Panduan Developer Xamarin

AWS Mobile SDK

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

AWS Mobile SDK: Panduan Developer Xamarin

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

..... viii
Apa itu AWS Mobile SDK for .NET and Xamarin? 1
Panduan dan topik terkait 1
Isi referensi yang diarsipkan 1
Apa yang disertakan dalam AWS Mobile SDK for .NET and Xamarin? 2
Kompatibilitas 2
Bagaimana cara mendapatkan AWS Mobile SDK for .NET and Xamarin? 3
Tentang Layanan Mobile AWS 3
Menyiapkan AWS Mobile SDK for .NET and Xamarin 5
Prasyarat 5
Langkah 1: Mendapatkan Kredensial AWS 5
Langkah 2: Mengatur Izin 6
Langkah 3: Membuat Proyek Baru 8
Windows 8
OS X 8
Langkah 4: Memasang AWS Mobile SDK for .NET and Xamarin 8
Windows 8
Mac (OS X) 9
Langkah 5: Mengonfigurasi AWS Mobile SDK for .NET and Xamarin 10
Mengatur Pencatatan 10
Mengatur Titik Akhir Wilayah 11
Mengonfigurasi Pengaturan Proksi HTTP 11
Melakukan Koreksi Clock Skew 11
Langkah Berikutnya 12
Memulai AWS Mobile SDK for .NET and Xamarin 13
Menyimpan dan Mengambil File dengan Amazon S3 13
Penyiapan Proyek 13
Inisialisasi Klien S3 TransferUtility 15
Meng-unggah File ke Amazon S3 15
Mengunduh File dari Amazon S3 16
Sinkronkan Data Pengguna dengan Cognito Sync 16
Penyiapan Proyek 13
Inisialisasi CognitoSyncManager 17
Menyinkronkan Data pengguna 17

Menyimpan dan Mengambil Data dengan DynamoDB 19
Penyiapan Proyek 13
Inisialisasi AmazonDynamo DBClient 21
Membuat Kelas 22
Menyimpan Item 22
Mengambil Item 23
Memperbarui Item 23
Menghapus Item 23
Melacak Data Penggunaan Aplikasi dengan Amazon Mobile Analytics 23
Penyiapan Proyek 13
Inisialisasi MobileAnalyticsManager 25
Melacak Peristiwa Sesi 25
Menerima Notifikasi Push menggunakan SNS (Xamarin iOS) 26
Penyiapan Proyek 13
Buat Klien SNS 29
Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh 29
Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda 30
Menerima Notifikasi Push menggunakan SNS (Xamarin Android) 30
Penyiapan Proyek 13
Buat klien SNS 29
Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh 29
Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda 30
Identitas Amazon Cognito 36
Apakah Identitas Amazon Cognito Itu? 36
Menggunakan Penyedia Publik untuk Melakukan Autentikasi Pengguna 36
Menggunakan Identitas Di-autentikasi Developer 36
Amazon Cognito Sync 38
Apakah Amazon Cognito Sync Itu? 38
Amazon Mobile Analytics 39
Konsep kunci 39
Tipe Laporan 39
Penyiapan Proyek 13
Prasyarat 5
Mengonfigurasi Pengaturan Mobile Analytics 24
Mengintegrasikan Mobile Analytics dengan Aplikasi Anda 41
Membuat Aplikasi di Konsol Mobile Analytics 24

Buat MobileAnalyticsManager Klien 41
Mencatat Peristiwa Monetisasi 42
Mencatat Peristiwa Kustom 42
Sesi Pecatatan 43
Amazon Simple Storage Service (S3) 45
Apa itu S3? 45
Konsep Kunci 39
Bucket 45
Objek 45
Metadata Objek 46
Penyiapan Proyek 13
Prasyarat 5
Buat Bucket S3 46
Mengatur Izin untuk S3 14
(opsional) Mengonfigurasi Versi Tanda Tangan untuk Permintaan S3 15
Mengintegrasikan S3 dengan Aplikasi Anda 48
Menggunakan S3 Transfer Utility 48
Inisialisasi TransferUtility 48
(opsional) Konfigurasikan TransferUtility 49
Mengunduh File 49
Meng-unggah sebuah file 50
Menggunakan Service Level S3 APIs 50
Menginisialisasi Amazon S3 Client 50
Mengunduh File 49
Meng-unggah File 50
Hapus Item 23
Menghapus Banyak Item 52
Buat Daftar Bucket 53
Daftar Objek 54
Dapatkan Wilayah Bucket 54
Dapatkan Kebijakan Bucket 55
Amazon DynamoDB 56
Apa itu Amazon DynamoDB? 56
Konsep kunci 39
Tabel 56
Item dan Atribut 56

Tipe Data 57
Kunci Utama 57
Indeks Sekunder 57
Melakukan Kueri dan Pemindaian 58
Penyiapan Proyek 13
Prasyarat 5
Membuat Tabel DynamoDB 19
Mengatur Izin untuk DynamoDB 20
Mengintegrasikan DynamoDB dengan Aplikasi Anda 60
Menggunakan Model Dokumen 61
Membuat Klien DynamoDB 62
Operasi CRUD 62
Menggunakan Model Persistensi Objek 64
Gambaran Umum 65
Tipe Data yang Didukung 65
Membuat Klien DynamoDB 62
Operasi CRUD 62
Melakukan Kueri dan Pemindaian 58
Menggunakan Level Layanan DynamoDB APIs 69
Membuat Klien DynamoDB 62
Operasi CRUD 62
Melakukan Kueri dan Pemindaian 58
Amazon Simple Notification Service (SNS) 74
Konsep kunci 39
Topik 74
Langganan 74
Publikasi 74
Penyiapan Proyek 13
Prasyarat 5
Mengintegrasikan SNS dengan Aplikasi Anda 75
Mengirim Notifikasi Push (Xamarin Android) 75
Penyiapan Proyek 13
Buat klien SNS 29
Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh 29
Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda 30
Mengirim Notifikasi Push (Xamarin iOS) 80

Penyiapan Proyek 13
Buat Klien SNS 29
Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh 29
Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda 30
Mengirim dan Menerima Notifikasi SMS 84
Membuat Topik 84
Berlangganan Topik Menggunakan Protokol SMS 85
Memublikasikan Pesan 86
Mengirim Pesan ke Titik Akhir HTTP/HTTPS 87
Mengonfigurasi Titik Akhir HTTP/HTTPS Anda untuk Menerima Pesan Amazon SNS 87
Melangganakan Titik Akhir HTTP/HTTPS Anda ke Topik Amazon SNS Anda 88
Mengonfirmasi Langganan Anda 88
Kirim Pesan ke Titik Akhir HTTP/HTTPS 88
Memecahkan masalah SNS 89
Menggunakan Status Pengiriman di Konsol Amazon SNS 89
Praktik Terbaik untuk Menggunakan AWS Mobile SDK for .NET and Xamarin 90
Perpustakaan Dokumentasi Layanan AWS 90
Identitas Amazon Cognito 36
Sinkronisasi Amazon Cognito 3
Amazon Mobile Analytics 39
Amazon S3 91
Amazon DynamoDB 91
Amazon Simply Notification Service (SNS) 91
Tautan Bermanfaat Lainnya 91
Pemecahan masalah 92
Pastikan IAM role Memiliki Izin yang Diperlukan 92
Menggunakan HTTP Proxy Debugger 93
Riwayat Dokumen 94

AWS Mobile SDK untuk Xamarin sekarang termasuk dalam. AWS SDK untuk .NET Panduan ini menjadi referensi versi yang diarsipkan dari Mobile SDK for Xamarin.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.

Apa itu AWS Mobile SDK for .NET and Xamarin?

AWS Mobile SDK untuk Xamarin termasuk dalam file. SDK untuk .NET Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK untuk .NET .

Panduan ini tidak lagi diperbarui—ia menjadi referensi versi yang diarsipkan dari Mobile SDK for Xamarin.

Panduan dan topik terkait

- Untuk pengembangan aplikasi seluler dan front-end, kami merekomendasikan untuk menggunakan AWS Amplify.
- Untuk pertimbangan khusus dalam menggunakan aplikasi Xamarin Anda, lihat Pertimbangan khusus untuk dukungan Xamarin di Panduan Pengembang. AWS SDK untuk .NET AWS SDK untuk .NET
- Untuk tujuan referensi, Anda dapat menemukan versi arsip SDK AWS Seluler untuk Xamarin di. GitHub

Isi referensi yang diarsipkan

AWS Mobile SDK yang diarsipkan untuk.NET dan Xamarin menyediakan seperangkat pustaka, contoh kode, dan dokumentasi .NET untuk membantu pengembang membangun aplikasi seluler yang terhubung untuk:

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

Aplikasi seluler yang ditulis menggunakan AWS Mobile SDK untuk.NET dan Xamarin APIs memanggil platform asli sehingga memiliki tampilan dan nuansa aplikasi asli. Pustaka .NET di SDK menyediakan pembungkus C# di sekitar AWS REST. APIs

Apa yang disertakan dalam AWS Mobile SDK for .NET and Xamarin?

Layanan AWS yang di-support saat ini mencakup, namun tidak terbatas pada:

- Amazon Cognito
- Amazon S3
- Amazon DynamoDB
- Amazon Mobile Analytics
- Layanan Pemberitahuan Sederhana Amazon

Layanan ini memungkinkan Anda untuk meng-autentikasi pengguna, menyimpan data pemain dan game, menyimpan objek di cloud, menerima notifikasi push, dan mengumpulkan serta menganalisis data penggunaan.

AWS Mobile SDK for .NET and Xamarin juga memungkinkan Anda untuk menggunakan sebagian besar layanan AWS yang di-support oleh AWS SDK for .NET. Layanan AWS khusus untuk pengembangan mobile dijelaskan dalam panduan developer ini. Untuk informasi selengkapnya tentang AWS SDK for .NET, lihat:

- Panduan Memulai AWS SDK for .NET
- Panduan Pengembang AWS SDK for .NET
- AWS SDK for .NET API Referensi

Kompatibilitas

AWS Mobile SDK for .NET and Xamarin dikirimkan sebagai Portable Class Library (PCL). PCL Support ditambahkan di Xamarin.Android 4.10.1 dan Xamarin.iOS 7.0.4. Proyek Perpustakaan Portabel dibangun ke Visual Studio.

IDEs

Untuk informasi lebih lanjut tentang penggunaan IDEs dengan versi arsip Xamarin SDK, lihat. Menyiapkan AWS Mobile SDK for .NET and Xamarin

Bagaimana cara mendapatkan AWS Mobile SDK for .NET and Xamarin?

Untuk mendapatkan AWS Mobile SDK for .NET and Xamarin, lihat Menyiapkan AWS Mobile SDK for .NET and Xamarin. AWS Mobile SDK untuk.NET dan NuGet Xamarin didistribusikan sebagai paket. Anda dapat menemukan daftar lengkap paket layanan AWS di paket AWS SDK pada NuGet atau di AWS SDK for GitHub .NET Repository.

Tentang Layanan Mobile AWS

Identitas Amazon Cognito

Semua panggilan yang dibuat untuk AWS memerlukan kredensial AWS. Daripada melakukan hard-coding atas kredensial Anda ke aplikasi, sebaiknya gunakan Identitas Amazon Cognito untuk memberikan kredensial AWS untuk aplikasi Anda. Ikuti instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin untuk mendapatkan kredensial AWS melalui Amazon Cognito.

Cognito juga mengizinkan Anda untuk meng-autentikasi pengguna menggunakan penyedia log-in publik seperti Amazon, Facebook, Twitter, dan Google serta penyedia yang men-support OpenID Connect. Cognito juga berfungsi dengan pengguna yang tidak di-autentikasi. Cognito memberikan kredensial sementara dengan hak akses terbatas yang Anda tentukan dengan peran Identity and Access Management (IAM) role. Cognito dikonfigurasi dengan membuat kolam identitas baru yang di-associate dengan IAM role. IAM role menentukan sumber daya/layanan yang dapat diakses aplikasi Anda.

Untuk memulai Identitas Cognito, lihat Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Untuk mem-pelajari selengkapnya tentang Identitas Cognito, lihat Identitas Amazon Cognito.

Amazon Cognito Sync

Cognito Sync adalah sebuah layanan AWS dan perpustakaan klien yang memungkinkan sinkronisasi lintas-perangkat data pengguna terkait aplikasi. Anda dapat menggunakan API Cognito Sync untuk menyinkronkan data profil pengguna di seluruh perangkat dan di seluruh penyedia login - Amazon, Facebook, Google, dan penyedia identitas kustom Anda sendiri.

Untuk memulai Cognito Sync, lihat Menyinkronkan Data Pengguna dengan Cognito Sync.

Untuk informasi selengkapnya tentang Cognito Sync, lihat Amazon Cognito Sync.

Mobile Analytics

Amazon Mobile Analytics memungkinkan Anda mengumpulkan, memvisualisasikan, dan memahami penggunaan aplikasi untuk aplikasi seluler Anda. Laporan tersedia untuk metrik pengguna aktif, sesi, retensi, pendapatan dalam aplikasi, dan peristiwa kustom, dan dapat di-filter berdasarkan platform dan rentang tanggal. Amazon Mobile Analytics dibangun untuk diskalakan dengan bisnis Anda dan dapat mengumpulkan dan memproses miliaran peristiwa dari jutaan titik akhir.

Untuk mulai menggunakan Mobile Analytics, lihat Melacak Data Penggunaan Aplikasi dengan Amazon Mobile Analytics.

Untuk informasi lebih lanjut tentang Mobile Analytics, lihat Amazon Mobile Analytics.

Dynamo DB

Amazon DynamoDB adalah layanan basis data yang cepat, sangat dapat diskalakan, sangat tersedia, hemat biaya, dan bukan basis data relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi.

Untuk memulai menggunakan Dynamo DB, lihat Menyimpan dan Mengambil Data dengan DynamoDB.

Untuk informasi selengkapnya tentang Dynamo DB, lihat Amazon DynamoDB.

Amazon Simple Notification Service

Amazon Simple Notification Service (SNS) adalah layanan notifikasi push yang cepat, fleksibel, dan terkelola penuh yang memungkinkan Anda mengirim pesan individu atau pesan fan-out ke sejumlah besar penerima. Amazon Simple Notification Service membuat pengiriman notifikasi push ke pengguna perangkat mobile, penerima email atau bahkan pengiriman pesan ke layanan terdistribusi lainnya menjadi sederhana dan hemat biaya.

Untuk mulai menggunakan SNS untuk Xamarin iOS, lihat Menerima Notifikasi Push menggunakan SNS (Xamarin iOS).

Untuk mulai menggunakan SNS untuk Xamarin Android, lihat Menerima Notifikasi Push menggunakan SNS (Xamarin Android).

Untuk informasi tentang SNS, lihat Amazon Simple Notification Service (SNS).

Menyiapkan AWS Mobile SDK for .NET and Xamarin

Anda dapat menyiapkan AWS Mobile SDK for .NET and Xamarin dan mulai membangun sebuah proyek baru atau Anda dapat mengintegrasikan SDK dengan proyek yang ada. Anda juga dapat melakukan klon dan menjalankan Sampel untuk mengetahui cara kerja SDK. Ikuti langkah-langkah berikut untuk menyiapkan dan mulai menggunakan AWS Mobile SDK for .NET and Xamarin.

Prasyarat

Sebelum Anda dapat menggunakan AWS Mobile SDK for .NET and Xamarin, Anda harus melakukan hal-hal berikut:

- Membuat Akun AWS.
- Pasang Xamarin.

Setelah Anda menyelesaikan prasyarat:

1. Mendapatkan kredensial AWS dengan menggunakan Amazon Cognito.
2. Mengatur izin yang diperlukan untuk setiap layanan AWS yang akan Anda gunakan dalam aplikasi Anda.
3. Membuat proyek baru di IDE Anda.
4. Memasang AWS Mobile SDK for .NET and Xamarin.
5. Mengonfigurasi AWS Mobile SDK for .NET and Xamarin.

Langkah 1: Mendapatkan Kredensial AWS

Untuk membuat panggilan ke AWS dalam aplikasi Anda, Anda harus terlebih dahulu mendapatkan kredensial AWS. Anda melakukan ini dengan menggunakan Amazon Cognito, layanan AWS yang memungkinkan aplikasi Anda untuk mengakses layanan di SDK tanpa harus menanamkan kredensial AWS privat Anda dalam aplikasi.

Untuk memulai Amazon Cognito, Anda perlu membuat kolam identitas. Kolam identitas adalah penyimpanan informasi yang spesifik untuk akun Anda dan diidentifikasi berdasarkan ID kolam identitas unik yang terlihat seperti berikut ini.:

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

1. Masuk ke Konsol Amazon Cognito, pilih Mengelola Identitas Federasian, lalu pilih Buat kolam identitas baru.
2. Masukkan nama untuk kolam identitas Anda dan pilih kotak centang untuk mengaktifkan akses ke identitas yang tidak diautentikasi. Pilih Buat Kolam untuk membuat kolam identitas Anda.
3. Pilih Izinkan untuk membuat dua peran default yang terkait dengan kolam identitas Anda, satu untuk pengguna yang tidak diautentikasi dan satu untuk pengguna diautentikasi. Peran default ini menyediakan kolam identitas Anda akses ke Amazon Cognito Sync dan Amazon Mobile Analytics.

Biasanya, Anda hanya akan menggunakan satu kolam identitas per aplikasi.

Setelah Anda membuat kolam identitas Anda, Anda akan mendapatkan kredensial AWS dengan membuat objek `CognitoAWSCredentials` (berikat identitas kolam ID Anda kepadanya) dan kemudian memberikannya ke konstruktor dari klien AWS sebagai berikut.:

```
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
);
```

Langkah 2: Mengatur Izin

Anda harus mengatur izin untuk setiap layanan AWS yang ingin Anda gunakan dalam aplikasi Anda. Pertama, Anda harus memahami bagaimana AWS melihat pengguna aplikasi Anda.

Ketika seseorang menggunakan aplikasi Anda dan membuat panggilan ke AWS, AWS menetapkan identitas untuk pengguna itu. Kolam identitas yang Anda buat di Langkah 1 adalah tempat di mana AWS menyimpan identitas ini. Ada dua jenis identitas: diautentikasi dan tidak diautentikasi. Identitas diautentikasi adalah milik pengguna yang diautentikasi oleh penyedia login publik (misalnya, Facebook, Amazon, Google). Identitas yang tidak diautentikasi adalah milik pengguna tamu.

Setiap identitas dikaitkan dengan AWS Identity and Access Management peran. Pada langkah 1, Anda membuat dua IAM role, satu untuk pengguna yang diautentikasi dan satu untuk pengguna yang tidak diautentikasi. Setiap IAM role memiliki satu atau lebih kebijakan yang melekat padanya yang menentukan layanan AWS apa yang ditetapkan oleh identitas tersebut untuk peran itu sehingga dapat mengaksesnya. Misalnya, kebijakan sampel berikut memberikan akses ke bucket Amazon S3.:

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

Untuk mengatur izin untuk layanan AWS yang ingin Anda gunakan dalam aplikasi Anda, Anda harus memodifikasi kebijakan yang melekat pada peran.

1. Buka Konsol IAM, dan kemudian pilih Peran. Ketik nama kolam identitas Anda di dalam kotak pencarian. Pilih IAM role yang ingin Anda konfigurasikan. Jika aplikasi Anda mengizinkan pengguna diautentikasi dan tidak diautentikasi, maka Anda perlu memberikan izin untuk kedua peran tersebut.
2. Klik Lampirkan Kebijakan, pilih kebijakan yang Anda inginkan, dan kemudian klik Lampirkan Kebijakan. Kebijakan default untuk IAM role yang Anda buat memberikan akses ke Amazon Cognito Sync dan Mobile Analytics.

Untuk informasi selengkapnya tentang cara membuat kebijakan atau memilih dari daftar kebijakan yang ada, lihat Kebijakan IAM.

Langkah 3: Membuat Proyek Baru

Windows

Anda dapat menggunakan Visual Studio untuk mengembangkan aplikasi Anda.

OS X

Anda dapat menggunakan Visual Studio untuk mengembangkan aplikasi Anda. Pengembangan iOS menggunakan Xamarin memerlukan akses ke Mac untuk menjalankan aplikasi Anda. Untuk informasi selengkapnya, lihat Memasang Xamarin.iOS pada Windows.

> **Note**
>
> IDE Rider komersial lintas platform dari JetBrains mencakup dukungan Xamarin pada platform Windows dan Mac.

Langkah 4: Memasang AWS Mobile SDK for .NET and Xamarin

Windows

Opsi 1: Memasang dengan menggunakan Konsol Manajer Paket

AWS Mobile SDK for .NET and Xamarin terdiri satu set unit .NET. Untuk memasang AWS Mobile SDK for .NET and Xamarin, jalankan perintah install-package untuk setiap paket di konsol Manajer Paket. Misalnya, untuk memasang Identitas Cognito, jalankan berikut ini.:

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

Paket Waktu Aktif AWS Core dan Identitas Amazon Cognito diperlukan untuk semua proyek. Berikut ini adalah daftar lengkap nama paket untuk setiap layanan.

Layanan	Nama paket
Waktu Aktif AWS Core	AWSSDK.Inti
Amazon Cognito Sync	AWSSDK.CognitoSync

Layanan	Nama paket
Identitas Amazon Cognito	AWSSDK.CognitoIdentity
Amazon DynamoDB	AWSSDK.Dinamo DBv2
Amazon Mobile Analytics	AWSSDK.MobileAnalytics
Amazon S3	AWSSDK.S3
Amazon SNS	AWSSDK.SimpleNotificationService

Untuk menyertakan paket pra-rilis, termasuk argumen baris perintah `-Pre` ketika memasang paket sebagai berikut.:

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

Anda dapat menemukan daftar lengkap paket layanan AWS di paket AWS SDK pada NuGet atau di AWS SDK for GitHub .NET Repository.

Opsi 2: Memasang dengan Menggunakan IDE Anda

Di Visual Studio

1. Klik kanan proyek, dan kemudian klik Kelola NuGet Paket.
2. Cari nama paket yang ingin Anda tambahkan ke proyek Anda. Untuk menyertakan NuGet paket prelease, pilih Include Prelease. Anda dapat menemukan daftar lengkap paket layanan AWS di paket AWS SDK. NuGet
3. Pilih paket, dan kemudian pilih Pasang.

Mac (OS X)

Di Visual Studio

1. Klik kanan pada folder paket, dan kemudian pilih Tambah Paket.
2. Cari nama paket yang ingin Anda tambahkan ke proyek Anda. Untuk menyertakan paket prelease, pilih Tampilkan NuGet paket pra-rilis. Anda dapat menemukan daftar lengkap paket layanan AWS di paket AWS SDK. NuGet

3. Pilih kotak centang di samping paket yang Anda inginkan, lalu pilih Tambahkan Paket.

> **Important**
>
> Jika Anda mengembangkan menggunakan Portable Class Library, Anda juga harus menambahkan AWSSDK NuGet paket.Core ke semua proyek yang berasal dari Portable Class Library.

Langkah 5: Mengonfigurasi AWS Mobile SDK for .NET and Xamarin

Mengatur Pencatatan

Anda menetapkan pengaturan pencatatan dengan menggunakan kelas `Amazon.AWSConfigs` dan kelas `Amazon.Util.LoggingConfig`. Anda dapat menemukan ini di unit `AWSSdk.Core`, tersedia melalui Nuget Package Manager dalam Visual Studio. Anda dapat menempatkan kode pengaturan pencatatan di metode `OnCreate` di file `MainActivity.cs` untuk aplikasi Android atau file `AppDelegate.cs` untuk aplikasi iOS. Anda juga harus menambahkan pernyataan `using Amazon` dan `using Amazon.Util` untuk file .cs.

Mengonfigurasi pengaturan pencatatan sebagai berikut.:

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

Saat Anda masuk SystemDiagnostics, kerangka kerja secara internal mencetak output ke System.Console. Jika Anda ingin mencatat tanggapan HTTP, atur bendera `LogResponses`. Nilai-nilai bisa Selalu, Tidak Pernah, atau OnError.

Anda juga dapat mencatat metrik performa untuk permintaan HTTP dengan menggunakan properti `LogMetrics`. Format pencatatan dapat ditentukan dengan menggunakan properti `LogMetricsFormat`. Nilai yang valid adalah JSON atau standar.

Mengatur Titik Akhir Wilayah

Mengonfigurasi wilayah default untuk semua klien layanan sebagai berikut.:

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

Hal ini akan menetapkan wilayah default untuk semua klien layanan di SDK. Anda dapat mengganti pengaturan ini dengan secara eksplisit menentukan wilayah pada saat membuat sebuah instans klien layanan, sebagai berikut.:

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

Mengonfigurasi Pengaturan Proksi HTTP

Jika jaringan Anda berada di belakang proxy, maka Anda dapat mengonfigurasi pengaturan proxy untuk permintaan HTTP sebagai berikut.

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

Melakukan Koreksi Clock Skew

Properti ini menentukan apakah SDK harus benar untuk clock skew klien dengan menentukan waktu server yang benar dan menerbitkan kembali permintaan dengan waktu yang benar.

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

Bidang ini diatur jika panggilan layanan mengakibatkan pengecualian dan SDK telah menentukan bahwa ada perbedaan antara waktu lokal dan server.

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

Untuk mem-pelajari selengkapnya tentang clock skew, lihat Koreksi Clock-skew di AWS Blog.

Langkah Berikutnya

Sekarang karena Anda telah mengatur AWS Mobile SDK for .NET and Xamarin, maka Anda dapat:

- Memulai. Baca Memulai AWS Mobile SDK for .NET and Xamarin untuk petunjuk quick-start tentang cara menggunakan dan mengonfigurasi layanan di AWS Mobile SDK for .NET and Xamarin.
- Jelajahi topik layanan. Pelajari tentang setiap layanan dan cara kerjanya di AWS Mobile SDK for .NET and Xamarin.
- Jalankan demo. Lihat Aplikasi Xamarin sampel yang menunjukkan kasus penggunaan umum. Untuk menjalankan aplikasi sampel, lakukan pengaturan atas AWS Mobile SDK for .NET and Xamarin seperti yang dijelaskan sebelumnya, dan kemudian ikuti petunjuk yang terdapat dalam file README dari masing-masing sampel.
- Pelajari APIs. Lihat | sdk-xamarin-ref |_.
- Ajukan pertanyaan: Posting pertanyaan di Forum AWS Mobile SDK atau buka masalah. GitHub

Memulai AWS Mobile SDK for .NET and Xamarin

AWS Mobile SDK for .NET and Xamarin menyediakan perpustakaan, sampel dan, dokumentasi yang diperlukan untuk memanggil layanan AWS dari aplikasi Xamarin.

Anda harus menyelesaikan semua instruksi di halaman Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum Anda mulai menggunakan layanan di bawah ini.

Layanan ini akan memulai topik yang akan memandu Anda melakukan hal-hal berikut:

Topik

- Menyimpan dan Mengambil File dengan Amazon S3
- Sinkronkan Data Pengguna dengan Cognito Sync
- Menyimpan dan Mengambil Data dengan DynamoDB
- Melacak Data Penggunaan Aplikasi dengan Amazon Mobile Analytics
- Menerima Notifikasi Push menggunakan SNS (Xamarin iOS)
- Menerima Notifikasi Push menggunakan SNS (Xamarin Android)

Untuk informasi tentang AWS Mobile lainnya SDKs, lihat AWS Mobile SDK.

Menyimpan dan Mengambil File dengan Amazon S3

Amazon Simple Storage Service (Amazon S3) menyediakan developer seluler dengan penyimpanan objek yang aman, tahan lama, sangat dapat diskalakan. Amazon S3 mudah digunakan, memiliki antarmuka layanan web yang sederhana untuk menyimpan dan mengambil data dalam jumlah berapa pun dan dari mana pun di web.

Tutorial di bawah ini menjelaskan cara mengintegrasikan S3 TransferUtility, utilitas tingkat tinggi untuk menggunakan S3 dengan aplikasi Anda. Untuk informasi selengkapnya tentang menggunakan aplikasi S3 dari Xamarin, lihat Amazon Simple Storage Service (S3).

Penyiapan Proyek

Prasyarat

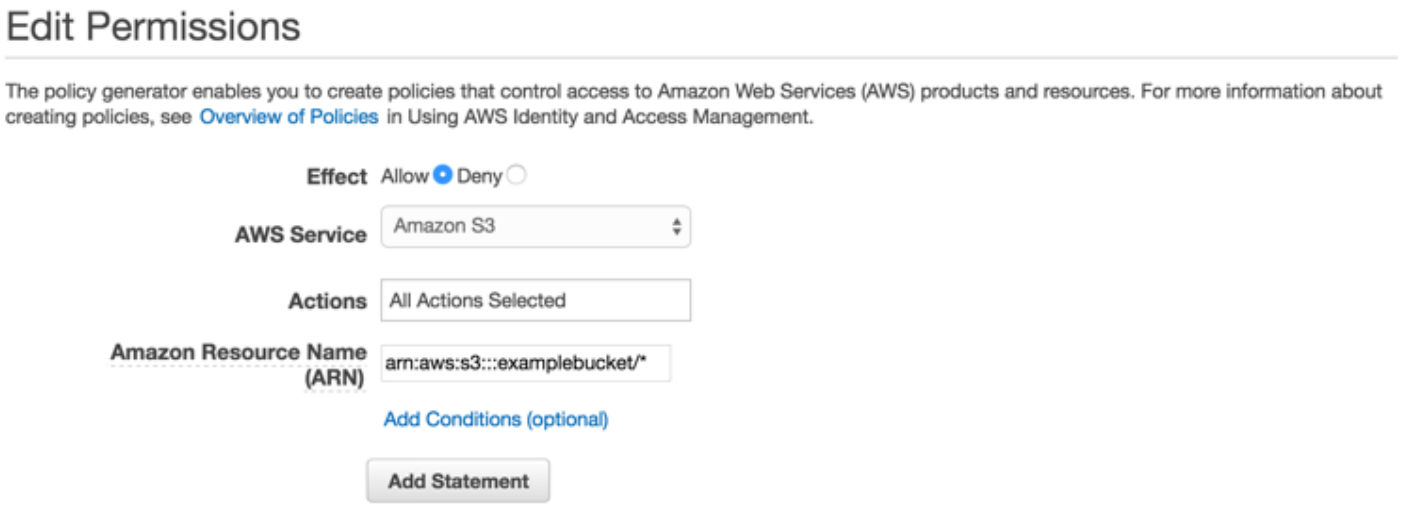
Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Tutorial ini juga mengasumsikan Anda telah membuat bucket S3. Untuk membuat bucket S3, kunjungi Konsol AWS S3.

Mengatur Izin untuk S3

Kebijakan IAM role default memberikan akses aplikasi Anda ke Amazon Mobile Analytics dan Amazon Cognito Sync. Agar kolam identitas Cognito Anda dapat mengakses Amazon S3, Anda harus mengubah peran kolam identitas.

1. Buka Konsol Identity and Access Management dan klik Peran yang di panel sebelah kiri.
2. Ketik nama kolam identitas Anda ke dalam kotak pencarian. Dua peran akan dicantumkan: satu untuk pengguna yang tidak di-autentikasi dan satu untuk pengguna di-autentikasi.
3. Klik peran untuk pengguna yang tidak di-autentikasi (hal ini akan menambahkan unauth pada nama kolam identitas Anda).
4. Klik Buat Kebijakan Peran, pilih Generator Kebijakan, dan kemudian klik Pilih.
5. Pada halaman Edit Izin, masukkan pengaturan yang ditunjukkan dalam gambar berikut, yang mengganti Amazon Resource Name (ARN) dengan punya Anda sendiri. ARN dari bucket S3 terlihat seperti `arn:aws:s3:::examplebucket/*` dan terdiri dari wilayah di mana bucket berada dan nama bucket. Pengaturan yang ditunjukkan di bawah ini akan memberikan kolam identitas Anda izin penuh untuk mengakses semua tindakan untuk bucket yang ditentukan.



1. Klik tombol Tambah Pernyataan dan kemudian klik Langkah Selanjutnya.
2. Penuntun akan menunjukkan konfigurasi yang telah Anda buat. Klik Terapkan Kebijakan.

Untuk informasi lebih lanjut tentang pemberian akses ke S3, lihat Memberikan Akses ke bucket Amazon S3.

Tambahkan NuGet Package for S3 ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk NuGet menambahkan paket S3 ke project Anda.

(opsional) Mengonfigurasi Versi Tanda Tangan untuk Permintaan S3

Setiap interaksi dengan Amazon S3 diautentikasi atau dilakukan secara anonim. AWS menggunakan algoritme Tanda Tangan Versi 4 atau Tanda Tangan Versi 2 untuk meng-autentikasi panggilan ke layanan.

Semua wilayah AWS baru yang dibuat setelah Januari 2014 hanya men-support Tanda Tangan Versi 4. Namun, banyak wilayah lama yang masih men-support permintaan Tanda Tangan Versi 4 dan Tanda Tangan Versi 2.

Jika bucket Anda berada di salah satu wilayah yang tidak mendukung permintaan Signature Version 2 seperti yang tercantum di halaman ini, Anda harus menyetel AWSConfigs S3. UseSignatureVersion4 properti untuk “benar” seperti ini:

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

Untuk informasi tentang versi Tanda Tangan AWS, lihat Meng-autentikasi Permintaan (Tanda Tangan AWS Versi 4).

Inisialisasi Klien S3 TransferUtility

Buat klien S3, berikan objek kredensial AWS Anda pada klien tersebut, dan kemudian berikan klien S3 ke utilitas transfer, seperti ini:

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

Meng-unggah File ke Amazon S3

Untuk meng-unggah file ke S3, panggil `Upload` pada objek Transfer Utility, yang memberikan parameter berikut:

- `file` - Nama string dari file yang ingin Anda unggah
- `bucketName` - Nama string dari bucket S3 untuk menyimpan file

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

Kode di atas mengasumsikan bahwa ada file di direktori Environment. SpecialFolder. ApplicationData. Unggahan secara otomatis menggunakan fungsionalitas unggah multi-bagian S3 pada file large untuk meningkatkan throughput.

Mengunduh File dari Amazon S3

Untuk mengunduh file dari S3, panggil `Download` pada objek Transfer Utility, yang memberikan parameter berikut:

- `file` - Nama string dari file yang ingin Anda unduh
- `bucketName` - Nama string dari bucket S3 tempat Anda ingin mengunduh file
- `key` - Sebuah string yang mewakili nama objek S3 (dalam kasus ini, sebuah file) yang akan diunduh

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

Untuk informasi lebih lanjut tentang mengakses Amazon S3 dari aplikasi Xamarin, lihat Amazon Simple Storage Service (S3).

Sinkronkan Data Pengguna dengan Cognito Sync

Amazon Cognito Sync memudahkan kita untuk menyimpan data pengguna seluler, seperti preferensi aplikasi atau status game di AWS Cloud tanpa harus menulis kode backend atau mengelola infrastruktur apa pun. Anda dapat menyimpan data secara lokal di perangkat pengguna yang

memungkinkan aplikasi Anda bekerja bahkan saat perangkat tersebut sedang offline. Anda juga dapat menyinkronkan data di seluruh perangkat pengguna sehingga pengalaman aplikasi mereka akan selalu konsisten terlepas dari perangkat yang mereka gunakan.

Tutorial di bawah ini menjelaskan cara mengintegrasikan Sync dengan aplikasi Anda.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Memberikan Akses ke Sumber Daya Cognito Sync Anda

Kebijakan default yang terkait dengan peran yang tidak di-autentikasi dan yang di-autentikasi yang Anda buat selama penyiapan memberikan aplikasi Anda akses ke Cognito Sync. Tidak diperlukan konfigurasi lebih lanjut.

Tambahkan NuGet Package for Cognito Sync ke Project Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk SyncManager NuGet menambahkan paket Cognito ke project Anda.

Inisialisasi CognitoSyncManager

Berikan penyedia kredensial Amazon Cognito Anda yang telah di-inisialisasi ke konstruktor `CognitoSyncManager`:

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

Menyinkronkan Data pengguna

Untuk menyinkronkan data pengguna yang tidak diautentikasi:

1. Buat set data.

2. Menambahkan data pengguna ke set data.
3. Menyinkronkan set data dengan cloud.

Membuat Set Data

Buat instans `Dataset`. Metode openOrCreate Dataset digunakan untuk membuat kumpulan data baru atau membuka instance kumpulan data yang sudah ada yang disimpan secara lokal di perangkat:

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

Menambahkan Data Pengguna ke Set Data

Data pengguna ditambahkan dalam bentuk pasangan kunci/nilai:

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

Set data Cognito berfungsi sebagai kamus, dengan nilai yang dapat diakses oleh kunci:

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

Sinkronisasi Set Data

Untuk menyinkronkan set data, panggil metode sinkronisasi-nya:

```
dataset.SynchronizeAsync();

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

Semua data yang ditulis ke set data akan disimpan secara lokal hingga set data disinkronkan. Kode yang ada di bagian ini mengasumsikan Anda menggunakan identitas Cognito yang tidak diautentikasi, sehingga ketika data pengguna disinkronkan dengan cloud, kode tersebut akan disimpan per perangkat. Perangkat ini memiliki ID perangkat yang terkait dengannya. Ketika data pengguna disinkronkan ke cloud, data tersebut akan dikaitkan dengan ID perangkat tersebut.

Untuk informasi selengkapnya tentang Cognito Sync, lihat Amazon Cognito Sync.

Menyimpan dan Mengambil Data dengan DynamoDB

Amazon DynamoDB adalah layanan basis data yang cepat, sangat dapat diskalakan, sangat tersedia, hemat biaya, dan bukan basis data relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi.

Tutorial di bawah ini menjelaskan cara untuk mengintegrasikan Model Persistensi Objek DynamoDB dengan aplikasi Anda, yang menyimpan objek di DynamoDB.

Penyiapan Proyek

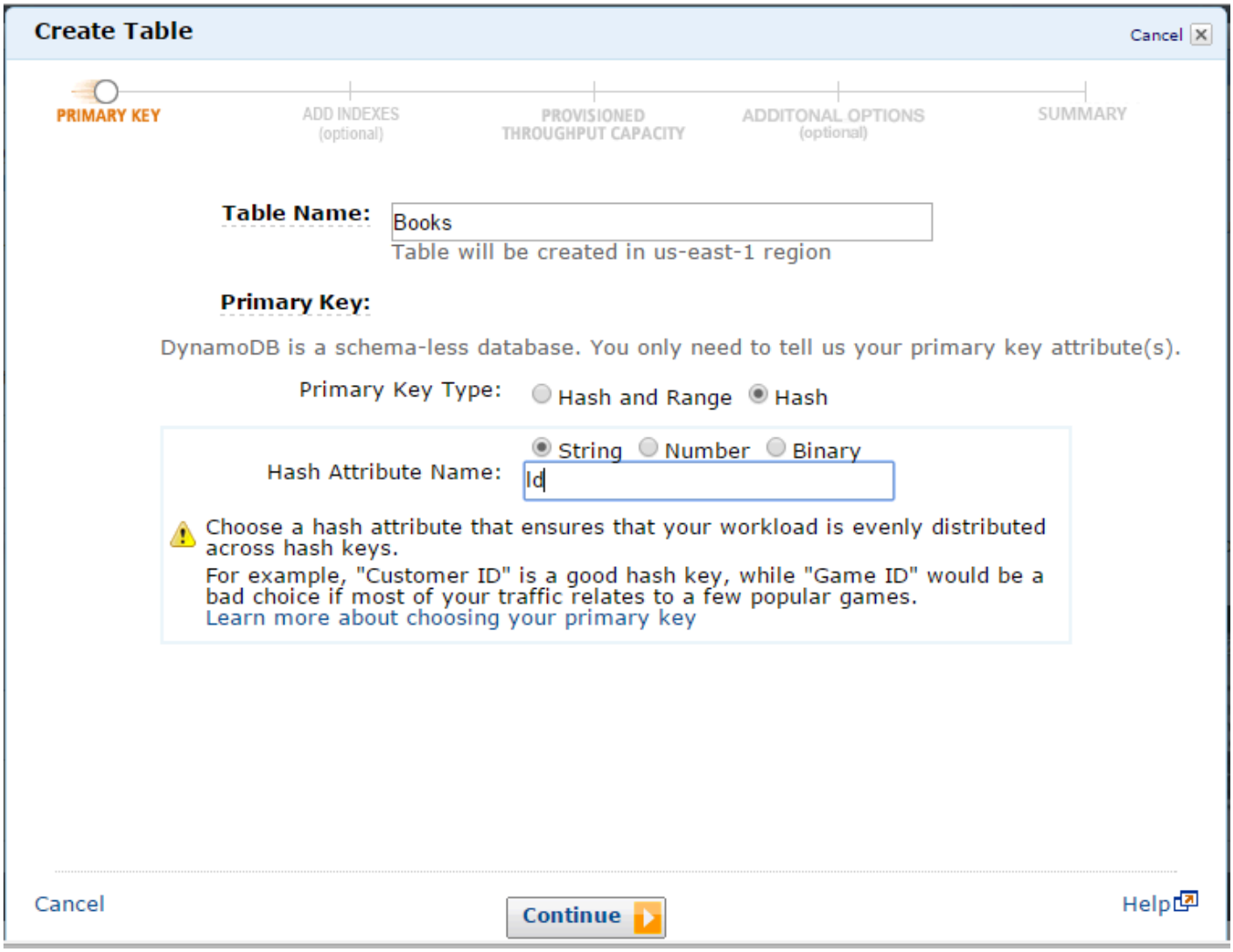
Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Membuat Tabel DynamoDB

Sebelum Anda dapat melakukan baca dan tulis data ke basis data DynamoDB, Anda harus membuat tabel. Saat membuat tabel Anda harus menentukan kunci primer. Kunci primer terdiri dari atribut hash dan atribut rentang opsional. Untuk informasi selengkapnya tentang cara atribut primer dan rentang digunakan, lihat Bekerja Dengan Tabel.

1. Buka Konsol DynamoDB dan klik Buat Tabel. Penuntun Buat Tabel akan muncul.
2. Tentukan nama tabel Anda, tipe kunci primer (Hash), dan nama atribut hash (“Id”) seperti yang ditunjukkan di bawah ini, dan kemudian klik Lanjutkan:

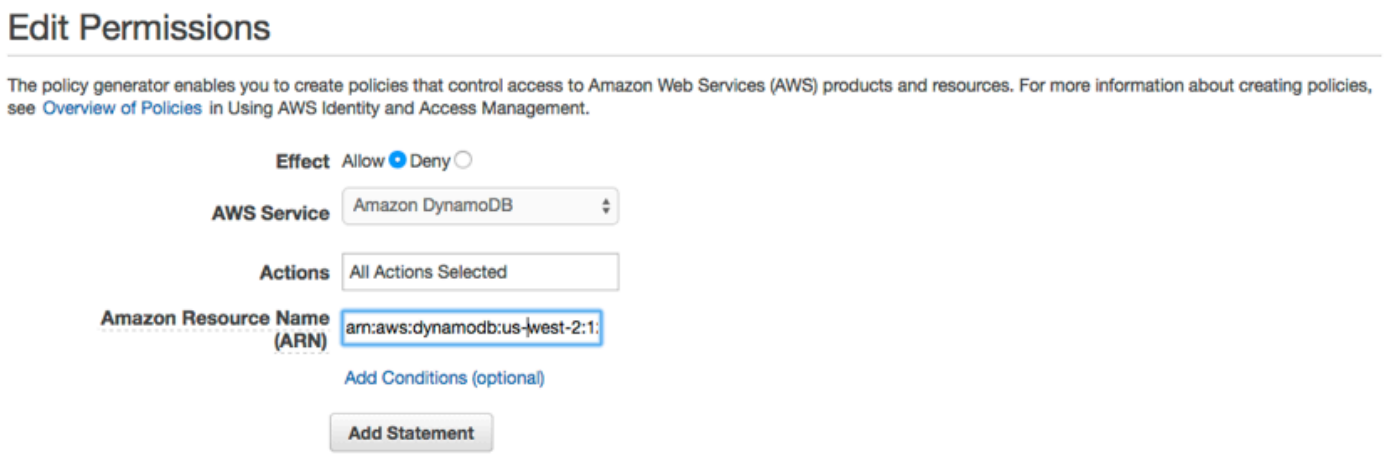


3. Biarkan bidang edit di layar berikutnya kosong dan klik Lanjutkan.
4. Setujui nilai default untuk Unit Kapasitas Baca dan Unit Kapasitas Tulis dan klik Lanjutkan.
5. Pada layar berikutnya, masukkan alamat email Anda di kotak teks Kirim notifikasi ke: dan klik Lanjutkan. Layar tinjauan akan muncul.
6. Klik Buat. Mungkin diperlukan waktu beberapa menit hingga tabel Anda selesai dibuat.

Mengatur Izin untuk DynamoDB

Agar kolam identitas Anda dapat mengakses Amazon DynamoDB, Anda harus mengubah peran kolam identitas tersebut.

1. Arahkan ke Konsol Identity and Access Management dan klik Peran di panel sebelah kiri. Cari nama kolam identitas Anda - dua peran akan dicantumkan, satu untuk pengguna yang tidak di-autentikasi dan satu untuk pengguna yang di-autentikasi.
2. Klik peran untuk pengguna yang tidak di-autentikasi (hal ini akan menambahkan “unauth” pada nama kolam identitas Anda) dan klik Buat Kebijakan Peran.
3. Pilih Generator Kebijakan dan klik Pilih.
4. Pada Edit Izin, masukkan pengaturan yang ditunjukkan pada citra berikut. Amazon Resource Name (ARN) dari tabel DynamoDB terlihat seperti `arn:aws:dynamodb:us-west-2:123456789012:table/Books` dan terdiri dari wilayah di mana tabel berada, nomor akun AWS pemilik, dan nama tabel dalam format `table/Books`. Untuk informasi selengkapnya tentang menentukan ARNs, lihat Nama Sumber Daya Amazon untuk DynamoDB.



5. Klik Tambah Pernyataan, dan kemudian klik Langkah Selanjutnya. Penuntun akan menampilkan konfigurasi yang dihasilkan.
6. Klik Terapkan Kebijakan.

Tambahkan NuGet paket untuk DynamoDB ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk NuGet menambahkan paket DynamoDB ke project Anda.

Inisialisasi AmazonDynamo DBClient

Teruskan penyedia kredensi Amazon Cognito yang diinisialisasi dan wilayah Anda ke konstruktor, lalu `AmazonDynamoDB` teruskan klien ke Dynamo: DBContext

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

```
DynamoDBContext context = new DynamoDBContext(client);
```

Membuat Kelas

Untuk menulis baris ke tabel, tentukan kelas untuk menyimpan data baris Anda. Kelas tersebut juga harus berisi properti yang menyimpan data atribut untuk baris dan akan dipetakan ke Tabel DynamoDB yang dibuat di konsol tersebut. Deklarasi kelas berikut menggambarkan kelas semacam itu:

```
[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; }
}
```

Menyimpan Item

Untuk menyimpan item, pertama buatlah objek:

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

Kemudian simpan:

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

Untuk memperbarui baris, modifikasi instans dari kelas `DDTableRow` dan panggil `AWSDynamoObjectMapper.save()` seperti yang ditunjukkan di atas.

Mengambil Item

Mengambil item menggunakan kunci primer:

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

Memperbarui Item

Untuk memperbarui item:

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

Menghapus Item

Untuk menghapus item:

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

Untuk informasi lebih lanjut tentang mengakses DynamoDB dari aplikasi Xamarin, lihat Amazon DynamoDB.

Melacak Data Penggunaan Aplikasi dengan Amazon Mobile Analytics

Amazon Mobile Analytics memungkinkan Anda mengukur penggunaan aplikasi dan pendapatan aplikasi. Dengan melacak kecenderungan kunci seperti pengguna baru vs. yang kembali, pendapatan aplikasi, retensi pengguna, dan peristiwa perilaku dalam aplikasi kustom, Anda dapat membuat keputusan didorong data untuk meningkatkan keterlibatan dan monetisasi aplikasi Anda.

Tutorial di bawah ini menjelaskan cara mengintegrasikan Mobile Analytics dengan aplikasi Anda.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Membuat Aplikasi di Konsol Mobile Analytics

Buka Konsol Amazon Mobile Analytics dan buat aplikasi. Catat nilai `appId`, karena Anda akan membutuhkannya nanti. Saat membuat aplikasi di Konsol Mobile Analytics, Anda harus menentukan ID kolam identitas Anda. Untuk petunjuk tentang cara membuat kolam identitas, lihat Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Untuk mem-pelajari selengkapnya tentang cara bekerja di konsol tersebut, lihat Panduan Pengguna Amazon Mobile Analytics.

Mengatur Izin untuk Mobile Analytics

Kebijakan default yang terkait dengan peran yang Anda buat selama penyiapan memberikan aplikasi Anda akses ke Mobile Analytics. Tidak diperlukan konfigurasi lebih lanjut.

Tambahkan NuGet Package untuk Mobile Analytics ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket Mobile NuGet Analytics ke project Anda.

Mengonfigurasi Pengaturan Mobile Analytics

Mobile Analytics menentukan beberapa pengaturan yang dapat dikonfigurasi dalam file awsconfig.xml:

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

- AllowUseDataNetwork - Sebuah boolean yang menentukan apakah peristiwa sesi dikirim pada jaringan data.

- DBWarningAmbang batas - Ini adalah batas pada ukuran database yang, setelah tercapai, akan menghasilkan log peringatan.
- Max DBSize - Ini adalah ukuran SQLIte Database. Ketika basis data mencapai ukuran maksimum, setiap peristiwa tambahan dihapus.
- MaxRequestSize - Ini adalah ukuran maksimum permintaan dalam Bytes yang harus dikirimkan dalam permintaan HTTP ke layanan analitik seluler.
- SessionTimeout - Ini interval waktu setelah aplikasi pergi ke latar belakang dan kapan sesi dapat dihentikan.

Pengaturan yang ditunjukkan di atas adalah nilai default untuk setiap item konfigurasi.

Inisialisasi MobileAnalyticsManager

Untuk menginisialisasi MobileAnalyticsManager, hubungi GetOrCreateInstance Anda`MobileAnalyticsManager`, meneruskan kredensi AWS Anda, wilayah Anda, ID aplikasi Mobile Analytics, dan objek konfigurasi opsional Anda:

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

Melacak Peristiwa Sesi

Android Xamarin

Menimpa metode `OnPause()` dan `OnResume()` aktivitas untuk mencatat peristiwa sesi.

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

protected override void OnPause()
{
    manager.PauseSession();
```

```
    base.OnPause();
}
```

Hal ini perlu dilaksanakan untuk setiap aktivitas dalam aplikasi Anda.

iOS Xamarin

Di AppDelegate .cs Anda:

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

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

Untuk informasi lebih lanjut tentang Mobile Analytics, lihat Amazon Mobile Analytics.

Menerima Notifikasi Push menggunakan SNS (Xamarin iOS)

Dokumen ini menjelaskan cara mengirim notifikasi push ke aplikasi Xamarin iOS menggunakan Amazon Simple Notification Service (SNS) dan AWS Mobile SDK for .NET and Xamarin.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Mengatur Izin untuk SNS

Ikuti Langkah 2 yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin untuk melampirkan kebijakan yang disebutkan di bawah ini ke peran aplikasi Anda. Hal ini akan memberikan aplikasi Anda izin yang tepat untuk mengakses SNS:

1. Buka Konsol IAM dan pilih IAM role yang akan dikonfigurasi.
2. Klik Lampirkan Kebijakan, pilih kebijakan SNSFull Akses Amazon dan klik Lampirkan Kebijakan.

> **Warning**
>
> Menggunakan Amazon SNSFull Access tidak disarankan di lingkungan produksi. Kami menggunakannya di sini untuk memungkinkan Anda untuk bangun dan berjalan dengan cepat. Untuk informasi selengkapnya tentang bagaimana menentukan izin untuk IAM role, lihat Gambaran Umum Izin IAM Role.

Dapatkan Keanggotaan dalam Program Developer iOS Apple

Anda akan perlu menjalankan aplikasi Anda pada perangkat fisik untuk menerima notifikasi push. Untuk menjalankan aplikasi Anda di perangkat, Anda harus memiliki keanggotaan di Keanggotaan Program Developer iOS Apple. Setelah Anda memiliki keanggotaan, Anda dapat menggunakan Xcode untuk membuat identitas penandatanganan. Untuk informasi selengkapnya, lihat dokumentasi Quick Start Distribusi Aplikasi.

Membuat Sertifikat iOS

Pertama, Anda harus membuat Sertifikat iOS. Kemudian, Anda harus membuat profil penyediaan yang dikonfigurasi untuk notifikasi push. Untuk melakukannya:

1. Buka Pusat Anggota Developer Apple, klik Sertifikat, Pengidentifikasi & Profil.
2. Klik Pengidentifikasi di bawah Aplikasi iOS, lalu klik tombol plus yang ada di sudut kanan atas halaman web untuk menambahkan ID Aplikasi iOS baru, lalu masukkan deskripsi ID Aplikasi.
3. Gulir ke bawah hingga ke bagian Tambah Akhiran ID dan pilih ID Aplikasi Eksplisit dan masukkan pengidentifikasi paket Anda.
4. Gulir ke bawah hingga ke bagian Layanan Aplikasi dan pilih bagian Notifikasi push.
5. Klik Lanjutkan.
6. Klik Kirim.
7. Klik Selesai.
8. Pilih ID Aplikasi yang baru saja dibuat dan kemudian klik Edit.
9. Gulir ke bawah hingga ke bagian Notifikasi Push. Klik Buat Sertifikat di bawah Sertifikat SSL Pengembangan.
10. Ikuti petunjuk untuk membuat Permintaan Penandatanganan Sertifikat (CSR), unggah permintaan, dan unduh sertifikat SSL yang akan digunakan untuk berkomunikasi dengan Layanan Notifikasi Apple (APNS).

11. Kembali ke halaman Sertifikat, Pengidentifikasi & Profil. Klik Semua di bawah Profil Penyediaan.
12. Klik tombol plus yang ada di sudut kanan atas untuk menambahkan profil penyediaan baru.
13. Pilih Aplikasi iOS, dan kemudian klik Lanjutkan.
14. Pilih ID Aplikasi Anda, dan kemudian klik Lanjutkan.
15. Pilih sertifikat developer Anda, dan kemudian klik Lanjutkan.
16. Pilih perangkat Anda, dan kemudian klik Lanjutkan.
17. Masukkan nama profil, dan kemudian klik Buat.
18. Unduh dan klik dua kali file penyediaan untuk memasang profil penyediaan.

Untuk informasi lebih lanjut tentang penyediaan profil yang dikonfigurasi untuk notifikasi push, lihat dokumentasi Mengonfigurasi Notifikasi Push.

Menggunakan Sertifikat untuk Membuat ARN Platform di Konsol SNS

1. Jalankan aplikasi KeyChain akses, pilih Sertifikat Saya di sisi kiri bawah layar, lalu klik kanan sertifikat SSL yang Anda buat untuk terhubung ke APNS dan pilih Ekspor. Anda akan diminta untuk menentukan nama untuk file dan kata sandi untuk melindungi sertifikat. Sertifikat tersebut akan disimpan dalam file P12.
2. Buka Konsol SNS dan klik Aplikasi yang ada di sisi kiri layar.
3. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
4. Masukkan Nama Aplikasi.
5. Pilih Pengembangan Apple untuk Platform notifikasi push.
6. Klik Pilih File dan pilih file P12 yang Anda buat saat Anda mengekspor sertifikat SSL.
7. Masukkan kata sandi yang Anda tentukan saat Anda mengekspor sertifikat SSL dan klik Memuat Kredensial dari File.
8. Klik Buat aplikasi platform.
9. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya. Anda akan membutuhkannya dalam langkah-langkah mendatang.

Tambahkan NuGet Package untuk SNS ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket NuGet Amazon Simple Notification Service ke project Anda.

Buat Klien SNS

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

Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh

Untuk mendaftarkan aplikasi, hubungi RegisterForRemoteNotifications UIApplication objek Anda, seperti yang ditunjukkan di bawah ini. Tempatkan kode berikut AppDelegate di.cs, masukkan ARN aplikasi platform Anda di mana diminta di bawah ini:

```
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 */
    });
  }
}
```

Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda

1. Buka Konsol SNS > Aplikasi.
2. Pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir.
3. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

Menerima Notifikasi Push menggunakan SNS (Xamarin Android)

Tutorial ini menjelaskan cara mengirim notifikasi push ke aplikasi Android Xamarin menggunakan Amazon Simple Notification Service (SNS) dan AWS Mobile SDK for .NET and Xamarin.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Mengatur Izin untuk SNS

Ikuti Langkah 2 yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin untuk melampirkan kebijakan yang disebutkan di bawah ini ke peran aplikasi Anda. Hal ini akan memberikan aplikasi Anda izin yang tepat untuk mengakses SNS:

1. Buka Konsol IAM dan pilih IAM role yang akan dikonfigurasi.
2. Klik Lampirkan Kebijakan, pilih kebijakan SNSFull Akses Amazon dan klik Lampirkan Kebijakan.

> **Warning**
>
> Menggunakan Amazon SNSFull Access tidak disarankan di lingkungan produksi. Kami menggunakannya di sini untuk memungkinkan Anda untuk bangun dan berjalan dengan cepat. Untuk informasi selengkapnya tentang bagaimana menentukan izin untuk IAM role, lihat Gambaran Umum Izin IAM Role.

Aktifkan Notifikasi Push pada Google Cloud

Pertama, tambahkan proyek Google API baru:

1. Buka Konsol Developer Google.
2. Klik Buat Proyek.
3. Di kotak Proyek Baru, masukkan nama proyek, catat ID proyek (Anda akan membutuhkannya nanti) dan klik Buat.

Selanjutnya, aktifkan layanan Google Cloud Messaging (GCM) untuk proyek Anda:

1. Di Konsol Developer Google, proyek baru Anda seharusnya sudah dipilih. Jika tidak, pilih di drop-down yang ada di bagian atas halaman.
2. Pilih APIs & auth dari bilah samping di sisi kiri halaman.
3. Di kotak pencarian, ketik “Google Cloud Messaging untuk Android” dan klik tautan Google Cloud Messaging untuk Android.
4. Klik Aktifkan API.

Akhirnya, dapatkan Kunci API:

1. Di Google Developers Console, pilih APIs & auth > Credentials.
2. Di bawah Akses API Publik, klik Buat kunci baru.
3. Pada dialog Buat kunci baru, klik Kunci server.
4. Dalam dialog yang dihasilkan, klik Buat dan salin kunci API yang ditampilkan. Anda akan menggunakan kunci API ini untuk melakukan autentikasi nanti.

Menggunakan ID Proyek untuk Membuat ARN Platform di Konsol SNS

1. Buka Konsol SNS.
2. Klik Aplikasi di sisi kiri layar.
3. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
4. Masukkan Nama Aplikasi.
5. Pilih Google Cloud Messaging (GCM) untuk Platform notifikasi push.
6. Tempelkan kunci API ke dalam kotak teks berlabel Kunci API.
7. Klik Buat aplikasi platform.
8. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya.

Tambahkan NuGet Package untuk SNS ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket NuGet Amazon Simple Notification Service ke project Anda.

Buat klien SNS

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

Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh

Untuk mendaftar notifikasi jarak jauh di Android, Anda harus membuat BroadcastReceiver yang dapat menerima pesan Google Cloud. Ubah nama paket di bawah ini saat diminta untuk melakukannya:

```
[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);
      }
}
```

Di bawah ini adalah layanan yang menerima pemberitahuan push dari BroadcastReceiver dan menampilkan notifikasi di bilah notifikasi perangkat:

```
[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());
  }
}
```

Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda

1. Buka Konsol SNS > Aplikasi.
2. Pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir.
3. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

Identitas Amazon Cognito

Apakah Identitas Amazon Cognito Itu?

Identitas Amazon Cognito memungkinkan Anda untuk membuat identitas unik untuk pengguna Anda dan meng-autentikasi mereka dengan penyedia identitas. Dengan sebuah identitas, Anda dapat memperoleh kredensial AWS sementara dengan hak terbatas untuk menyinkronkan data dengan Amazon Cognito Sync, atau mengakses secara langsung layanan AWS lainnya. Identitas Amazon Cognito men-support penyedia identitas publik—Amazon, Facebook, dan Google—serta identitas yang tidak diautentikasi. Ia juga men-support identitas yang di-autentikasi developer, yang memungkinkan Anda mendaftar dan melakukan autentikasi pengguna melalui proses autentikasi backend Anda sendiri.

Untuk informasi selengkapnya tentang Identitas Cognito, lihat Panduan Developer Amazon Cognito.

Untuk informasi tentang ketersediaan Wilayah Autentikasi Cognito, lihat Ketersediaan Wilayah Layanan AWS.

Menggunakan Penyedia Publik untuk Melakukan Autentikasi Pengguna

Dengan Menggunakan Identitas Amazon Cognito, Anda dapat membuat identitas unik untuk pengguna Anda dan meng-autentikasi-nya untuk akses aman ke sumber daya AWS Anda seperti Amazon S3 atau Amazon DynamoDB. Identitas Amazon Cognito men-support penyedia identitas publik—Amazon, Facebook, Twitter/Digits, Google, atau penyedia OpenID mana pun yang kompatibel dengan Connect—serta identitas yang tidak diautentikasi.

Untuk informasi tentang cara menggunakan penyedia identitas publik seperti Amazon, Facebook, Twitter/Digits, atau Google untuk melakukan autentikasi pengguna, lihat Penyedia Eksternal dalam Panduan Developer Amazon Cognito.

Menggunakan Identitas Di-autentikasi Developer

Amazon Cognito men-support identitas di-autentikasi developer, selain federasi identitas web melalui Facebook, Google, dan Amazon. Dengan identitas yang di-autentikasi developer, Anda dapat mendaftar dan melakukan autentikasi pengguna melalui proses autentikasi yang ada milik Anda sendiri, sekaligus masih tetap menggunakan Amazon Cognito Sync untuk menyinkronkan data pengguna dan mengakses sumber daya AWS. Menggunakan identitas yang di-autentikasi developer

melibatkan interaksi antara perangkat pengguna akhir, backend Anda untuk autentikasi, dan Amazon Cognito.

Untuk informasi tentang identitas di-autentikasi developer, lihat Identitas Di-autentikasi Developer dalam Panduan Developer Amazon Cognito.

Amazon Cognito Sync

Apakah Amazon Cognito Sync Itu?

Cognito Sync adalah layanan AWS dan perpustakaan klien yang memungkinkan sinkronisasi data pengguna lintas perangkat (misalnya skor game, preferensi pengguna, status permainan). Anda dapat menggunakan API Cognito Sync untuk menyinkronkan data pengguna di seluruh perangkat. Untuk menggunakan Cognito Sync di aplikasi Anda, Anda harus menyertakan | dalam proyek Anda.

Untuk petunjuk tentang cara mengintegrasikan Amazon Cognito Sync di aplikasi Anda, lihat Panduan Developer Amazon Cognito Sync.

Amazon Mobile Analytics

Amazon Mobile Analytics adalah layanan untuk mengumpulkan, memvisualisasikan, memahami, dan mengekstraksi data penggunaan aplikasi dalam skala besar. Mobile Analytics dengan mudah menangkap data perangkat standar dan peristiwa kustom dan secara otomatis menghitung laporan atas nama Anda. Selain laporan agregatan yang tercantum di bawah ini, Anda juga dapat menyiapkan data Anda untuk diekspor ke Redshift dan S3 secara otomatis untuk dilakukan analisis lebih lanjut.

Dengan menggunakan Amazon Mobile Analytics, Anda dapat melacak perilaku pelanggan, melakukan penggabungan metrik, membuat visualisasi data, dan mengidentifikasi pola yang berarti.

Konsep kunci

Tipe Laporan

Di luar kotak, Mobile Analytics menyediakan laporan berikut di Konsol Mobile Analytics:

- Pengguna Aktif Harian (DAU), Pengguna Aktif Bulanan (MAU), dan Pengguna Baru
- Faktor Sticky (DAU dibagi MAU)
- Hitungan Sesi dan Sesi Rata-rata per Pengguna Aktif Harian
- Pendapatan Rata-rata per Pengguna Aktif Harian (ARPDAU) dan Pendapatan Rata-rata per Pengguna Aktif yang Membayar Harian (ARPPDAU)
- Retensi Hari 1, 3, dan 7 serta Retensi Pekan 1, 2, dan 3
- Peristiwa Kustom

Laporan-laporan ini disediakan melalui enam tab pelaporan di konsol tersebut:

- Ikhtisar — Lacak sembilan laporan yang dipilih sebelumnya di simple-to-review dasbor untuk mendapatkan gambaran cepat tentang keterlibatan: MAU, DAU, Pengguna Baru, Sesi Harian, Faktor Lengket, Retensi 1 Hari, ARPDAU, Pengguna Pembayaran Harian, ARPDAU.
- Pengguna Aktif — Melacak jumlah pengguna yang terlibat dengan aplikasi Anda secara harian dan bulanan dan memantau faktor sticky untuk mengukur keterlibatan, daya tarik, dan monetisasi.
- Sesi — Melacak seberapa sering aplikasi Anda digunakan pada hari tertentu dan seberapa sering setiap pengguna membuka aplikasi Anda dalam sehari.

- Retensi — Melacak tingkat di mana pelanggan kembali ke aplikasi Anda secara harian dan mingguan.
- Penghasilan — Melacak tren pendapatan dalam aplikasi untuk mengidentifikasi area untuk peningkatan monetisasi.
- Peristiwa kustom — Melacak tindakan pengguna yang ditetapkan khusus untuk aplikasi Anda.

Untuk mempelajari lebih lanjut tentang laporan Mobile Analytics dan bekerja di konsol Mobile Analytics, lihat Gambaran Umum Laporan Konsol Mobile Analytics dalam Panduan Developer Mobile Analytics.

Penyiapan Proyek

Prasyarat

Untuk menggunakan Mobile Analytics dalam aplikasi Anda, Anda harus menambahkan SDK ke proyek Anda. Untuk melakukannya, ikuti petunjuk di Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Mengonfigurasi Pengaturan Mobile Analytics

Mobile Analytics menentukan beberapa pengaturan yang dapat dikonfigurasi dalam file awsconfig.xml:

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

- SessionTimeout- Jika aplikasi tetap berada di latar belakang untuk waktu yang lebih lama dari klien Mobile Analytics menghentikan sesi saat ini dan sesi baru dibuat saat aplikasi kembali ke latar depan. SessionTimeout Kami merekomendasikan untuk menggunakan nilai mulai dari 5 sampai 10. Nilai default-nya adalah 5.
- Max DBSize - Ukuran maksimum database (dalam byte) yang digunakan untuk penyimpanan lokal acara. Jika ukuran basis data melebihi nilai ini, maka peristiwa tambahan akan diabaikan.

Kami merekomendasikan untuk menggunakan nilai mulai dari 1MB sampai 10MB. Nilai default-nya adalah 5242880 (5MB).

- DBWarningThreashold - Ambang Peringatan. Nilai yang valid berkisar antara 0 - 1. Jika nilai melebihi ambang batas peringatan, maka catatan akan dihasilkan. Nilai default-nya adalah 0.9.
- MaxRequestSize- Ukuran maksimum permintaan HTTP yang dibuat untuk layanan Mobile Analytics. Nilai-nya ditentukan dalam byte dan dapat berkisar antara 1-512KB. Nilai default-nya adalah 102400 (100KB). Jangan gunakan nilai yang lebih besar dari 512KB, hal ini dapat menyebabkan layanan menolak permintaan HTTP.
- AllowUseDataNetwork- Nilai yang menunjukkan apakah panggilan layanan diizinkan melalui jaringan data seluler. Gunakan opsi ini dengan hati-hati karena hal ini dapat menaikkan penggunaan data pelanggan.

Pengaturan yang ditunjukkan di atas adalah nilai default untuk setiap item konfigurasi.

Mengintegrasikan Mobile Analytics dengan Aplikasi Anda

Bagian di bawah ini menjelaskan cara mengintegrasikan Mobile Analytics dengan aplikasi Anda.

Membuat Aplikasi di Konsol Mobile Analytics

Buka Konsol Amazon Mobile Analytics dan buat aplikasi. Catat nilai `appId`, karena Anda akan membutuhkannya nanti. Saat membuat aplikasi di Konsol Mobile Analytics, Anda harus menentukan ID kolam identitas Anda. Untuk petunjuk tentang cara membuat kolam identitas, lihat Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Untuk mempelajari lebih lanjut tentang bekerja di Konsol Mobile Analytics, lihat Gambaran Umum Laporan Konsol Mobile Analytics dalam Panduan Developer Mobile Analytics.

Buat MobileAnalyticsManager Klien

Untuk menginisialisasi MobileAnalyticsManager, hubungi GetOrCreateInstance Anda`MobileAnalyticsManager`, meneruskan kredensi AWS Anda, wilayah Anda, ID aplikasi Mobile Analytics, dan objek konfigurasi opsional Anda:

```
// Initialize the MobileAnalyticsManager
analyticsManager = MobileAnalyticsManager.GetOrCreateInstance(
```

```
  cognitoCredentials,
  RegionEndpoint.USEast1,
  APP_ID,
  config
);
```

`APP_ID` dibuat untuk Anda selama penuntun pembuatan aplikasi. Kedua nilai ini harus sesuai dengan nilai yang ada di Konsol Mobile Analytics. `APP_ID` digunakan untuk mengelompokkan data Anda dalam grup di konsol Mobile Analytics. Untuk menemukan ID aplikasi setelah membuat aplikasi di konsol Mobile Analytics, telusuri Konsol Mobile Analytics, klik ikon roda gigi yang ada di sudut kanan atas layar. Ini akan menampilkan halaman Manajemen Aplikasi yang mencantumkan semua aplikasi terdaftar dan aplikasinya IDs.

Mencatat Peristiwa Monetisasi

AWS Mobile SDK for .NET and Xamarin menyediakan kelas `MonetizationEvent`, yang memungkinkan Anda membuat peristiwa monetisasi untuk melacak pembelian yang dilakukan dalam aplikasi mobile. Potongan kode berikut menunjukkan cara membuat peristiwa monetisasi:

```
// 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 monetiziation event
analyticsManager.RecordEvent(monetizationEvent);
```

Mencatat Peristiwa Kustom

Mobile Analytics memungkinkan Anda untuk menentukan peristiwa kustom. Peristiwa kustom ditentukan sepenuhnya oleh Anda; mereka membantu Anda melacak tindakan pengguna tertentu untuk aplikasi atau game Anda. Untuk informasi selengkapnya tentang peristiwa Kustom, lihat Peristiwa-Kustom.

Untuk contoh ini, katakanlah aplikasi kita adalah game, dan kita ingin mencatat peristiwa ketika pengguna menyelesaikan sebuah tingkat. Buat acara “LevelComplete” dengan membuat `AmazonMobileAnalyticsEvent` instance baru:

```
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);
```

Sesi Pecatatan

Xamarin iOS

Ketika aplikasi kehilangan fokus Anda dapat menjeda sesi. Untuk aplikasi iOS, di AppDelegate file.cs, ganti `DidEnterBackground` dan panggil `MobileAnalyticsManager.PauseSesion` dan `MobileAnalyticsManager.ResumeSession` seperti `WillEnterForeground` yang ditunjukkan pada cuplikan berikut:

```
public override void DidEnterBackground(UIApplication application)
{
  // ...
  _manager.PauseSesson();
  // ...
}

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

Xamarin Android

Untuk aplikasi Android, panggil `MobileAnalyticsManager.PauseSesion` metode OnPause () dan `MobileAnalyticsManager.ResumeSession` metode OnResume () seperti yang ditunjukkan pada cuplikan kode berikut:

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

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

Secara default, jika pengguna mengalihkan fokus dari aplikasi selama kurang dari 5 detik, dan beralih kembali ke aplikasi, maka sesi akan dilanjutkan. Jika pengguna mengalihkan fokus dari aplikasi selama 5 detik atau lebih, maka sesi baru akan dibuat. Pengaturan ini dapat dikonfigurasi dalam file konfigurasi aws_mobile_analytics.json dengan mengatur properti “SESSION_DELTA” menjadi jumlah detik untuk menunggu sebelum membuat sesi baru.

Amazon Simple Storage Service (S3)

Apa itu S3?

Amazon Simple Storage Service (Amazon S3), menyediakan developer seluler dengan penyimpanan objek yang aman, tahan lama, sangat dapat diskalakan. Amazon S3 mudah digunakan, memiliki antarmuka layanan web yang sederhana untuk menyimpan dan mengambil data dalam jumlah berapa pun dan dari mana pun di web. Dengan Amazon S3, Anda hanya membayar untuk penyimpanan yang benar-benar Anda gunakan. Tidak ada biaya minimum dan tidak ada biaya penyiapan.

Amazon S3 menyediakan penyimpanan objek hemat biaya untuk berbagai kasus penggunaan termasuk aplikasi cloud, distribusi konten, backup dan pengarsipan, pemulihan bencana, dan analitik big data.

Untuk informasi tentang ketersediaan Wilayah AWS S3, lihat Ketersediaan Wilayah Layanan AWS.

Konsep Kunci

Bucket

Setiap objek yang Anda simpan di Amazon S3 berada di dalam bucket. Anda dapat menggunakan bucket untuk mengelompokkan objek terkait dengan cara yang sama dengan Anda menggunakan direktori untuk file kelompok dalam sistem file. Bucket memiliki properti, seperti izin akses dan status versi, dan Anda dapat menentukan wilayah tempat Anda ingin mereka berada.

Untuk mem-pelajari selengkapnya tentang Bucket S3, lihat Bekerja dengan bucket dalam Panduan Developer S3.

Objek

Objek adalah data yang Anda simpan di Amazon S3. Setiap objek berada dalam bucket yang Anda buat di wilayah AWS tertentu.

Objek yang disimpan di suatu wilayah tidak pernah keluar dari wilayah tersebut kecuali Anda secara tegas mentransfernya ke wilayah lain. Misalnya, objek yang disimpan di wilayah EU (Ireland) tidak akan pernah meninggalkan wilayah itu. Objek yang disimpan di wilayah Amazon S3 secara fisik tetap

berada di wilayah tersebut. Amazon S3 tidak menyimpan salinan atau memindahkannya ke wilayah lain mana pun. Namun, Anda dapat mengakses objek dari mana saja, selama Anda memiliki izin yang diperlukan untuk melakukannya.

Objek dapat berupa jenis file apa pun: citra, backup, data, film, dll. Objek bisa berukuran sebesar 5 TB. Anda dapat memiliki jumlah objek tak terbatas dalam satu bucket.

Sebelum Anda dapat mengunggah objek ke Amazon S3, Anda harus memiliki izin tertulis ke bucket. Untuk informasi lebih lanjut tentang pengaturan izin bucket, lihat Mengedit Izin bucket dalam Panduan Developer S3.

Untuk mem-pelajari lebih lanjut tentang objek S3, lihat Bekerja dengan Objek dalam Panduan Developer S3.

Metadata Objek

Setiap objek di Amazon S3 memiliki serangkaian pasangan nilai-kunci yang mewakili metadata-nya. Ada dua tipe metadata:

- Metadata sistem – Kadang-kadang diproses oleh Amazon S3, misalnya, Content-Type, dan Content-Length.
- Metadata pengguna – Tidak pernah diproses oleh Amazon S3. Metadata pengguna disimpan bersama objek dan dikembalikan bersama objek. Dengan ukuran maksimum metadata pengguna adalah 2 KB, kunci dan nilainya harus sesuai dengan standar AS-ASCII.

Untuk mem-pelajari selengkapnya tentang metadata objek S3, lihat Mengedit Metadata Objek.

Penyiapan Proyek

Prasyarat

Untuk menggunakan Amazon S3 dalam aplikasi Anda, Anda harus menambahkan SDK ke proyek Anda. Untuk melakukannya, ikuti petunjuk di Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Buat Bucket S3

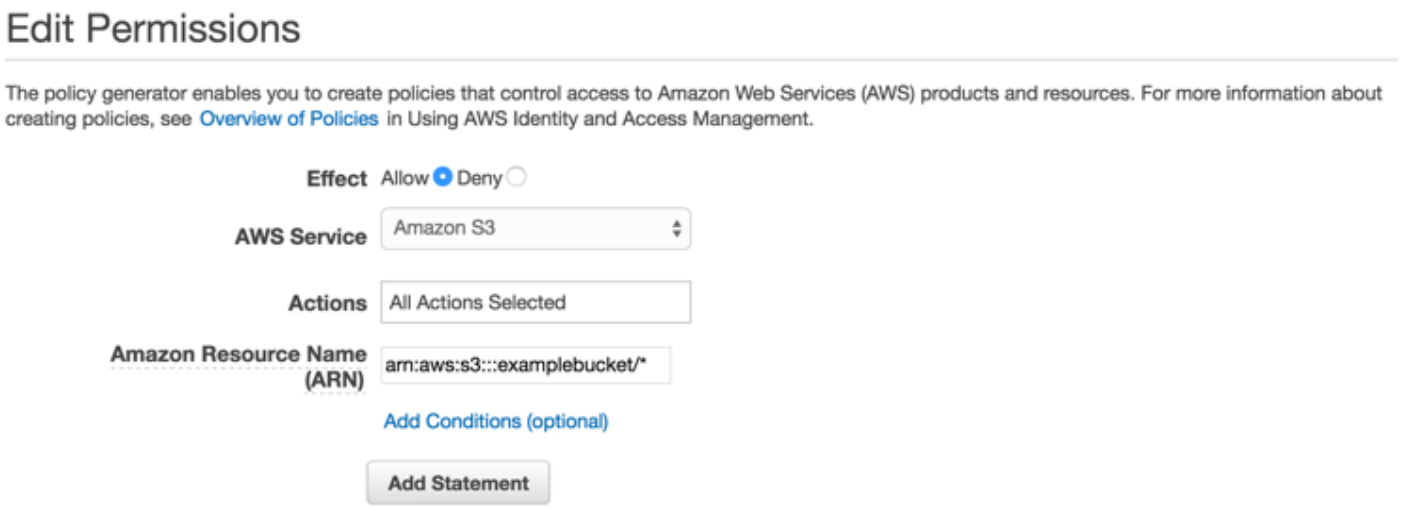
Amazon S3 menyimpan sumber daya aplikasi Anda di bucket S3 Amazon - kontainer penyimpanan cloud yang berada di wilayah tertentu. Setiap bucket Amazon S3 harus memiliki nama yang unik secara global. Anda dapat menggunakan Konsol Amazon S3 untuk membuat bucket.

1. Masuklah ke Konsol Amazon S3 dan klik Buat Bucket.
2. Masukkan nama bucket, pilih wilayah, dan klik Buat.

Mengatur Izin untuk S3

Kebijakan IAM role default memberikan akses aplikasi Anda ke Amazon Mobile Analytics dan Amazon Cognito Sync. Agar kolam identitas Cognito Anda dapat mengakses Amazon S3, Anda harus mengubah peran kolam identitas.

1. Buka Konsol Identity and Access Management dan klik Peran yang di panel sebelah kiri.
2. Ketik nama kolam identitas Anda ke dalam kotak pencarian. Dua peran akan dicantumkan: satu untuk pengguna yang tidak di-autentikasi dan satu untuk pengguna di-autentikasi.
3. Klik peran untuk pengguna yang tidak di-autentikasi (hal ini akan menambahkan unauth pada nama kolam identitas Anda).
4. Klik Buat Kebijakan Peran, pilih Generator Kebijakan, dan kemudian klik Pilih.
5. Pada halaman Edit Izin, masukkan pengaturan yang ditunjukkan dalam gambar berikut, yang mengganti Amazon Resource Name (ARN) dengan punya Anda sendiri. ARN dari bucket S3 terlihat seperti `arn:aws:s3:::examplebucket/*` dan terdiri dari wilayah di mana bucket berada dan nama bucket. Pengaturan yang ditunjukkan di bawah ini akan memberikan kolam identitas Anda izin penuh untuk mengakses semua tindakan untuk bucket yang ditentukan.



1. Klik tombol Tambah Pernyataan dan kemudian klik Langkah Selanjutnya.
2. Penuntun akan menunjukkan konfigurasi yang telah Anda buat. Klik Terapkan Kebijakan.

Untuk informasi lebih lanjut tentang pemberian akses ke S3, lihat Memberikan Akses ke bucket Amazon S3.

(opsional) Mengonfigurasi Versi Tanda Tangan untuk Permintaan S3

Setiap interaksi dengan Amazon S3 diautentikasi atau dilakukan secara anonim. AWS menggunakan algoritme Tanda Tangan Versi 4 atau Tanda Tangan Versi 2 untuk meng-autentikasi panggilan ke layanan.

Semua wilayah AWS baru yang dibuat setelah Januari 2014 hanya men-support Tanda Tangan Versi 4. Namun, banyak wilayah lama yang masih men-support permintaan Tanda Tangan Versi 4 dan Tanda Tangan Versi 2.

Jika bucket Anda berada di salah satu wilayah yang tidak mendukung permintaan Signature Version 2 seperti yang tercantum di halaman ini, Anda harus menyetel AWSConfigs S3. UseSignatureVersion4 properti untuk “benar” seperti ini:

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

Untuk informasi tentang versi Tanda Tangan AWS, lihat Meng-autentikasi Permintaan (Tanda Tangan AWS Versi 4).

Mengintegrasikan S3 dengan Aplikasi Anda

Ada dua cara untuk berinteraksi dengan S3 dalam aplikasi Xamarin Anda. Kedua metode tersebut diuraikan secara mendalam dalam topik-topik berikut:

Menggunakan S3 Transfer Utility

S3 Transfer Utility memudahkan kita untuk mengunggah dan mengunduh file ke S3 dari aplikasi Xamarin Anda.

Inisialisasi TransferUtility

Buat klien S3, berikan objek kredensial AWS Anda pada klien tersebut, dan kemudian berikan klien S3 ke utilitas transfer, seperti ini:

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

```
var transferUtility = new TransferUtility(s3Client);
```

(opsional) Konfigurasikan TransferUtility

Ada tiga properti opsional yang dapat Anda konfigurasi:

- ConcurrentServiceRequests- Menentukan berapa banyak thread aktif atau jumlah permintaan web asinkron bersamaan yang akan digunakan untuk mengunggah/mengunduh file. Nilai default adalah 10.
- MinSizeBeforePartUpload- Mendapat atau menetapkan ukuran bagian minimum untuk mengunggah bagian dalam byte. Ukuran default-nya adalah 16 MB. Mengurangi ukuran bagian minimum menyebabkan unggahan multipart dibagi menjadi lebih banyak bagian yang lebih kecil. Mengatur nilai ini terlalu rendah akan mengakibatkan efek negatif terhadap kecepatan transfer, menyebabkan latensi tambahan dan komunikasi jaringan untuk setiap bagian.
- NumberOfUploadThreads- Mendapat atau menetapkan jumlah thread eksekusi. Properti ini menentukan berapa banyak utas aktif yang akan digunakan untuk meng-unggah file. Nilai default-nya adalah 10 utas.

Untuk mengonfigurasi TransferUtility klien S3, buat objek konfigurasi, atur properti Anda, dan teruskan objek ke TransferUtility konstruktor Anda seperti:

```
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);
```

Mengunduh File

Untuk mengunduh file dari S3, panggil `Download` pada objek Transfer Utility, yang memberikan parameter berikut:

- `file` - Nama string dari file yang ingin Anda unduh
- `bucketName` - Nama string dari bucket S3 tempat Anda ingin mengunduh file

- `key` - Sebuah string yang mewakili nama objek S3 (dalam kasus ini, sebuah file) yang akan diunduh

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

Meng-unggah sebuah file

Untuk meng-unggah file ke S3, panggil `Upload` pada objek Transfer Utility, yang memberikan parameter berikut:

- `file` - Nama string dari file yang ingin Anda unggah
- `bucketName` - Nama string dari bucket S3 untuk menyimpan file

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

Kode di atas mengasumsikan bahwa ada file di direktori Environment. SpecialFolder. ApplicationData. Unggahan secara otomatis menggunakan fungsionalitas unggah multi-bagian S3 pada file large untuk meningkatkan throughput.

Menggunakan Service Level S3 APIs

Selain menggunakan S3 TransferUtility, Anda juga dapat berinteraksi dengan S3 menggunakan S3 tingkat rendah. APIs

Menginisialisasi Amazon S3 Client

Untuk menggunakan Amazon S3, pertama-tama kita perlu membuat instance Amazons3Client yang mengambil referensi ke instance Cognito AWSCredentials yang Anda buat sebelumnya dan wilayah Anda:

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

Mengunduh File

Untuk mengunduh file dari 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);
    }
}
```

Meng-unggah File

Untuk meng-unggah file ke 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);
```

Hapus Item

Untuk menghapus item di 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);
```

Menghapus Banyak Item

Untuk menghapus beberapa objek dari bucket menggunakan permintaan HTTP tunggal:

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

Anda dapat menentukan hingga 1000 kunci.

Buat Daftar Bucket

Berikan izin untuk mencantumkan semua bucket yang dimiliki oleh pengirim permintaan yang di-autentikasi:

```
// 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);
}
```

Daftar Objek

Anda dapat mengembalikan beberapa atau semua (hingga 1000) objek yang disimpan dalam bucket S3 Anda. Untuk melakukannya, Anda harus memiliki akses baca ke 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);
  }
}
```

Dapatkan Wilayah Bucket

Untuk mendapatkan wilayah tempat bucket berada:

```
// 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);
```

Dapatkan Kebijakan Bucket

Untuk mendapatkan kebijakan 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

Apa itu Amazon DynamoDB?

Amazon DynamoDB adalah layanan basis data yang cepat, sangat dapat diskalakan dan bukan relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi.

Konsep kunci

Konsep model data DynamoDB termasuk tabel, item, dan atribut.

Tabel

Di Amazon DynamoDB, basis data adalah kumpulan tabel. Sebuah tabel adalah kumpulan item, dan setiap item adalah kumpulan atribut.

Dalam basis data relasional, tabel memiliki skema yang telah ditetapkan seperti nama tabel, kunci utama, daftar nama kolom dan jenis data mereka. Semua catatan yang disimpan dalam tabel harus memiliki set kolom yang sama. Sebaliknya, DynamoDB hanya mengharuskan tabel memiliki kunci utama, tetapi tidak mengharuskan Anda untuk menentukan semua nama atribut dan jenis data di muka.

Untuk mem-pelajari selengkapnya tentang cara menggunakan tabel, lihat Cara Menggunakan Tabel di DynamoDB.

Item dan Atribut

Sebuah item dalam tabel DynamoDB dapat memiliki sejumlah atribut, meskipun ukuran item terbatas pada 400 KB. Ukuran item adalah jumlah dari panjang nama dan nilai atributnya (panjang biner dan UTF-8).

Setiap atribut dalam sebuah item adalah pasangan nama-nilai. Atribut dapat berupa set nilai-tunggal atau multi-nilai. Misalnya, item buku dapat memiliki atribut judul dan penulis. Setiap buku memiliki satu judul tetapi dapat memiliki banyak penulis. Atribut multinilai adalah satu set; nilai-nilai duplikat tidak diperbolehkan.

Misalnya, anggap Anda akan menyimpan katalog produk di DynamoDB. Anda dapat membuat tabel, ProductCatalog, dengan atribut Id sebagai kunci utama. Kunci utama tersebut secara unik

mengidentifikasi setiap item, sehingga tidak ada dua produk yang ada dalam tabel yang dapat memiliki ID yang sama.

Untuk mem-pelajari selengkapnya tentang cara menggunakan item, lihat Cara Menggunakan Item di DynamoDB.

Tipe Data

Amazon DynamoDB men-support tipe data berikut:

- Tipe skalar – Angka, String, Biner, Boolean, dan Null.
- Tipe multi-nilai – Set String, Set Angka, dan Set Biner.
- Tipe dokumen – Daftar dan Peta.

Untuk informasi lebih lanjut tentang Jenis Data skalar, Jenis Data Multi-Nilai, dan Jenis Data Dokumen, lihat Jenis data DynamoDB.

Kunci Utama

Ketika Anda membuat tabel, selain nama tabel, Anda harus menentukan kunci primer dari tabel. Kunci utama secara unik mengidentifikasi setiap item dalam tabel, sehingga tidak ada dua item dapat memiliki kunci yang sama. DynamoDB men-support dua jenis kunci utama berikut ini:

- Kunci Hash: Kunci utama terbuat dari satu atribut, atribut hash. DynamoDB membangun indeks hash tak berurutan pada atribut kunci primer ini. Setiap item dalam tabel diidentifikasi secara unik berdasarkan nilai kunci hash-nya.
- Kunci Hash dan Rentang: Kunci primer terbuat dari dua atribut. Atribut pertama adalah atribut hash dan yang kedua adalah atribut rentang. DynamoDB membangun indeks hash tak berurutan pada atribut kunci primer hash, dan indeks rentang diurutkan pada atribut kunci primer rentang. Setiap item dalam tabel diidentifikasi secara unik berdasarkan kombinasi nilai kunci hash dan rentang-nya. Hal ini dimungkinkan untuk dua item memiliki nilai kunci hash yang sama, tetapi kedua item tersebut harus memiliki nilai kunci rentang yang berbeda.

Indeks Sekunder

Bila Anda membuat tabel dengan kunci hash dan rentang, Anda secara opsional dapat menentukan satu atau lebih indeks sekunder pada tabel itu. Indeks sekunder memungkinkan Anda melakukan kueri data dalam tabel menggunakan kunci alternatif, selain kueri terhadap kunci primer.

DynamoDB men-support dua jenis indeks sekunder: indeks sekunder lokal dan indeks sekunder global.

- Indeks sekunder lokal: Indeks yang memiliki kunci hash yang sama seperti tabel, tetapi kunci rentang yang berbeda.
- Indeks sekunder global: Indeks yang meiliki kunci hash dan kunci rentang yang dapat berbeda dari yang ada pada tabel.

Anda dapat menentukan hingga 5 indeks sekunder global dan 5 indeks sekunder lokal untuk setiap tabel. Untuk informasi selengkapnya, lihat Meningkatkan Akses Data dengan Indeks Sekunder dalam DynamoDB dalam Panduan Developer DynamoDB.

Melakukan Kueri dan Pemindaian

Selain menggunakan kunci utama untuk mengakses item, Amazon DynamoDB juga menyediakan APIs dua untuk mencari data: Query dan Scan. Sebaiknya Anda membaca Panduan Kueri dan Pemindaian dalam Panduan Developer DynamoDB untuk membuat Anda terbiasa dengan beberapa praktik terbaik.

Kueri

Operasi Kueri mencari item dalam tabel atau indeks sekunder hanya menggunakan nilai atribut kunci primer. Anda harus memberikan nama atribut kunci hash dan nilai yang berbeda untuk mencari. Secara opsional, Anda dapat memberikan nama dan nilai atribut kunci rentang, dan menggunakan operator perbandingan untuk memperbaiki hasil pencarian.

Untuk kueri sampel, lihat:

- Menggunakan Model Dokumen
- Menggunakan Model Persistensi Objek
- Menggunakan Level Layanan DynamoDB APIs

Untuk informasi tentang Kueri, lihat Kueri di Panduan Developer DynamoDB.

Scan

Operasi Pemindaian membaca setiap item dalam tabel atau indeks sekunder. Secara default, operasi Pemindaian mengembalikan semua atribut data untuk setiap item dalam tabel atau indeks.

Anda dapat menggunakan ProjectionExpression parameter sehingga Scan hanya mengembalikan beberapa atribut, bukan semuanya.

Untuk pemindaian sampel, lihat:

- Menggunakan Model Dokumen
- Menggunakan Model Persistensi Objek
- Menggunakan Level Layanan DynamoDB APIs

Untuk informasi tentang Pemindaian, lihat Pemindaian di Panduan Developer DynamoDB.

Penyiapan Proyek

Prasyarat

Untuk menggunakan DynamoDB dalam aplikasi Anda, Anda harus menambahkan SDK ke dalam proyek Anda. Untuk melakukannya, ikuti petunjuk di Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Membuat Tabel DynamoDB

Untuk membuat tabel, buka Konsol DynamoDB dan ikuti langkah-langkah ini:

1. Klik Buat Tabel.
2. Masukkan nama tabel.
3. Pilih Hash sebagai tipe kunci primer.
4. Pilih satu jenis dan masukkan nilai untuk nama atribut hash. Klik Lanjutkan.
5. Pada halaman Tambahkan Indeks, jika Anda berencana untuk menggunakan indeks sekunder global, atur Jenis Indeks ke “Indeks Sekunder Global” dan di bawah Kunci Hash Indeks, masukkan nilai untuk indeks sekunder. Hal ini akan memungkinkan Anda untuk melakukan kueri dan pemindaian menggunakan indeks primer maupun indeks sekunder. Klik Tambahkan Indeks ke Tabel, dan kemudian klik Lanjutkan. Untuk melewatkan penggunaan indeks sekunder global, klik Lanjutkan.
6. Atur kapasitas baca dan tulis ke tingkat yang Anda inginkan. Untuk informasi lebih lanjut tentang konfigurasi kapasitas, lihat Throughput yang Disediakan di Amazon DynamoDB. Klik Lanjutkan.
7. Pada layar berikutnya, masukkan email notifikasi untuk membuat alarm throughput, jika diinginkan. Klik Lanjutkan.

8. Di halaman ringkasan, klik Buat. DynamoDB akan membuat basis data Anda.

Mengatur Izin untuk DynamoDB

Untuk menggunakan DynamoDB dalam aplikasi, Anda harus mengatur izin yang benar. Kebijakan IAM berikut memungkinkan pengguna untuk menghapus, mendapatkan, menempatkan, melakukan kueri, memindai, dan memperbarui item dalam tabel DynamoDB tertentu, yang diidentifikasi oleh 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"
    }
  ]
}
```

Anda dapat mengubah kebijakan di Konsol IAM. Anda harus menambahkan atau menghapus tindakan yang diizinkan berdasarkan kebutuhan aplikasi Anda.

Untuk mem-pelajari selengkapnya tentang kebijakan IAM, lihat Menggunakan IAM.

Untuk mem-pelajari selengkapnya tentang kebijakan khusus DynamoDB, lihat Menggunakan IAM untuk Mengendalikan Akses ke Sumber Daya DynamoDB dalam Panduan Developer DynamoDB.

Mengintegrasikan DynamoDB dengan Aplikasi Anda

AWS Mobile SDK for .NET and Xamarin menyediakan perpustakaan tingkat tinggi untuk digunakan dengan DynamoDB. Anda juga dapat membuat permintaan langsung terhadap API DynamoDB tingkat rendah, tetapi untuk sebagian besar kasus penggunaan, perpustakaan tingkat tinggi lebih direkomendasikan. AmazonDynamoDBClient Ini adalah bagian yang sangat berguna dari

perpustakaan tingkat tinggi. Dengan menggunakan kelas ini, Anda dapat melakukan berbagai operasi buat, baca, perbarui, dan hapus (CRUD) dan menjalankan kueri.

AWS Mobile SDK untuk.NET dan Xamarin memungkinkan Anda APIs melakukan panggilan menggunakan AWS SDK for .NET untuk bekerja dengan DynamoDB. Semua APIs tersedia di AWSSDK .dll. Untuk informasi tentang mengunduh AWS SDK for .NET, lihat AWS SDK for .NET.

Ada tiga cara yang dapat Anda gunakan untuk berinteraksi dengan DynamoDB dalam aplikasi Xamarin Anda:

- Model Dokumen: API ini menyediakan kelas pembungkus pada API tingkat rendah DynamoDB untuk lebih menyederhanakan tugas-tugas pemrograman Anda. Tabel dan Dokumen adalah kelas pembungkus kunci. Anda dapat menggunakan model dokumen untuk operasi data seperti membuat, mengambil, memperbarui dan menghapus item. API tersedia di Amazon.DynamoDB. DocumentModel namespace.
- Model Persistensi Objek: API Persistensi Objek memungkinkan Anda memetakan kelas sisi klien ke tabel DynamoDB. Kemudian, setiap instans objek memetakan ke item dalam tabel yang sesuai. DBContext Kelas Dynamo dalam API ini menyediakan metode bagi Anda untuk menyimpan objek sisi klien ke tabel, mengambil item sebagai objek dan melakukan kueri dan pemindaian. Anda dapat menggunakan model Persitensi Objek untuk operasi data seperti membuat, mengambil, memperbarui dan menghapus item. Anda harus terlebih dahulu membuat tabel Anda menggunakan API Klien Layanan dan kemudian menggunakan model persistensi objek untuk memetakan kelas Anda ke tabel. API tersedia di Amazon.DynamoDB. DataModel namespace.
- API Klien Layanan: Ini adalah API tingkat protokol yang memetakan DynamoDB API dengan erat. Anda dapat menggunakan API tingkat rendah ini untuk semua operasi tabel dan item seperti membuat, memperbarui, atau menghapus tabel dan item. Anda juga dapat melakukan kueri dan memindai tabel Anda. API ini tersedia di namespace Amazon.DynamoDB.

Ketiga model ini diuraikan secara mendalam dalam topik berikut:

Menggunakan Model Dokumen

Model Dokumen menyediakan kelas pembungkus pada .NET API tingkat rendah. Tabel dan Dokumen adalah kelas pembungkus kunci. Anda dapat menggunakan Model Dokumen untuk membuat, mengambil, memperbarui dan menghapus item. Untuk membuat, memperbarui, dan menghapus tabel, Anda harus menggunakan API tingkat rendah. Untuk petunjuk tentang cara

menggunakan API tingkat rendah, lihat Menggunakan Level Layanan DynamoDB. APIs API tingkat rendah tersedia di Amazon.DynamoDB. DocumentModel namespace.

Untuk mem-pelajari selengkapnya tentang Model Dokumen, lihat Model Dokumen .NET.

Membuat Klien DynamoDB

Untuk membuat klien DynamoDB:

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

Operasi CRUD

Simpan Item

Buat item:

```
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";
```

Simpan item ke tabel DynamoDB:

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

Ambil Item

Untuk mengambil item:

```
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);
}
```

Perbarui Item

Untuk memperbarui item:

```
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);
}
```

Untuk memperbarui item secara bersyarat:

```
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);
}
```

Hapus Item

Untuk menghapus item:

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

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

Melakukan Kueri dan Pemindaian

Untuk melakukan kueri dan mengambil semua buku yang penulisnya adalah “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);
  });
}
```

Memindai contoh kode di bawah ini akan mengembalikan semua buku dalam tabel kami:

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

Menggunakan Model Persistensi Objek

AWS Mobile SDK for .NET and Xamarin menyediakan model Persistensi Objek yang memungkinkan Anda memetakan kelas sisi klien Anda ke tabel DynamoDB. Setiap instans objek memetakan ke item dalam tabel yang sesuai. Untuk menyimpan objek sisi klien Anda ke tabel, model Object Persistence

menyediakan DBContext kelas Dynamo, titik masuk ke DynamoDB. Kelas ini menyediakan koneksi ke DynamoDB dan memungkinkan Anda mengakses tabel, melakukan berbagai operasi CRUD, dan menjalankan kueri.

Model Persistensi Objek tidak menyediakan API untuk membuat, memperbarui, atau menghapus tabel. Model ini hanya menyediakan operasi data. Untuk membuat, memperbarui, dan menghapus tabel, Anda harus menggunakan API tingkat rendah. Untuk petunjuk tentang cara menggunakan API tingkat rendah, lihat Menggunakan Level Layanan DynamoDB. APIs

Gambaran Umum

Model Persistensi Objek menyediakan sekumpulan atribut untuk memetakan kelas sisi klien ke tabel, dan properti/bidang ke atribut tabel. Model Persistensi Objek men-support pemetaan eksplisit dan default antara properti kelas dan atribut tabel.

- Pemetaan eksplisit: Untuk memetakan properti ke kunci utama, Anda harus menggunakan atribut model Dynamo DBHash Key dan Dynamo DBRange Key Object Persistence. Selain itu, untuk atribut kunci non-primer, jika nama properti di kelas Anda dan atribut tabel yang sesuai yang ingin Anda petakan tidak sama, maka Anda harus menentukan pemetaan dengan secara eksplisit menambahkan atribut Dynamo. DBProperty
- Pemetaan default - Secara default, model Persistensi Objek memetakan properti kelas ke atribut dengan nama yang sama dalam tabel.

Anda tidak perlu memetakan setiap properti kelas tunggal. Anda mengidentifikasi properti ini dengan menambahkan DBIgnore atribut Dynamo. Menyimpan dan mengambil sebuah instans dari sebuah objek akan menghilangkan properti yang ditandai dengan atribut ini.

Tipe Data yang Didukung

Model Persistensi Objek mendukung sekumpulan jenis data .NET primitif, koleksi, dan jenis data arbitrer. Model ini mendukung jenis data primitif berikut.

- bool
- byte
- char
- DateTime
- desimal, ganda, mengambang

- Int16, Int32, Int64
- SByte
- string
- UInt16, UInt32, UInt64

Model Persistensi Objek juga mendukung jenis kumpulan .NET dengan keterbatasan sebagai berikut:

- Jenis koleksi harus mengimplementasikan ICollection antarmuka.
- Jenis kumpulan harus terdiri dari tipe primitif yang didukung. Misalnya, ICollection<string>, ICollection<bool>.
- Jenis kumpulan harus menyediakan konstruktor nir-parameter.

Untuk informasi lebih lanjut tentang model Persistensi Objek, lihat Model Persistensi Objek .NET.

Membuat Klien DynamoDB

Untuk membuat klien DynamoDB:

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

Operasi CRUD

Menyimpan Objek

Membuat objek:

```
[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
};
```

Menyimpan objek ke tabel DynamoDB:

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

Mengambil Objek

Untuk mengambil sebuah objek:

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

Memperbarui Obyek

Untuk memperbarui objek:

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

Menghapus Objek

Untuk menghapus objek:

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

Melakukan Kueri dan Pemindaian

Untuk melakukan kueri dan mengambil semua buku yang penulisnya adalah “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());
  });
}
```

Memindai contoh kode di bawah ini akan mengembalikan semua buku dalam tabel kami:

```
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());
  });
}
```

Menggunakan Level Layanan DynamoDB APIs

Tingkat Layanan Dynamo APIs memungkinkan Anda membuat, memperbarui, dan menghapus tabel. Anda dapat menggunakan API tingkat rendah untuk melakukan operasi pembuatan, pembacaan, pembaruan, dan penghapusan (CRUD) biasanya pada suatu item dalam tabel.

Membuat Klien DynamoDB

Untuk membuat klien DynamoDB:

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

Operasi CRUD

Simpan Item

Untuk menyimpan item ke tabel 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);
```

Ambil Item

Untuk mengambil item:

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

Perbarui Item

Untuk memperbarui item:

```
// 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);
```

Menghapus Item

Untuk menghapus item:

```
// 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);
```

Melakukan Kueri dan Pemindaian

Untuk melakukan kueri dan mengambil semua buku yang penulisnya adalah “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);
    });

  }
}
```

Memindai contoh kode di bawah ini akan mengembalikan semua buku dalam tabel kami:

```
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)

Dengan menggunakan SNS dan AWS Mobile SDK for .NET and Xamarin, Anda dapat tulis aplikasi yang dapat menerima notifikasi push seluler. Untuk informasi tentang SNS, lihat Amazon Simple Notification Service.

Konsep kunci

Amazon SNS memungkinkan aplikasi dan pengguna akhir pada perangkat yang berbeda untuk menerima notifikasi melalui notifikasi Mobile Push (Apple, Google, dan Kindle Fire Devices), HTTP/HTTPS, Email/Email -JSON, SMS atau Amazon Simple Queue Service (SQS) antrian, atau fungsi AWS Lambda. SNS memungkinkan Anda mengirim pesan individual atau pesan fan-out ke sejumlah besar penerima yang berlangganan pada satu topik.

Topik

Topik adalah “titik akses” untuk mengizinkan penerima berlangganan secara dinamik untuk salinan identik dari notifikasi yang sama. Satu topik dapat men-support pengiriman ke beberapa jenis titik akhir – misalnya, Anda dapat mengelompokkan bersama penerima iOS, Android, dan SMS dalam grup.

Langganan

Untuk menerima pesan yang dipublikasikan ke topik, Anda harus berlangganan titik akhir untuk topik tersebut. Titik akhir adalah aplikasi seluler, web server, alamat email, atau antrean Amazon SQS yang dapat menerima pesan notifikasi dari Amazon SNS. Setelah Anda berlangganan titik akhir untuk topik dan langganan itu dikonfirmasi, titik akhir akan menerima semua pesan yang dipublikasikan ke topik tersebut.

Publikasi

Ketika Anda memublikasikan ke topik, SNS akan memberikan salinan yang diformat dengan tepat pesan Anda untuk setiap pelanggan topik tersebut. Untuk Notifikasi Push Mobile, Anda dapat memublikasikan langsung ke titik akhir atau berlangganan titik akhir untuk topik.

Penyiapan Proyek

Prasyarat

Untuk menggunakan SNS dalam aplikasi Anda, Anda harus menambahkan SDK untuk proyek Anda. Untuk melakukannya, ikuti petunjuk di Menyiapkan AWS Mobile SDK for .NET and Xamarin.

Mengatur Izin untuk SNS

Untuk informasi tentang pengaturan izin untuk SNS, lihat Mengelola Akses ke Topik Amazon SNS Anda.

Tambahkan NuGet Package untuk SNS ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket NuGet Amazon Simple Notification Service ke project Anda.

Mengintegrasikan SNS dengan Aplikasi Anda

Ada banyak cara untuk berinteraksi dengan SNS dalam aplikasi Xamarin anda:

Mengirim Notifikasi Push (Xamarin Android)

Dokumen ini menjelaskan cara mengirim notifikasi push ke aplikasi Xamarin Android menggunakan Amazon Simple Notification Service (SNS) dan AWS Mobile SDK for .NET and Xamarin.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Mengatur Izin untuk SNS

Ikuti Langkah 2 yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin untuk melampirkan kebijakan yang disebutkan di bawah ini ke peran aplikasi Anda. Hal ini akan memberikan aplikasi Anda izin yang tepat untuk mengakses SNS:

1. Buka Konsol IAM dan pilih IAM role yang akan dikonfigurasi.

2. Klik Lampirkan Kebijakan, pilih kebijakan SNSFull Akses Amazon dan klik Lampirkan Kebijakan.

> **Warning**
>
> Menggunakan Amazon SNSFull Access tidak disarankan di lingkungan produksi. Kami menggunakannya di sini untuk memungkinkan Anda untuk bangun dan berjalan dengan cepat. Untuk informasi selengkapnya tentang bagaimana menentukan izin untuk IAM role, lihat Gambaran Umum Izin IAM Role.

Aktifkan Notifikasi Push pada Google Cloud

Pertama, tambahkan proyek Google API baru:

1. Buka Konsol Developer Google.
2. Klik Buat Proyek.
3. Di kotak Proyek Baru, masukkan nama proyek, catat ID proyek (Anda akan membutuhkannya nanti) dan klik Buat.

Selanjutnya, aktifkan layanan Google Cloud Messaging (GCM) untuk proyek Anda:

1. Di Konsol Developer Google, proyek baru Anda seharusnya sudah dipilih. Jika tidak, pilih di drop-down yang ada di bagian atas halaman.
2. Pilih APIs & auth dari bilah samping di sisi kiri halaman.
3. Di kotak pencarian, ketik “Google Cloud Messaging untuk Android” dan klik tautan Google Cloud Messaging untuk Android.
4. Klik Aktifkan API.

Akhirnya, dapatkan Kunci API:

1. Di Google Developers Console, pilih APIs & auth > Credentials.
2. Di bawah Akses API Publik, klik Buat kunci baru.
3. Pada dialog Buat kunci baru, klik Kunci server.
4. Dalam dialog yang dihasilkan, klik Buat dan salin kunci API yang ditampilkan. Anda akan menggunakan kunci API ini untuk melakukan autentikasi nanti.

Menggunakan ID Proyek untuk Membuat ARN Platform di Konsol SNS

1. Buka Konsol SNS.
2. Klik Aplikasi di sisi kiri layar.
3. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
4. Masukkan Nama Aplikasi.
5. Pilih Google Cloud Messaging (GCM) untuk Platform notifikasi push.
6. Tempelkan kunci API ke dalam kotak teks berlabel Kunci API.
7. Klik Buat aplikasi platform.
8. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya.

Tambahkan NuGet Package untuk SNS ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket NuGet Amazon Simple Notification Service ke project Anda.

Buat klien SNS

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

Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh

Untuk mendaftar notifikasi jarak jauh di Android, Anda harus membuat BroadcastReceiver yang dapat menerima pesan Google Cloud. Ubah nama paket di bawah ini saat diminta untuk melakukannya:

```
[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);
      }
}
```

Di bawah ini adalah layanan yang menerima pemberitahuan push dari BroadcastReceiver dan menampilkan notifikasi di bilah notifikasi perangkat:

```
[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());
  }
}
```

Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda

1. Buka Konsol SNS > Aplikasi.
2. Pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir.
3. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

Mengirim Notifikasi Push (Xamarin iOS)

Dokumen ini menjelaskan cara mengirim notifikasi push ke aplikasi Xamarin iOS menggunakan Amazon Simple Notification Service (SNS) dan AWS Mobile SDK for .NET and Xamarin.

Penyiapan Proyek

Prasyarat

Anda harus menyelesaikan semua instruksi yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin sebelum memulai tutorial ini.

Mengatur Izin untuk SNS

Ikuti Langkah 2 yang ada di Menyiapkan AWS Mobile SDK for .NET and Xamarin untuk melampirkan kebijakan yang disebutkan di bawah ini ke peran aplikasi Anda. Hal ini akan memberikan aplikasi Anda izin yang tepat untuk mengakses SNS:

1. Buka Konsol IAM dan pilih IAM role yang akan dikonfigurasi.
2. Klik Lampirkan Kebijakan, pilih kebijakan SNSFull Akses Amazon dan klik Lampirkan Kebijakan.

> **Warning**
>
> Menggunakan Amazon SNSFull Access tidak disarankan di lingkungan produksi. Kami menggunakannya di sini untuk memungkinkan Anda untuk bangun dan berjalan dengan cepat. Untuk informasi selengkapnya tentang bagaimana menentukan izin untuk IAM role, lihat Gambaran Umum Izin IAM Role.

Dapatkan Keanggotaan dalam Program Developer iOS Apple

Anda akan perlu menjalankan aplikasi Anda pada perangkat fisik untuk menerima notifikasi push. Untuk menjalankan aplikasi Anda di perangkat, Anda harus memiliki keanggotaan di Keanggotaan Program Developer iOS Apple. Setelah Anda memiliki keanggotaan, Anda dapat menggunakan Xcode untuk membuat identitas penandatanganan. Untuk informasi selengkapnya, lihat dokumentasi Quick Start Distribusi Aplikasi.

Membuat Sertifikat iOS

Pertama, Anda harus membuat Sertifikat iOS. Kemudian, Anda harus membuat profil penyediaan yang dikonfigurasi untuk notifikasi push. Untuk melakukannya:

1. Buka Pusat Anggota Developer Apple, klik Sertifikat, Pengidentifikasi & Profil.

2. Klik Pengidentifikasi di bawah Aplikasi iOS, lalu klik tombol plus yang ada di sudut kanan atas halaman web untuk menambahkan ID Aplikasi iOS baru, lalu masukkan deskripsi ID Aplikasi.
3. Gulir ke bawah hingga ke bagian Tambah Akhiran ID dan pilih ID Aplikasi Eksplisit dan masukkan pengidentifikasi paket Anda.
4. Gulir ke bawah hingga ke bagian Layanan Aplikasi dan pilih bagian Notifikasi push.
5. Klik Lanjutkan.
6. Klik Kirim.
7. Klik Selesai.
8. Pilih ID Aplikasi yang baru saja dibuat dan kemudian klik Edit.
9. Gulir ke bawah hingga ke bagian Notifikasi Push. Klik Buat Sertifikat di bawah Sertifikat SSL Pengembangan.
10. Ikuti petunjuk untuk membuat Permintaan Penandatanganan Sertifikat (CSR), unggah permintaan, dan unduh sertifikat SSL yang akan digunakan untuk berkomunikasi dengan Layanan Notifikasi Apple (APNS).
11. Kembali ke halaman Sertifikat, Pengidentifikasi & Profil. Klik Semua di bawah Profil Penyediaan.
12. Klik tombol plus yang ada di sudut kanan atas untuk menambahkan profil penyediaan baru.
13. Pilih Aplikasi iOS, dan kemudian klik Lanjutkan.
14. Pilih ID Aplikasi Anda, dan kemudian klik Lanjutkan.
15. Pilih sertifikat developer Anda, dan kemudian klik Lanjutkan.
16. Pilih perangkat Anda, dan kemudian klik Lanjutkan.
17. Masukkan nama profil, dan kemudian klik Buat.
18. Unduh dan klik dua kali file penyediaan untuk memasang profil penyediaan.

Untuk informasi lebih lanjut tentang penyediaan profil yang dikonfigurasi untuk notifikasi push, lihat dokumentasi Mengonfigurasi Notifikasi Push.

Menggunakan Sertifikat untuk Membuat ARN Platform di Konsol SNS

1. Jalankan aplikasi KeyChain akses, pilih Sertifikat Saya di sisi kiri bawah layar, lalu klik kanan sertifikat SSL yang Anda buat untuk terhubung ke APNS dan pilih Ekspor. Anda akan diminta untuk menentukan nama untuk file dan kata sandi untuk melindungi sertifikat. Sertifikat tersebut akan disimpan dalam file P12.
2. Buka Konsol SNS dan klik Aplikasi yang ada di sisi kiri layar.

3. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
4. Masukkan Nama Aplikasi.
5. Pilih Pengembangan Apple untuk Platform notifikasi push.
6. Klik Pilih File dan pilih file P12 yang Anda buat saat Anda mengekspor sertifikat SSL.
7. Masukkan kata sandi yang Anda tentukan saat Anda mengekspor sertifikat SSL dan klik Memuat Kredensial dari File.
8. Klik Buat aplikasi platform.
9. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya. Anda akan membutuhkannya dalam langkah-langkah mendatang.

Tambahkan NuGet Package untuk SNS ke Proyek Anda

Ikuti Langkah 4 petunjuk dalam Menyiapkan AWS Mobile SDK untuk.NET dan Xamarin untuk menambahkan paket NuGet Amazon Simple Notification Service ke project Anda.

Buat Klien SNS

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

Daftarkan Aplikasi Anda untuk Notifikasi Jarak Jauh

Untuk mendaftarkan aplikasi, hubungi RegisterForRemoteNotifications UIApplication objek Anda, seperti yang ditunjukkan di bawah ini. Tempatkan kode berikut AppDelegate di.cs, masukkan ARN aplikasi platform Anda di mana diminta di bawah ini:

```
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 */
    });
  }
}
```

Mengirim Pesan dari Konsol SNS untuk Titik Akhir Anda

1. Buka Konsol SNS > Aplikasi.
2. Pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir.
3. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

Mengirim dan Menerima Notifikasi SMS

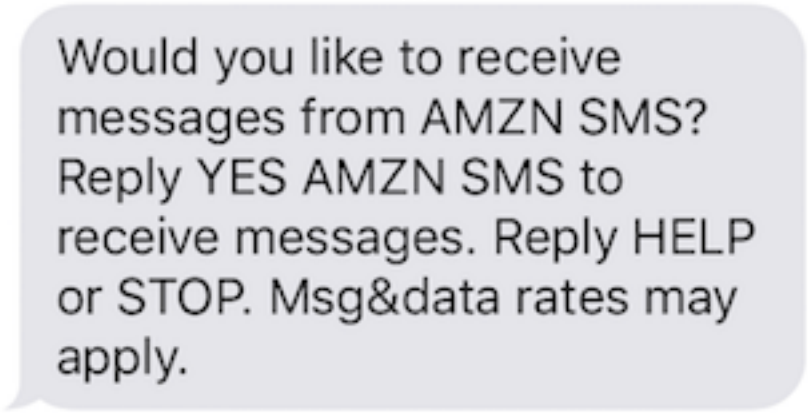
Anda dapat menggunakan Amazon Simple Notification Service (Amazon SNS) untuk mengirim dan menerima notifikasi Short Message Service (SMS) ke ponsel dan ponsel pintar yang mengaktifkan SMS.

> **Note**
>
> Notifikasi SMS saat ini didukung untuk nomor telepon di Amerika Serikat. Pesan SMS hanya dapat dikirim dari topik yang dibuat di wilayah US East (N. Virginia). Namun, Anda dapat memublikasikan pesan ke topik yang Anda buat di wilayah US East (N. Virginia) dari setiap wilayah lainnya.

Membuat Topik

Untuk membuat topik:

1. Di konsol Amazon SNS, klik Buat topik baru. Kotak dialog Buat topik baru akan muncul.
2. Dalam kotak Nama topik, ketik nama topik.
3. Dalam kotak Nama tampilan, ketik nama tampilan. Topik harus memiliki nama tampilan yang ditetapkan untuknya karena sepuluh (10) karakter pertama dari nama tampilan tersebut akan digunakan sebagai bagian awal dari prefiks pesan teks. Nama tampilan yang Anda masukkan akan muncul dalam pesan konfirmasi yang dikirimkan SNS ke pengguna (nama tampilan di bawah ini adalah “AMZN SMS”).



1. Klik Buat topik. Topik baru akan muncul di halaman Topik.
2. Pilih topik baru dan kemudian klik ARN topik. Halaman Detail Topik akan muncul.
3. Salin ARN topik, karena Anda akan membutuhkannya ketika Anda berlangganan topik di langkah berikutnya.

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

Berlangganan Topik Menggunakan Protokol SMS

Membuat klien SNS, memberikan objek kredensial Anda dan wilayah kolam identitas Anda:

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

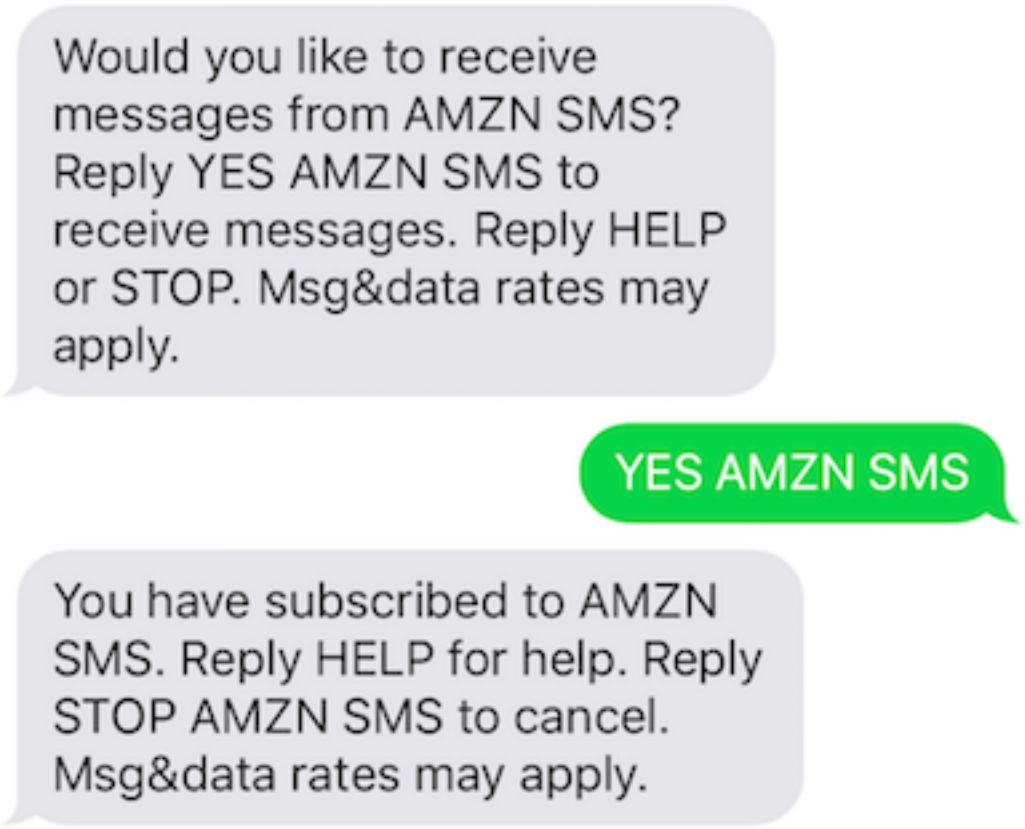
Untuk berlangganan topik, aktifkan `SubscribeAsync` dan berikan ARN topik yang ingin Anda langgani kepadanya, protokol (“sms”), dan nomor telepon:

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

Anda akan menerima arn berlangganan di objek respon berlangganan. Langganan arn Anda terlihat seperti ini:

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

Ketika perangkat berlangganan topik, SNS akan mengirim pesan konfirmasi ke perangkat, dan pengguna harus mengonfirmasi bahwa mereka ingin menerima notifikasi, seperti yang ditunjukkan di bawah ini:



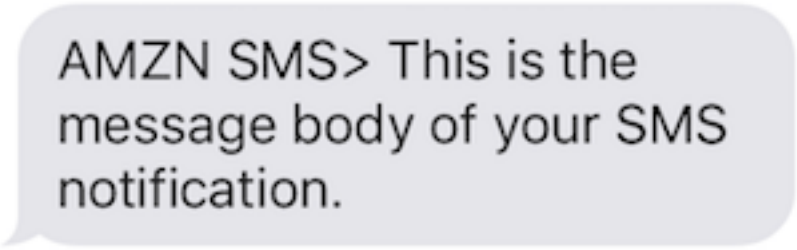
Setelah pengguna berlangganan topik, mereka akan menerima pesan SMS ketika Anda memublikasikannya ke topik tersebut.

Memublikasikan Pesan

Untuk memublikasikan pesan ke topik:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon SNS.

2. Di panel navigasi kiri, klik Topik lalu pilih topik yang ingin Anda publikasikan.
3. Klik Publikasikan topik.
4. Dalam kotak Subjek, ketik subjek.
5. Dalam kotak Pesan, ketik pesan. Amazon SNS mengirimkan teks yang Anda masukkan dalam kotak pesan untuk pelanggan SMS kecuali Anda juga memasukkan teks ke dalam kotak Subjek. Karena Amazon SNS menyertakan prefiks nama tampilan dengan semua pesan SMS yang Anda kirim, jumlah prefiks nama tampilan dan muatan pesan tidak melebihi 140 karakter ASCII atau 70 karakter Unicode. Amazon SNS memotong pesan yang melebihi batas ini.
6. Klik Publikasikan pesan. Amazon SNS menampilkan kotak dialog konfirmasi. Pesan SMS muncul di perangkat Anda yang mengaktifkan SMS, seperti yang ditunjukkan di bawah ini.



Mengirim Pesan ke Titik Akhir HTTP/HTTPS

Anda dapat menggunakan Amazon SNS untuk mengirimkan pesan notifikasi ke satu atau lebih titik akhir HTTP atau HTTPS. Prosesnya adalah sebagai berikut:

1. Mengonfigurasi titik akhir Anda untuk menerima pesan Amazon SNS.
2. Melanggankan titik akhir HTTP/HTTPS ke topik.
3. Mengonfirmasi langganan Anda.
4. Mempublikasikan notifikasi ke topik. Amazon SNS kemudian akan mengirimkan permintaan HTTP POST yang menyampaikan isi notifikasi ke titik akhir yang berlangganan.

Mengonfigurasi Titik Akhir HTTP/HTTPS Anda untuk Menerima Pesan Amazon SNS

Ikuti instruksi di Langkah 1 Mengirim Pesan Amazon SNS ke Titik Akhir HTTP/HTTPS untuk mengonfigurasi titik akhir Anda.

Melangganken Titik Akhir HTTP/HTTPS Anda ke Topik Amazon SNS Anda

Membuat klien SNS, memberikan objek kredensial Anda dan wilayah kolam identitas Anda:

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

Untuk mengirimkan pesan ke titik akhir HTTP atau HTTPS melalui topik, Anda harus melangganken titik akhir ke topik Amazon SNS. Anda menentukan titik akhir menggunakan URL-nya:

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

Mengonfirmasi Langganan Anda

Setelah Anda berlangganan titik akhir, Amazon SNS akan mengirimkan pesan konfirmasi berlangganan ke titik akhir. Kode pada titik akhir harus mengambil nilai `SubscribeURL` dari pesan konfirmasi berlangganan dan mengunjungi lokasi yang ditentukan oleh `SubscribeURL` sendiri atau membuatnya tersedia untuk Anda sehingga Anda dapat secara manual mengunjungi `SubscribeURL` (misalnya menggunakan peramban web).

Amazon SNS tidak akan mengirimkan pesan ke titik akhir sampai langganan dikonfirmasi. Saat Anda mengunjungi `SubscribeURL`, respons akan berisi dokumen XML yang berisi elemen `SubscriptionArn` yang menentukan ARN untuk berlangganan.

Kirim Pesan ke Titik Akhir HTTP/HTTPS

Anda dapat mengirimkan pesan ke langganan topik dengan memublikasikan ke topik. Aktifkan `PublishAsync` dan berikan topik ARN dan pesan Anda padanya.

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

Memecahkan masalah SNS

Menggunakan Status Pengiriman di Konsol Amazon SNS

Konsol Amazon SNS berisi fitur Status Pengiriman yang memungkinkan Anda mengumpulkan umpan balik tentang upaya pengiriman pesan Anda yang berhasil dan tidak berhasil ke platform notifikasi push seluler (Apple (APNS), Google (GCM), Amazon (ADM), Windows (WNS dan MPNS) dan Baidu.

Hal ini juga memberikan informasi penting lainnya seperti waktu tinggal di Amazon SNS. Informasi ini ditangkap dalam grup CloudWatch Log Amazon yang dibuat secara otomatis oleh Amazon SNS saat fitur ini diaktifkan melalui konsol Amazon SNS atau melalui Amazon SNS. APIs

Untuk petunjuk tentang cara menggunakan fitur Status Pengiriman, lihat Menggunakan fitur Status Pengiriman Amazon SNS pada AWS Mobile Blog.

Praktik Terbaik untuk Menggunakan AWS Mobile SDK for .NET and Xamarin

Hanya ada beberapa hal mendasar dan praktik terbaik yang membantu untuk mengetahui kapan menggunakan AWS Mobile SDK for .NET and Xamarin.

- Menggunakan Amazon Cognito untuk mendapatkan kredensial AWS alih-alih melakukan hard-coding pada kredensial Anda dalam aplikasi Anda. Jika Anda melakukan hard-coding pada kredensial Anda dalam aplikasi Anda, maka Anda mungkin harus mengekspos kredensial tersebut ke publik, yang memungkinkan orang lain untuk membuat panggilan ke AWS menggunakan kredensial Anda. Untuk petunjuk tentang cara menggunakan Amazon Cognito untuk mendapatkan kredensial AWS, lihat Menyiapkan AWS Mobile SDK for .NET and Xamarin.
- Untuk praktik terbaik menggunakan S3, lihat artikel ini di AWS Blog.
- Untuk praktik terbaik menggunakan DynamoDB, lihat Praktik Terbaik DynamoDB dalam Panduan Developer DynamoDB.

Kami selalu mencari untuk membantu pelanggan kami menjadi sukses dan menyambut umpan balik, jadi jangan ragu untuk memposting di forum AWS atau membuka masalah di GitHub.

Perpustakaan Dokumentasi Layanan AWS

Setiap layanan di AWS Mobile SDK for .NET and Xamarin memiliki panduan developer terpisah dan referensi layanan API yang menyediakan informasi tambahan yang mungkin berguna bagi Anda.

Identitas Amazon Cognito

- Panduan Pengembang Cognito
- Referensi API Layanan Identitas Cognito

Sinkronisasi Amazon Cognito

- Panduan Pengembang Cognito
- Referensi API Layanan Sinkronisasi Cognito

Amazon Mobile Analytics

- Panduan Pengembang Mobile Analytics
- Referensi API Layanan Mobile Analytics

Amazon S3

- Panduan Pengembang S3
- Panduan Memulai S3
- Referensi API Layanan S3

Amazon DynamoDB

- Panduan Pengembang DynamoDB
- Panduan Memulai DynamoDB
- Referensi API Layanan DynamoDB

Amazon Simply Notification Service (SNS)

- Panduan Pengembang SNS
- Referensi API Layanan SNS

Tautan Bermanfaat Lainnya

- Daftar Istilah AWS
- Tentang AWS Credentials

Pemecahan masalah

Topik ini menjelaskan beberapa gagasan untuk memecahkan masalah yang mungkin Anda alami saat menggunakan AWS Mobile SDK for .NET and Xamarin.

Pastikan IAM role Memiliki Izin yang Diperlukan

Ketika memanggil layanan AWS, pastikan aplikasi Anda sedang menggunakan identitas dari kolam identitas Cognito. Setiap identitas dalam kolam tersebut di-associate dengan peran IAM (Identity and Access Management).

Peran memiliki satu file kebijakan atau lebih yang terkait dengannya yang menentukan sumber daya AWS yang ditetapkan pengguna untuk peran yang memiliki akses. Secara default dua peran dibuat per kolam identitas, satu untuk pengguna yang diautentikasi, dan satu untuk pengguna yang tidak diautentikasi.

Anda perlu memodifikasi file kebijakan yang ada atau meng-associate file kebijakan baru dengan izin yang diperlukan oleh aplikasi Anda. Jika aplikasi Anda memungkinkan pengguna baik yang di-autentikasi maupun yang tidak di-autentikasi, kedua peran harus diberikan izin untuk mengakses sumber daya AWS yang dibutuhkan aplikasi Anda.

File kebijakan berikut menunjukkan cara memberikan akses ke bucket S3:

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

File kebijakan berikut menunjukkan cara untuk memberikan akses ke basis data 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"
      }
   ]
}
```

Untuk informasi lebih lanjut tentang menentukan kebijakan, lihat Kebijakan IAM.

Menggunakan HTTP Proxy Debugger

Jika layanan AWS yang dipanggil aplikasi Anda memiliki titik akhir HTTP atau HTTPS, maka Anda dapat menggunakan HTTP/HTTPS proxy debugger untuk melihat permintaan dan respon untuk mendapatkan wawasan lebih banyak tentang apa yang terjadi. Ada sejumlah HTTP proxy debugger yang tersedia seperti:

- Charles - sebuah web debugging proxy untuk Windows dan OSX
- Fiddler - sebuah web debugging proxyfidd untuk Windows

Charles dan Fiddler, keduanya memerlukan beberapa konfigurasi untuk dapat melihat lalu lintas SSL terenkripsi, silakan baca dokumentasi untuk alat-alat ini untuk mendapatkan informasi lebih lanjut. Jika Anda menggunakan web debugging proxy yang tidak dapat dikonfigurasi untuk menampilkan lalu lintas terenkripsi, buka file aws_endpoints_json dan atur tag HTTP untuk layanan AWS yang harus Anda debug menjadi BETUL.

Riwayat Dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS Mobile SDK for .NET and Xamarin.

- Versi API: 2015-08-27
- Pembaruan Dokumentasi Terakhir: 2021-02-23

Perubahan	Versi API	Deskripsi	Tanggal rilis
Diarsipkan	2015-08-27	AWS Mobile SDK untuk Xamarin termasuk dalam. AWS SDK untuk .NET Panduan ini menjadi referensi versi yang diarsipkan dari Mobile SDK for Xamarin.	2021-02-23
Rilis Umumnya Tersedia (GA)	2015-08-27	Rilis Umumnya Tersedia (GA)	2015-08-27
Rilis beta	2015-07-28	Rilis beta	rn2015-07-28