave principal.

DynamoDB admite dos tipos de índices secundarios: los índices secundarios locales y los índices secundarios globales.

- Índice secundario local: Índice que tiene la misma clave hash que la tabla, pero una clave de rango distinta.
- Índice secundario global: Índice con una clave hash y de rango que puede diferir de las claves de la tabla.

Puede definir un máximo de 5 índices secundarios globales y 5 índices secundarios locales por cada tabla. Para obtener más información, consulte [Uso de índices secundarios para mejorar el acceso a los datos en DynamoDB](#) en la Guía para desarrolladores de DynamoDB.

Consulta y examen

Además de utilizar las claves principales para obtener acceso a los elementos, Amazon DynamoDB proporciona dos API para realizar búsquedas en los datos: Consulta y examen. Es recomendable que lea [Guidelines for Query and Scan](#) en la Guía para desarrolladores de DynamoDB para familiarizarse con algunas prácticas recomendadas.

Consulta

Una operación de consulta (Query) encuentra los elementos de una tabla o un índice secundario usando únicamente los valores de los atributos de la clave principal. Debe proporcionar un nombre de atributo de

clave hash y un valor para la búsqueda. Si lo desea, puede proporcionar un nombre y un valor de atributo de clave de rango y utilizar un operador de comparación para limitar los resultados de búsqueda.

Para ver las consultas de muestra, lea:

- [Uso del modelo de documento \(p. 45\)](#)
- [Uso del modelo de persistencia de objetos \(p. 47\)](#)
- [Uso de las API de nivel de servicio de DynamoDB \(p. 50\)](#)

Para obtener más información acerca de las consultas, vea [Consulta](#) en la Guía para desarrolladores de DynamoDB.

Examen

Una operación de examen (Scan) lee todos los elementos de una tabla o de un índice secundario. De manera predeterminada, una operación de examen devuelve todos los atributos de datos de todos los elementos de la tabla o el índice. Puede utilizar el `ProjectionExpression` para que Scan solo devuelva algunos de los atributos, en lugar de todos ellos.

Para ver los exámenes de muestra, lea:

- [Uso del modelo de documento \(p. 45\)](#)
- [Uso del modelo de persistencia de objetos \(p. 47\)](#)
- [Uso de las API de nivel de servicio de DynamoDB \(p. 50\)](#)

Para obtener más información acerca de las operaciones de análisis, consulte [Análisis](#) en la Guía para desarrolladores de DynamoDB.

Configuración del proyecto

Requisitos previos

Para utilizar DynamoDB en su aplicación, tendrá que añadir el SDK al proyecto. Para ello, siga las instrucciones que se describen en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Creación de una tabla de DynamoDB

Para crear una tabla, vaya a la [consola de DynamoDB](#) y siga estos pasos:

1. Haga clic en Create Table (Crear tabla).
2. Escriba el nombre de la tabla.
3. Seleccione Hash (Hash) como tipo de clave principal.
4. Seleccione un tipo y escriba un valor para el nombre del atributo hash. Haga clic en Continue (Continuar).
5. En la página Add Indexes (Añadir índices), si tiene previsto usar índices secundarios globales, defina Index Type (Tipo de índice) como "Global Secondary Index" y en Index Hash Key (Clave hash de índice), escriba un valor para el índice secundario. Esto le permitirá realizar consultas y exámenes usando el índice principal y el índice secundario. Haga clic en Add Index To Table (Añadir índice a tabla) y, a continuación, haga clic en Continue (Continuar). Para omitir la utilización de los índices secundarios globales, haga clic en Continue (Continuar).

6. Establezca la capacidad de lectura y escritura en los niveles deseados. Para obtener más información acerca de la configuración de la capacidad, consulte [Provisioned Throughput in Amazon DynamoDB](#). Haga clic en Continue (Continuar).
7. En la siguiente pantalla, escriba un correo electrónico de notificación para crear alarmas de rendimiento si lo desea. Haga clic en Continue (Continuar).
8. En la página de resumen, haga clic en Create (Crear). DynamoDB creará la base de datos.

Definición de permisos para DynamoDB

Para utilizar DynamoDB en una aplicación, debe definir los permisos correctos. La siguiente política de IAM permite al usuario eliminar, obtener, colocar, consultar, examinar y actualizar elementos en una tabla de DynamoDB, que se identifica mediante un [ARN](#):

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

Puede modificar las políticas en la [consola de IAM](#). Debe añadir o eliminar acciones permitidas en función de las necesidades de su aplicación.

Para obtener más información acerca de las políticas de IAM, consulte [cómo usar IAM](#).

Para obtener más información acerca de las políticas específicas de DynamoDB, consulte [Cómo usar IAM para controlar el acceso a los recursos de DynamoDB](#) en la Guía para desarrolladores de DynamoDB.

Integración de DynamoDB en una aplicación

AWS Mobile SDK for .NET and Xamarin proporciona una biblioteca de alto nivel para trabajar con DynamoDB. También se pueden realizar solicitudes directamente en la API de DynamoDB de bajo nivel, pero para la mayoría de los casos de uso se recomienda trabajar con la biblioteca de alto nivel. LaAmazonDynamoDBClient es una parte especialmente útil de la biblioteca de alto nivel. Con esta clase, puede realizar varias operaciones de creación, lectura, actualización y eliminación (CRUD, Create, Read, Update and Delete) y ejecutar consultas.

AWS Mobile SDK for .NET and Xamarin le permite realizar llamadas a través de las API de AWS SDK for .NET trabajar con DynamoDB. Todas las API están disponibles en laAWSSDK.dll. Para obtener información acerca de la descarga de AWS SDK para .NET, consulte [AWS SDK para .NET](#).

Existen tres formas de interactuar con DynamoDB en una aplicación de Xamarin:

- Modelo de documento: Esta API proporciona clases contenedoras relacionadas con la API de bajo nivel de DyanmoDB para simplificar las tareas de programación. Table y Document son

las clases contenedoras principales. Puede utilizar el modelo de documentos para operaciones de datos como, por ejemplo, crear, recuperar, actualizar y eliminar elementos. La API está disponible en `Amazon.DynamoDB.DocumentModel` espacio de nombres.

- **Modelo de persistencia de objetos:** La API de persistencia de objetos permite asignar las clases del lado del cliente a las tablas de DynamoDB. A continuación, cada instancia de objeto se mapea a un elemento en las tablas correspondientes. La clase `DynamoDBContext` de esta API proporciona métodos para guardar los objetos del lado del cliente en una tabla, recuperar elementos como objetos y realizar operaciones de consulta y examen. Puede utilizar el modelo de persistencia de objetos para operaciones de datos como, por ejemplo, crear, recuperar, actualizar y eliminar elementos. En primer lugar, debe crear sus tablas con la API de cliente de servicio y, a continuación, utilizar el modelo de persistencia de objetos para asignar las clases a las tablas. La API está disponible en `Amazon.DynamoDB.DataModel` espacio de nombres.
- **API del cliente de servicio:** Esta es la API de nivel de protocolo que más se ajusta a la API de DynamoDB. Puede utilizar esta API de bajo nivel para todas las operaciones de tablas y elementos como, por ejemplo, crear, actualizar y eliminar tablas y elementos. También puede consultar y examinar sus tablas. Esta API está disponible en el espacio de nombres `Amazon.DynamoDB`.

Estos tres modelos se exploran de manera detallada en los siguientes temas:

Uso del modelo de documento

El modelo de documento proporciona clases de encapsulamiento para la API .NET de bajo nivel. `Table` y `Document` son las clases contenedoras principales. Puede utilizar el modelo de documento para crear, recuperar, actualizar y eliminar elementos. Para crear, actualizar y eliminar las tablas, debe utilizar la API de bajo nivel. Para obtener instrucciones sobre cómo utilizar la API de bajo nivel, consulte [Uso de las API de nivel de servicio de DynamoDB \(p. 50\)](#). La API de bajo nivel está disponible en `Amazon.DynamoDB.DocumentModel` espacio de nombres.

Para obtener más información sobre el modelo de documento, consulte [Modelo de documento de .NET](#).

Creación de un cliente de DynamoDB

Para crear un cliente de DynamoDB:

```
var client = new AmazonDynamoDBClient(credentials, region);
DynamoDBContext context = new DynamoDBContext(client);
```

Operaciones de CRUD

Almacenamiento de un elemento

Crear un elemento:

```
Table table = Table.LoadTable(client, "Books");
id = Guid.NewGuid().ToString();
var books = new Document();
books["Id"] = id;
books["Author"] = "Mark Twain";
books["Title"] = "Adventures of Huckleberry Finn";
books["ISBN"] = "112-111111";
books["Price"] = "10";
```

Guardar un elemento en una tabla de DynamoDB:

```
var book = await table.PutItemAsync(books);
```

Recuperación de un elemento

Para recuperar un elemento:

```
public async Task GetItemAsync(AWSCredentials credentials, RegionEndpoint region)
{
    var client = new AmazonDynamoDBClient(credentials, region);
    Table books = Table.LoadTable(client, "Books");
    var book = await books.GetItemAsync(id);
}
```

Actualización de un elemento

Para actualizar un elemento:

```
public async Task UpdateItemAttributesAsync(AWSCredentials credentials, RegionEndpoint region)
{
    var book = new Document();
    book["Id"] = id;
    book["PageCount"] = "200";
    var client = new AmazonDynamoDBClient(credentials, region);
    Table books = Table.LoadTable(client, "Books");
    Document updatedBook = await books.UpdateItemAsync(book);
}
```

Para actualizar un elemento condicionalmente:

```
public async Task UpdateItemConditionallyAsync(AWSCredentials credentials, RegionEndpoint region) {
    var book = new Document();
    book["Id"] = id;
    book["Price"] = "30";

    // For conditional price update, creating a condition expression.
    Expression expr = new Expression();
    expr.ExpressionStatement = "Price = :val";
    expr.ExpressionAttributeValues[":val"] = 10.00;

    var client = new AmazonDynamoDBClient(credentials, region);
    Table books = Table.LoadTable(client, "Books");

    Document updatedBook = await books.UpdateItemAsync(book);
}
```

Eliminación de un elemento

Para eliminar un elemento:

```
public async Task DeleteItemAsync(AWSCredentials credentials, RegionEndpoint region)
{
    var client = new AmazonDynamoDBClient(credentials, region);
    Table books = Table.LoadTable(client, "Books");
}
```

```
await books.DeleteItemAsync(id);  
}
```

Consulta y examen

Para consultar y recuperar todos los libros cuyo autor sea "Mark Twain":

```
public async Task QueryAsync(AWSCredentials credentials, RegionEndpoint region) {  
    var client = new AmazonDynamoDBClient(credentials, region);  
    Table books = Table.LoadTable(client, "Books");  
    var search = books.Query(new QueryOperationConfig() {  
        IndexName = "Author-Title-index",  
        Filter = new QueryFilter("Author", QueryOperator.Equal, "Mark Twain")  
    });  
    Console.WriteLine("ScanAsync: printing query response");  
    var documents = await search.GetRemainingAsync();  
    documents.ForEach((d) => {  
        PrintDocument(d);  
    });  
}
```

El código de ejemplo de examen que se muestra a continuación devuelve todos los libros de la tabla:

```
public async Task ScanAsync(AWSCredentials credentials, RegionEndpoint region) {  
    var client = new AmazonDynamoDBClient(credentials, region);  
    Table books = Table.LoadTable(client, "Books");  
    var search = books.Scan(new ScanOperationConfig() {  
        ConsistentRead = true  
    });  
    Console.WriteLine("ScanAsync: printing scan response");  
    var documents = await search.GetRemainingAsync();  
    documents.ForEach((d) => {  
        PrintDocument(d);  
    });  
}
```

Uso del modelo de persistencia de objetos

El AWS Mobile SDK for .NET and Xamarin proporciona un modelo de persistencia de objetos que le permite mapear las clases en el cliente a una tabla de DynamoDB. A continuación, cada instancia de objeto se mapea a un elemento de la tabla correspondiente. Para guardar los objetos de cliente en una tabla, el modelo de persistencia de objetos proporciona la clase `DynamoDBContext`, un punto de entrada a DynamoDB. Esta categoría le ofrece una conexión a DynamoDB y le permite obtener acceso a tablas, realizar diversas operaciones CRUD y ejecutar consultas.

El modelo de persistencia de objetos no proporciona una API para crear, actualizar o eliminar tablas. Solo ofrece operaciones de datos. Para crear, actualizar y eliminar las tablas, debe utilizar la API de bajo nivel. Para obtener instrucciones sobre cómo utilizar la API de bajo nivel, consulte [Uso de las API de nivel de servicio de DynamoDB \(p. 50\)](#).

Información general

El modelo de persistencia de objetos proporciona un conjunto de atributos para mapear las clases de cliente a tablas, y propiedades/campos a atributos de tabla. El modelo de persistencia de objetos admite el mapeo explícito y predeterminado entre las propiedades de clase y los atributos de tabla.

- Mapeo explícito: Para mapear una propiedad a una clave principal, debe usar `DynamoDBHashKey` `DynamoDBRangeKey` Atributos del modelo de persistencia de objetos. Además, en el caso de los atributos de clave no principal, si un nombre de propiedad de la clase y el atributo de tabla correspondiente al que desea mapearlo no son iguales, debe definir el mapeo agregando explícitamente el atributo `DynamoDBProperty`.
- Mapeo predeterminado: de forma predeterminada, el modelo de persistencia de objetos mapea las propiedades de clase a los atributos con el mismo nombre de la tabla.

No tiene que mapear cada propiedad de clase. Puede identificar estas propiedades agregando el atributo `DynamoDBIgnore`. Al guardar y recuperar una instancia de un objeto se podría omitir cualquier propiedad marcada con este atributo.

Tipos de datos admitidos

El modelo de persistencia de objetos admite un conjunto de tipos de datos, colecciones y tipos de datos arbitrarios de .NET primitivos. El modelo admite los siguientes tipos de datos primitivos.

- `bool`
- `byte`
- `char`
- `DateTime`
- `decimal`, `double`, `float`
- `Int16`, `Int32`, `Int64`
- `SByte`
- `string`
- `UInt16`, `UInt32`, `UInt64`

El modelo de persistencia de objetos también admite los tipos de colecciones .NET con las limitaciones siguientes:

- El tipo de colección debe implementar la interfaz `ICollection`.
- El tipo de colección se debe componer de los tipos de primitivas admitidos. Por ejemplo, `ICollection<string>`, `ICollection<bool>`.
- El tipo de colección debe proporcionar un constructor sin parámetros.

Para obtener más información sobre el modelo de persistencia de objetos, consulte [Modelo de persistencia de objetos de .NET](#).

Creación de un cliente de DynamoDB

Para crear un cliente de DynamoDB:

```
var client = new AmazonDynamoDBClient(credentials, region);
DynamoDBContext context = new DynamoDBContext(client);
```

Operaciones de CRUD

Guardar un objeto

Crear un objeto:

```
[DynamoDBTable("Books")]
public class Book {
    [DynamoDBHashKey] // Hash key.
    public string Id {
        get;
        set;
    }

    [DynamoDBGlobalSecondaryIndexHashKey]
    public string Author {
        get;
        set;
    }

    [DynamoDBGlobalSecondaryIndexRangeKey]
    public string Title {
        get;
        set;
    }
    public string ISBN {
        get;
        set;
    }
    public int Price {
        get;
        set;
    }
    public string PageCount {
        get;
        set;
    }
}

Book myBook = new Book
{
    Id = id,
    Author = "Charles Dickens",
    Title = "Oliver Twist",
    ISBN = "111-1111111001",
    Price = 10,
    PageCount = 300
};
```

Guardar un objeto en una tabla de DynamoDB:

```
context.Save(myBook);
```

Recuperar un objeto

Para recuperar un objeto:

```
Book retrievedBook = context.Load<Book>(1);
```

Actualizar un objeto

Para actualizar un objeto:

```
Book retrievedBook = context.Load<Book>(1);
retrievedBook.ISBN = "111-1111111001";
```



```
context.Save(retrievedBook);
```

Eliminar un objeto

Para eliminar un objeto:

```
Book retrievedBook = context.Load<Book>(1);  
context.Delete(retrievedBook);
```

Consulta y examen

Para consultar y recuperar todos los libros cuyo autor sea "Charles Dickens":

```
public async Task QueryAsync(AWSCredentials credentials, RegionEndpoint region) {  
    var client = new AmazonDynamoDBClient(credentials, region);  
    DynamoDBContext context = new DynamoDBContext(client);  
  
    var search = context.FromQueryAsync < Book > (new  
        Amazon.DynamoDBv2.DocumentModel.QueryOperationConfig() {  
            IndexName = "Author-Title-index",  
            Filter = new Amazon.DynamoDBv2.DocumentModel.QueryFilter("Author",  
                Amazon.DynamoDBv2.DocumentModel.QueryOperator.Equal, "Charles Dickens")  
        });  
  
    Console.WriteLine("items retrieved");  
  
    var searchResponse = await search.GetRemainingAsync();  
    searchResponse.ForEach((s) => {  
        Console.WriteLine(s.ToString());  
    });  
}
```

El código de ejemplo de examen que se muestra a continuación devuelve todos los libros de la tabla:

```
public async Task ScanAsync(AWSCredentials credentials, RegionEndpoint region) {  
    var client = new AmazonDynamoDBClient(credentials, region);  
    DynamoDBContext context = new DynamoDBContext(client);  
  
    var search = context.FromScanAsync < Book > (new  
        Amazon.DynamoDBv2.DocumentModel.ScanOperationConfig() {  
            ConsistentRead = true  
        });  
  
    Console.WriteLine("items retrieved");  
  
    var searchResponse = await search.GetRemainingAsync();  
    searchResponse.ForEach((s) => {  
        Console.WriteLine(s.ToString());  
    });  
}
```

Uso de las API de nivel de servicio de DynamoDB

Las API de nivel de servicio de Dynamo le permiten crear, actualizar y eliminar tablas. También puede realizar operaciones típicas de creación, lectura, actualización y eliminación (CRUD, Create, Read, Update and Delete) en los elementos de una tabla usando esta API.

Creación de un cliente de DynamoDB

Para crear un cliente de DynamoDB:

```
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials,region);
```

Operaciones de CRUD

Almacenamiento de un elemento

Para guardar un elemento en una tabla de DynamoDB:

```
// Create a client
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials,region);

// Define item attributes
Dictionary<string, AttributeValue> attributes = new Dictionary<string, AttributeValue>();

// Author is hash-key
attributes["Author"] = new AttributeValue { S = "Mark Twain" };
attributes["Title"] = new AttributeValue { S = "The Adventures of Tom Sawyer" };
attributes["PageCount"] = new AttributeValue { N = "275" };
attributes["Price"] = new AttributeValue{N = "10.00"};
attributes["Id"] = new AttributeValue{N="10"};
attributes["ISBN"] = new AttributeValue{S="111-1111111"};

// Create PutItem request
PutItemRequest request = new PutItemRequest
{
    TableName = "Books",
    Item = attributes
};

// Issue PutItem request
var response = await client.PutItemAsync(request);
```

Recuperación de un elemento

Para recuperar un elemento:

```
// Create a client
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials,region);

Dictionary<string, AttributeValue> key = new Dictionary<string, AttributeValue>
{
    { "Id", new AttributeValue { N = "10" } }
};

// Create GetItem request
GetItemRequest request = new GetItemRequest
{
    TableName = "Books",
    Key = key,
};

// Issue request
var result = await client.GetItemAsync(request);

// View response
```

```
Console.WriteLine("Item:");
Dictionary<string, AttributeValue> item = result.Item;
foreach (var keyValuePair in item)
{
    Console.WriteLine("Author := {0}", item["Author"]);
    Console.WriteLine("Title := {0}", item["Title"]);
    Console.WriteLine("Price:= {0}", item["Price"]);
    Console.WriteLine("PageCount := {0}", item["PageCount"]);
}
```

Actualización de un elemento

Para actualizar un elemento:

```
// Create a client
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials,region);

Dictionary<string, AttributeValue> key = new Dictionary<string, AttributeValue>
{
    { "Id", new AttributeValue { N = "10" } }
};

// Define attribute updates
Dictionary<string, AttributeValueUpdate> updates = new Dictionary<string,
    AttributeValueUpdate>();
// Add a new string to the item's Genres SS attribute
updates["Genres"] = new AttributeValueUpdate()
{
    Action = AttributeAction.ADD,
    Value = new AttributeValue { SS = new List<string> { "Bildungsroman" } }
};

// Create UpdateItem request
UpdateItemRequest request = new UpdateItemRequest
{
    TableName = "Books",
    Key = key,
    AttributeUpdates = updates
};

// Issue request
var response = await client.UpdateItemAsync(request);
```

Eliminación de un elemento

Para eliminar un elemento:

```
// Create a client
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials,region);

Dictionary<string, AttributeValue> key = new Dictionary<string, AttributeValue>
{
    { "Id", new AttributeValue { N = "10" } }
};

// Create DeleteItem request
DeleteItemRequest request = new DeleteItemRequest
{
    TableName = "Books",
    Key = key
};
```

```
// Issue request  
var response = await client.DeleteItemAsync(request);
```

Consulta y examen

Para consultar y recuperar todos los libros cuyo autor sea "Mark Twain":

```
public void Query(AWSCredentials credentials, RegionEndpoint region) {  
    using(var client = new AmazonDynamoDBClient(credentials, region)) {  
        var queryResponse = await client.QueryAsync(new QueryRequest() {  
            TableName = "Books",  
            IndexName = "Author-Title-index",  
            KeyConditionExpression = "Author = :v_Id",  
            ExpressionAttributeValues = new Dictionary < string, AttributeValue > {  
                {  
                    ":v_Id", new AttributeValue {  
                        S = "Mark Twain"  
                    }  
                }  
            }  
        });  
        queryResponse.Items.ForEach((i) => {  
            Console.WriteLine(i["Title"].S);  
        });  
    }  
}
```

El código de ejemplo de examen que se muestra a continuación devuelve todos los libros de la tabla:

```
public void Scan(AWSCredentials credentials, RegionEndpoint region) {  
    using(var client = new AmazonDynamoDBClient(credentials, region)) {  
        var queryResponse = client.Scan(new ScanRequest() {  
            TableName = "Books"  
        });  
        queryResponse.Items.ForEach((i) => {  
            Console.WriteLine(i["Title"].S);  
        });  
    }  
}
```

Amazon Simple Notification Service (SNS)

Con SNS y el SDK de AWS Mobile para .NET y Xamarin podrá escribir aplicaciones que pueden recibir notificaciones de inserción móviles. Para obtener más información, consulte [Amazon Simple Notification Service](#).

Conceptos clave

Amazon SNS permite que las aplicaciones y los usuarios finales en dispositivos diferentes reciban notificaciones mediante notificaciones de Mobile Push (Apple, Google y aparatos Kindle), HTTP/HTTPS, email/email-JSON, SMS o colas de Amazon Simple Queue Service (SQS), o funciones de AWS Lambda. SNS le permite enviar mensajes individuales o distribuir mensajes a un gran número de destinatarios suscritos a un único tema.

Temas

Un tema es un "punto de acceso" que permite a los destinatarios suscribirse de forma dinámica para obtener copias de la una notificación. Un tema puede admitir envíos a varios tipos de punto de enlace: por ejemplo, se pueden agrupar destinatarios de iOS, Android y SMS.

Suscripciones

Para recibir los mensajes publicados en un tema, tiene que suscribirse a un punto de enlace en ese tema. Un punto de enlace es una aplicación móvil, un servidor web, una dirección web o una cola de Amazon SQS que puede recibir mensajes de notificación de Amazon SNS. Una vez se suscriba a un punto de enlace en un tema y la suscripción se haya confirmado, el punto de enlace recibirá todos los mensajes publicados en el tema.

Publicación

Cuando publica en un tema, SNS envía copias correctamente formateadas de su mensaje a cada persona suscrita al tema. Para Mobile Push Notifications, puede publicar directamente en el punto de enlace o suscribir el punto de enlace en un tema.

Configuración del proyecto

Requisitos previos

Para utilizar SNS en su aplicación, tendrá que añadir el SDK a su proyecto. Para ello, siga las instrucciones que se describen en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).

Establecer permisos de SNS

Para obtener más información sobre cómo establecer permisos de SNS, consulte [Managing Access to Your Amazon SNS Topics](#).

AñadirNuGetPaquete de SNS para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para agregar Amazon Simple Notification ServiceNuGetpaquete para tu proyecto.

Integrar SNS en la aplicación

Hay muchas formas de interactuar con SNS en la aplicación Xamarin:

Enviar notificaciones de inserción (Xamarin para Android)

En este documento se explica cómo enviar notificaciones de inserción a una aplicación de Xamarin Android con Amazon Simple Notification Service (SNS) y AWS Mobile SDK for .NET and Xamarin.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Establecer permisos de SNS

Siga el paso 2 de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) para adjuntar la política que se menciona a continuación a los roles de su aplicación. Esto otorgará a su aplicación los permisos correctos para obtener acceso a SNS:

1. Vaya a la [consola de IAM](#) y seleccione el rol de IAM que desea configurar.
2. ClicAdjuntar política, seleccione AmazonSNSFullAccesspolítica y haga clic enAdjuntar política.

Warning

UsarUse AmazonSNSFullAccessNo es recomendable en un entorno de producción. Aquí lo utilizamos para que pueda comenzar a trabajar rápidamente. Para obtener más información acerca de la especificación de los permisos de un rol de IAM, consulte la [información general sobre los permisos de los roles de IAM](#).

Habilitar notificaciones de inserción en Google Cloud

En primer lugar, añada un nuevo proyecto de API de Google:

1. Vaya a la [Consola para desarrolladores de Google](#).
2. Haga clic en Create Project (Crear proyecto).
3. En el cuadro New Project (Nuevo proyecto), escriba un nombre de proyecto, tome nota del ID de proyecto (lo necesitará más adelante) y haga clic en Create (Crear).

A continuación, habilite el servicio Google Cloud Messaging (GCM) para el proyecto:

1. En la [consola para desarrolladores de Google](#), el nuevo proyecto ya debería estar seleccionado. En caso contrario, selecciónela en el menú desplegable situado en la parte superior de la página.
2. Seleccione APIs & auth (API y autenticación) en la barra lateral situada en la parte izquierda de la página.
3. En el cuadro de búsqueda, escriba "Google Cloud Messaging for Android" y haga clic en el enlace Google Cloud Messaging for Android (Google Cloud Messaging para Android).
4. Haga clic en Enable API (Habilitar API).

Por último, obtenga una clave de API:

1. En la consola de Google para desarrolladores, seleccione APIs & auth (API y autenticación) > Credentials (Credenciales).
2. En Public API access (Acceso a API pública), haga clic en Create new key (Crear nueva clave).
3. En el cuadro de diálogo Create a new key (Crear nueva clave), haga clic en Server key (Clave de servidor).
4. En el cuadro de diálogo resultante, haga clic en Create (Crear) y copie la clave de API que se muestra. Utilizará esta clave de API para realizar la autenticación más adelante.

Uso del ID de proyecto para crear un ARN de plataforma en la consola de SNS

1. Vaya a la [consola de SNS](#).
2. Haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.
3. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
4. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
5. Seleccione Google Cloud Messaging (GCM) en Push notification platform (Plataforma de notificación de inserción).
6. Pegue la clave de API en el cuadro de texto API key (Clave de API).
7. Haga clic en Create platform application (Crear aplicación de plataforma).
8. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación.

AñadirNuGetPaquete de SNS para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin \(p. 4\)](#) para agregar Amazon Simple Notification ServiceNuGetpaquete para su proyecto.

Creación de un cliente de SNS

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

Registro de una aplicación para recibir notificaciones remotas

Si desea registrarse para recibir notificaciones remotas en Android, tendrá que crear un `BroadcastReceiver` que puede recibir mensajes de Google Cloud. Cambie el nombre del paquete cuando se le pida a continuación:

```
[BroadcastReceiver(Permission = "com.google.android.c2dm.permission.SEND")]
[IntentFilter(new string[] {
    "com.google.android.c2dm.intent.RECEIVE"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
[IntentFilter(new string[] {
    "com.google.android.c2dm.intent.REGISTRATION"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
[IntentFilter(new string[] {
    "com.google.android.gcm.intent.RETRY"
}, Categories = new string[] {
    "com.amazonaws.sns" /* change to match your package */
})]
public class GCMBroadcastReceiver: BroadcastReceiver {
    const string TAG = "PushHandlerBroadcastReceiver";
    public override void OnReceive(Context context, Intent intent) {
        GCMIntentService.RunIntentInService(context, intent);
        SetResult(Result.Ok, null, null);
    }
}

[BroadcastReceiver]
[IntentFilter(new[] {
    Android.Content.Intent.ActionBootCompleted
})]
public class GCMBootReceiver: BroadcastReceiver {
    public override void OnReceive(Context context, Intent intent) {
        GCMIntentService.RunIntentInService(context, intent);
        SetResult(Result.Ok, null, null);
    }
}
```

A continuación se puede ver el servicio que recibe la notificación de inserción del `BroadcastReceiver` y muestra la notificación en la barra de notificaciones del dispositivo:

```
[Service]
public class GCMIntentService: IntentService {
    static PowerManager.WakeLock sWakeLock;
    static object LOCK = new object();

    public static void RunIntentInService(Context context, Intent intent) {
        lock(LOCK) {
            if (sWakeLock == null) {
                // This is called from BroadcastReceiver, there is no init.
                var pm = PowerManager.FromContext(context);
                sWakeLock = pm.NewWakeLock(
                    WakeLockFlags.Partial, "My WakeLock Tag");
            }
        }

        sWakeLock.Acquire();
    }
}
```



```

intent.SetClass(context, typeof(GCMIntentService));
context.StartService(intent);
}

protected override void OnHandleIntent(Intent intent) {
    try {
        Context context = this.ApplicationContext;
        string action = intent.Action;

        if (action.Equals("com.google.android.c2dm.intent.REGISTRATION")) {
            HandleRegistration(intent);
        } else if (action.Equals("com.google.android.c2dm.intent.RECEIVE")) {
            HandleMessage(intent);
        }
    } finally {
        lock(LOCK) {
            //Sanity check for null as this is a public method
            if (sWakeLock != null) sWakeLock.Release();
        }
    }
}

private void HandleRegistration(Intent intent) {
    string registrationId = intent.GetStringExtra("registration_id");
    string error = intent.GetStringExtra("error");
    string unregistration = intent.GetStringExtra("unregistered");

    if (string.IsNullOrEmpty(error)) {
        var response = await SnsClient.CreatePlatformEndpointAsync(new
CreatePlatformEndpointRequest {
            Token = registrationId,
            PlatformApplicationArn = "YourPlatformArn" /* insert your platform application ARN
here */
        });
    }
}

private void HandleMessage(Intent intent) {
    string message = string.Empty;
    Bundle extras = intent.Extras;
    if (!string.IsNullOrEmpty(extras.GetString("message"))) {
        message = extras.GetString("message");
    } else {
        message = extras.GetString("default");
    }

    Log.Info("Messages", "message received = " + message);
    ShowNotification(this, "SNS Push", message);
    //show the message
}

public void ShowNotification(string contentTitle,
string contentText) {
    // Intent
    Notification.Builder builder = new Notification.Builder(this)
        .SetContentTitle(contentTitle)
        .SetContentText(contentText)
        .SetDefaults(NotificationDefaults.Sound | NotificationDefaults.Vibrate)
        .SetSmallIcon(Resource.Drawable.Icon)
        .SetSound(RingtoneManager.GetDefaultUri(RingtoneType.Notification));

    // Get the notification manager:
    NotificationManager notificationManager =
this.GetSystemService(Context.NotificationService) as NotificationManager;

```

```
notificationManager.Notify(1001, builder.Build());  
}  
}
```

Envío de un mensaje desde la consola de SNS a un punto de enlace

1. Vaya a la [consola de SNS](#) y haga clic en [Applications \(Aplicaciones\)](#).
2. Seleccione su aplicación de plataforma, seleccione un punto de enlace y haga clic en Publish to endpoint (Publicar en punto de enlace).
3. Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

Envío de notificaciones de inserción (Xamarin iOS)

En este documento se explica cómo enviar notificaciones de inserción a una aplicación de Xamarin iOS con Amazon Simple Notification Service (SNS) y AWS Mobile SDK for .NET and Xamarin.

Configuración del proyecto

Requisitos previos

Debe seguir las instrucciones en [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) antes de comenzar este tutorial.

Establecer permisos de SNS

Siga el paso 2 de [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#) para adjuntar la política que se menciona a continuación a los roles de su aplicación. Esto otorgará a su aplicación los permisos correctos para obtener acceso a SNS:

1. Vaya a la [consola de IAM](#) y seleccione el rol de IAM que desea configurar.
2. ClicAdjuntar política, seleccione AmazonSNSFullAccesspolítica y haga clic enAdjuntar política.

Warning

UsarUse AmazonSNSFullAccessno se recomienda en un entorno de producción. Aquí lo utilizamos para que pueda comenzar a trabajar rápidamente. Para obtener más información acerca de la especificación de los permisos de un rol de IAM, consulte la [información general sobre los permisos de los roles de IAM](#).

Obtención de una suscripción a Apple iOS Developer Program

Tendrá que ejecutar su aplicación en un dispositivo físico para recibir notificaciones de inserción. Para ejecutar la aplicación en un dispositivo, debe estar suscrito a [Apple iOS Developer Program](#). Una vez que se haya suscrito, podrá utilizar Xcode para generar una identidad de firma. Para obtener más información, consulte la documentación [App Distribution Quick Start](#) de Apple.

Creación de un certificado de iOS

En primer lugar, debe crear un certificado de iOS. A continuación, debe crear un perfil de aprovisionamiento configurado para las notificaciones de inserción. Para ello:

1. Vaya a [Apple Developer Member Center](#) y haga clic en Certificates, Identifiers & Profiles.
2. Haga clic en Identifiers (Identificadores) en iOS Apps (Aplicaciones iOS), haga clic en el botón más situado en la esquina superior derecha de la página web para añadir un nuevo ID de aplicación de iOS y escriba una descripción de ID de aplicación.
3. Desplácese hasta la sección Add ID Suffix (Agregar sufijo de ID), seleccione Explicit App ID (ID de aplicación explícito) y escriba el identificador de su paquete.
4. Desplácese hacia abajo hasta la sección App Services (Servicios de aplicación) y seleccione Push Notifications (Notificaciones de inserción).
5. Haga clic en Continue (Continuar).
6. Haga clic en Submit.
7. Haga clic en Done (Hecho).
8. Seleccione el ID de aplicación que acaba de crear y, a continuación, haga clic en Edit.
9. Desplácese hasta la sección Push Notifications (Notificaciones de inserción). Haga clic en Create Certificate en Development SSL Certificate.
10. Siga las instrucciones para crear una solicitud de firma del certificado (CSR), cargar la solicitud y descargar un certificado SSL que se utilizará para la comunicación con Apple Push Notification Service (APNS).
11. Vuelva a la página Certificates, Identifiers & Profiles. Haga clic en All en Provisioning Profiles.
12. Haga clic en el botón más situado en la esquina superior derecha para añadir un nuevo perfil de aprovisionamiento.
13. Seleccione iOS App Development y haga clic en Continue.
14. Seleccione su ID de aplicación y haga clic en Continue.
15. Seleccione su certificado de desarrollador y haga clic en Continue.
16. Seleccione su dispositivo y haga clic en Continue.
17. Escriba un nombre de perfil y haga clic en Generate.
18. Para instalar el perfil de aprovisionamiento, descargue el archivo de aprovisionamiento y haga doble clic en él.

Para obtener más información acerca del aprovisionamiento de un perfil configurado para las notificaciones de inserción, consulte la documentación [Configuring Push Notifications](#) de Apple.

Uso del certificado para crear un ARN de plataforma en la consola de SNS

1. Ejecute la KeyChain app de acceso, seleccione Mis certificados en la parte inferior izquierda de la pantalla y, a continuación, haga clic con el botón derecho del ratón en el certificado SSL que ha generado para conectarse a APNS y seleccione Exportar. Se le pedirá que especifique un nombre para el archivo y una contraseña para proteger el certificado. El certificado se guardará en un archivo P12.
2. Vaya a la [consola de SNS](#) y haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.
3. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
4. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
5. Seleccione Apple Development (Desarrollo Apple) en Push notification platform (Plataforma de notificación de inserción).
6. Haga clic en Choose File (Elegir archivo) y seleccione el archivo P12 que creó al exportar el certificado SSL.
7. Escriba la contraseña que especificó cuando exportó el certificado SSL y haga clic en Load Credentials From File (Cargar credenciales desde archivo).

8. Haga clic en Create platform application (Crear aplicación de plataforma).
9. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación. Lo necesitará en los próximos pasos.

AñadirNuGetPaquete para SNS para su proyecto

Siga el paso 4 de las instrucciones de [Configuración del SDK de AWS Mobile para .NET y Xamarin de \(p. 4\)](#) para agregar Amazon Simple Notification ServiceNuGetpaquete para su proyecto.

Creación de un cliente de SNS

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

Registro de una aplicación para recibir notificaciones remotas

Para registrar una solicitud, llameRegisterForRemoteNotificationsen el objeto UIApplication, como se muestra a continuación. Coloque el siguiente código enAppDelegate.cs, insertando el ARN de la aplicación de plataforma donde se le indique:

```
public override bool FinishedLaunching(UIApplication app, NSDictionary options) {  
    // do something  
    var pushSettings = UIUserNotificationSettings.GetSettingsForTypes (  
        UIUserNotificationType.Alert |  
        UIUserNotificationType.Badge |  
        UIUserNotificationType.Sound,  
        null  
    );  
    app.RegisterUserNotifications(pushSettings);  
    app.RegisterForRemoteNotifications();  
    // do something  
    return true;  
}  
  
public override void RegisteredForRemoteNotifications(UIApplication application, NSData token) {  
    var deviceToken = token.Description.Replace("<", "").Replace(">", "").Replace(" ", "");  
    if (!string.IsNullOrEmpty(deviceToken)) {  
        //register with SNS to create an endpoint ARN  
        var response = await SnsClient.CreatePlatformEndpointAsync(  
            new CreatePlatformEndpointRequest {  
                Token = deviceToken,  
                PlatformApplicationArn = "YourPlatformArn" /* insert your platform application ARN  
here */  
            });  
    }  
}
```

Envío de un mensaje desde la consola de SNS a un punto de enlace

1. Vaya a la [consola de SNS](#) y haga clic en Applications (Aplicaciones).
2. Seleccione su aplicación de plataforma, seleccione un punto de enlace y haga clic en Publish to endpoint (Publicar en punto de enlace).

3. Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

Enviar y recibir notificaciones SMS.

Puede utilizar Amazon Simple Notification Service (Amazon SNS) para enviar y recibir notificaciones SMS (servicio de mensajes cortos) en teléfonos móviles y smartphones que tengan SMS habilitado.

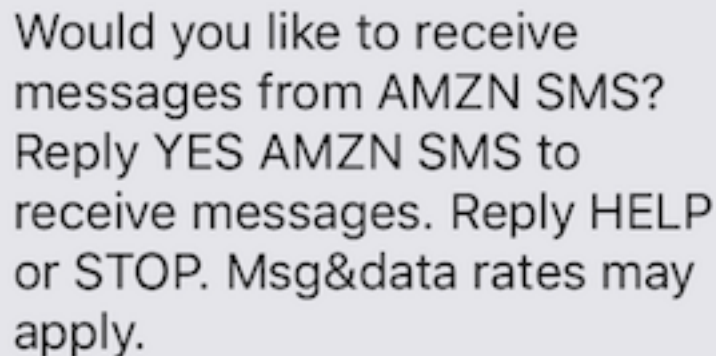
Note

Las notificaciones SMS actualmente se admiten para números de teléfono en los Estados Unidos. Los mensajes SMS solo se pueden enviar desde temas creados en la región EE.UU. Este (Norte de Virginia). Sin embargo, puede publicar mensajes en los temas que usted cree en la región EE.UU. Este (Norte de Virginia) desde cualquier otra región.

Creación de un tema

Para crear un tema:

1. En la consola de Amazon SNS, haga clic en Create new topic (Crear nuevo tema). Aparecerá el cuadro de diálogo Create new topic.
2. En el cuadro Topic name, escriba un nombre para el tema.
3. En el cuadro Display name, escriba el nombre que desee mostrar. El tema debe tener un nombre de visualización asignado, ya que los diez (10) primeros caracteres de ese nombre se utilizan como la parte inicial del prefijo del mensaje de texto. El nombre de visualización que introduzca aparecerá en el mensaje de confirmación que SNS envía al usuario (el nombre de visualización del ejemplo a continuación es "AMZN SMS").



Would you like to receive
messages from AMZN SMS?
Reply YES AMZN SMS to
receive messages. Reply HELP
or STOP. Msg&data rates may
apply.

1. Haga clic en Create topic (Crear tema). El nuevo tema aparecerá en la página Topics.
2. Seleccione el nuevo tema y después haga clic en el ARN del tema. Aparecerá la página Topic Details.
3. Copie el ARN del tema, ya que lo necesitará cuando se suscriba a un tema en el siguiente paso.

```
arn:aws:sns:us-west-2:111122223333:MyTopic
```

Suscribirse a un tema utilizando el protocolo SMS

Cree un cliente de SNS, pasando su objeto credentials y la región de su grupo de identidades:

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

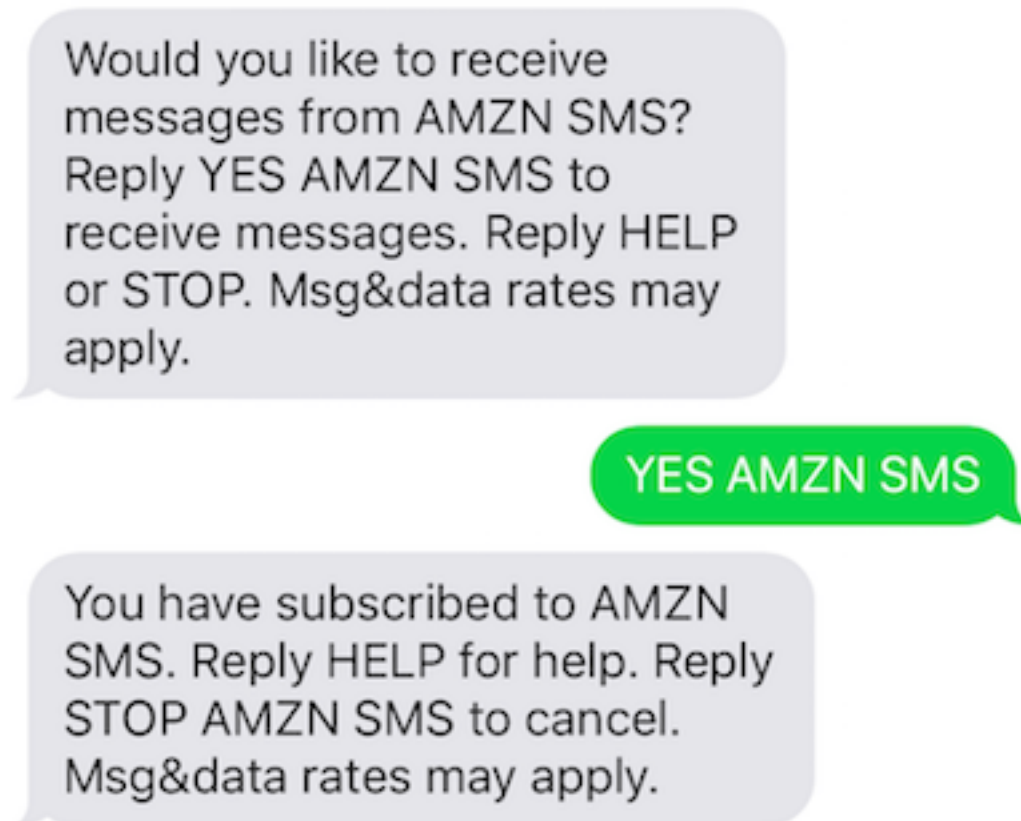
Para suscribirse a un tema, invoque `SubscribeAsync` y pase el ARN del tema al que desea suscribirse, el protocolo ("SMS") y el número de teléfono:

```
var response = await snsClient.SubscribeAsync(topicArn, "sms", "1234567890");
```

Recibirá un ARN suscribe en el objeto de respuesta `suscribe`. Su ARN suscribe tiene este aspecto:

```
arn:aws:sns:us-west-2:123456789012:MyTopic:6b0e71bd-7e97-4d97-80ce-4a0994e55286
```

Cuando un dispositivo se suscribe a un tema, SNS envía un mensaje de confirmación a ese dispositivo y los usuarios tienen que confirmar que desean recibir notificaciones, tal y como se muestra a continuación:



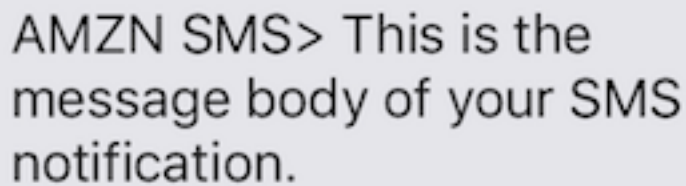
Una vez los usuarios se hayan suscrito al tema, recibirán mensajes SMS cuando los publique en él.

Publicar un mensaje

Para publicar un mensaje en un tema:

1. Inicie sesión en la consola de administración de AWS y abra la [consola de Amazon SNS](#).

2. En el panel de navegación de la izquierda, haga clic en Topics (Temas) y después seleccione el tema en el que desea publicar.
3. Haga clic en Publish to topic (Publicar en tema).
4. En el cuadro Subject, escriba un asunto.
5. En el cuadro Message, escriba un mensaje. Amazon SNS envía el texto que escriba en el cuadro Message a los suscriptores SMS a menos que también introduzca el texto en el cuadro Subject. Amazon SNS incluye un prefijo del nombre de visualización con todos los mensajes SMS que envíe. La suma del prefijo del nombre de visualización y la carga del mensaje no puede superar 140 caracteres ASCII o 70 caracteres Unicode. Amazon SNS trunca los mensajes que superen estos límites.
6. Haga clic en Publish message (Publicar mensaje). Amazon SNS muestra un cuadro de diálogo de confirmación. El mensaje SMS aparecerá en su dispositivo SMS habilitado como se muestra a continuación.



AMZN SMS> This is the message body of your SMS notification.

Envío de mensajes a puntos de enlace HTTP/HTTPS

Puede utilizar Amazon SNS para enviar mensajes de notificación a uno o varios puntos de enlace HTTP o HTTPS. El proceso es el siguiente:

1. Configure el punto de enlace para recibir mensajes de Amazon SNS.
2. Suscriba un punto de enlace HTTP/HTTPS a un tema.
3. Confirme la suscripción.
4. Publique una notificación en el tema. A continuación, Amazon SNS envía una solicitud HTTP POST que entrega el contenido de la notificación en el punto de enlace suscrito.

Configuración de un punto de enlace HTTP/HTTPS para recibir mensajes de Amazon SNS

Siga las instrucciones descritas en el paso 1 de [Sending Amazon SNS Messages to HTTP/HTTPS Endpoints](#) para configurar el punto de enlace.

Suscripción del punto de enlace HTTP/HTTPS a un tema de Amazon SNS

Cree un cliente de SNS, pasando su objeto credentials y la región de su grupo de identidades:

```
var snsClient = new AmazonSimpleNotificationServiceClient(credentials, region);
```

Para enviar mensajes a un punto de enlace HTTP o HTTPS a través de un tema, debe suscribir el punto de enlace al tema de Amazon SNS. El punto de enlace se especifica por medio de su URL:

```
var response = await snsClient.SubscribeAsync(  
    "topicArn",  
    "http", /* "http" or "https" */  
    "endpointUrl" /* endpoint url beginning with http or https */  
);
```

Confirmación de la suscripción

Tras la suscripción a un punto de enlace, Amazon SNS enviará un mensaje de confirmación de la suscripción al punto de enlace. El código del punto de enlace debe recuperar el valor de `SubscribeURL` del mensaje de confirmación de la suscripción y visitar la ubicación especificada por el valor de `SubscribeURL` o ponerla a disposición del usuario para que pueda visitar manualmente la dirección `SubscribeURL` (por ejemplo, si se usa un navegador web).

Amazon SNS no enviará mensajes al punto de enlace hasta que se confirme la suscripción. Cuando visite la dirección `SubscribeURL`, la respuesta contendrá un documento XML que incluye un elemento `SubscriptionArn` que especifica el ARN de la suscripción.

Envío de mensajes al punto de enlace HTTP/HTTPS

Puede enviar un mensaje a las suscripciones de un tema mediante su publicación en el tema. Invoque `PublishAsync` y transfírale el ARN del tema y su mensaje.

```
var response = await snsClient.PublishAsync(topicArn, "This is your message");
```

Solución de problemas de SNS

Usar Delivery Status en la consola de Amazon SNS

La consola de Amazon SNS contiene la característica `Delivery Status`, que le permite recopilar comentarios sobre los intentos de entrega correctos o incorrectos de los mensajes a las plataformas de notificaciones de inserción móviles (Apple (APNS), Google (GCM), Amazon (ADM), Windows (WNS y MPNS) y Baidu).

Además, proporciona información adicional importante como la permanencia en Amazon SNS. Esta información se captura en `AmazonCloudWatchGrupo de log` que Amazon SNS crea automáticamente si esta característica está habilitada a través de la consola de Amazon SNS o de las API de Amazon SNS.

Para obtener instrucciones sobre cómo utilizar la característica `Delivery Status`, consulte [Using the Delivery Status feature of Amazon SNS](#) en AWS Mobile Blog.

Prácticas recomendadas para utilizar el SDK de AWS Mobile para .NET y Xamarin

Existen solo unos cuantos fundamentos y prácticas recomendadas útiles para saber cuándo utilizar el SDK de AWS Mobile para .NET y Xamarin.

- Utilice Amazon Cognito para obtener las credenciales de AWS en lugar de tener que codificar de forma rígida las credenciales en el código de la aplicación. Si codifica las credenciales de forma rígida en su aplicación podría acabar revelándolas, lo que permitiría que otras personas realizaran llamadas a AWS utilizándolas. Para obtener información sobre cómo utilizar Amazon Cognito para conseguir las credenciales de AWS, consulte [Setting Up the AWS Mobile SDK for .NET and Xamarin \(p. 4\)](#).
- Consulte [este artículo en el blog de AWS](#), para obtener más información acerca de las prácticas recomendadas sobre el uso de S3.
- Consulte la sección [DynamoDB Best Practices](#) en la Guía para desarrolladores de DynamoDB para obtener más información acerca de las prácticas recomendadas sobre el uso de DynamoDB.

Queremos ayudar a nuestros clientes a alcanzar el éxito y nos gusta conocer sus comentarios, por lo que le agradeceríamos que los [publicar en los foros de AWS](#) o [abrir un problema en GitHub](#).

Biblioteca de documentación del servicio de AWS

Cada servicio de AWS Mobile SDK for .NET and Xamarin tiene una guía independiente para desarrolladores y una referencia de la API del servicio con información adicional que podría resultarle de utilidad.

Amazon Cognito Identity

- [Guía para desarrolladores de Cognito](#)
- [Referencia de la API del servicio de identidad de Cognito](#)

Amazon Cognito Sync

- [Guía para desarrolladores de Cognito](#)
- [Referencia de la API del servicio de sincronización de Cognito](#)

Amazon Mobile Analytics

- [Guía para desarrolladores de Mobile Analytics](#)
- [Referencia de la API del servicio de Mobile Analytics](#)

Simple Storage Service (Amazon S3)

- [Guía para desarrolladores de S3](#)
- [Guía de introducción a S3](#)
- [Referencia de la API del servicio de S3](#)

Amazon DynamoDB

- [Guía para desarrolladores de DynamoDB](#)
- [Guía de introducción a DynamoDB](#)
- [Referencia de la API de servicio de DynamoDB](#)

Amazon Simple Notification Service (SNS)

- [Guía para desarrolladores de SNS](#)
- [Referencia de la API de servicio de SNS](#)

Otros enlaces útiles

- [Glosario de AWS](#)
- [Acerca de las credenciales de AWS](#)

Solución de problemas

En este tema se describen algunas ideas para solucionar problemas que se produzcas al utilizar el AWS Mobile SDK for .NET and Xamarin.

Comprobar que un rol de IAM tiene los permisos requeridos

Cuando llame a los servicios de AWS, su aplicación debe usar una identidad de un grupo de identidades Cognito. Cada identidad del grupo está asociada a un rol de IAM (Identity and Access Management).

Un rol tiene uno o varios archivos de políticas asociados que especifican a qué recursos de AWS tienen acceso los usuarios asignados al rol. De forma predeterminada, se crean dos roles por grupo de identidades: uno para los usuarios sin autenticar y otro para los usuarios autenticados.

Tendrá que modificar el archivo de política existente o asociar un nuevo archivo de política con los permisos requeridos por la aplicación. Si su aplicación admite usuarios autenticados y no autenticados, se deben conceder a los dos roles permisos para obtener acceso a los recursos de AWS que necesita la aplicación.

El archivo de política siguiente muestra cómo dar acceso a un bucket de S3:

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

El archivo de política siguiente muestra cómo dar acceso a una base de datos de DynamoDB:

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

```
} ]
```

Para obtener más información acerca de la especificación de políticas, consulte [Políticas de IAM](#).

Uso de un depurador de proxy HTTP

Si el servicio de AWS al que llama la aplicación tiene un punto de enlace HTTP o HTTPS, puede utilizar un depurador de proxy HTTP/HTTPS para ver las solicitudes y las respuestas con el fin de obtener más información acerca de lo que está sucediendo. Hay diversos depuradores de proxy HTTP disponibles, como:

- [Charles](#): un proxy de depuración web para Windows y OSX
- [Fiddler](#): un proxy de depuración web para Windows

Tanto Charles como Fiddler requieren algunos ajustes para poder ver el tráfico cifrado SSL. Lea la documentación de estas herramientas para obtener más información. Si utiliza un proxy de depuración web que no se puede configurar para mostrar el tráfico cifrado, abra el archivo `aws_endpoints_json` y establezca la etiqueta HTTP del servicio de AWS que necesita para depurar en `true`.

Historial de revisión

En la siguiente tabla se describen los cambios importantes que se han realizado en la documentación desde la última versión del SDK para móviles de AWS para .NET y Xamarin.

- Versión de API: 27-08-2015
- Última actualización de documentación: 2021-02-23

Cambio	Versión de API	Descripción	Fecha de lanzamiento de la nueva versión
Archived	27-08-2015	LaAWSEI SDK móvil para Xamarin está incluido en elAWS SDK for .NET. En esta guía se hace referencia a la versión archivada del SDK para móviles para Xamarin.	2021-02-23
Versión de GA	27-08-2015	Versión de GA	27-08-2015
Versión beta	28-07-2015	Versión beta	m28-07-2015