
Amazon Mobile Analytics

User Guide



Amazon Mobile Analytics: User Guide

Copyright © 2015 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, AWS CloudTrail, AWS CodeDeploy, Amazon Cognito, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Amazon Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC, and Amazon WorkDocs. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

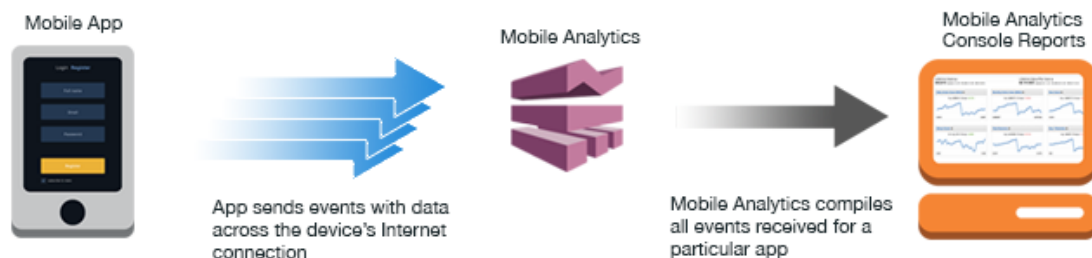
Table of Contents

What Is Amazon Mobile Analytics?	1
Incorporating Mobile Analytics	2
Using the Mobile SDK	2
Getting Started	3
Step 1: Sign Up	3
Step 2: Follow the Steps to Add an App	3
Step 3: Integrate Mobile Analytics in to Your App	4
Step 4: Review the Data in Amazon Mobile Analytics Reports	4
Adding and Managing Apps	5
Accessing Mobile Analytics	5
Adding an App	5
If You Have No Existing Identity Pools	5
Adding Additional Apps with the Default Cognito Identity Pool	8
If You Have Existing Identity Pools but No Default Cognito Identity Pool	10
If You Have Existing Identity Pools and a Default Cognito Identity Pool	13
Renaming an App	15
Viewing Integration Steps	16
Using the Mobile SDK	18
Platforms Supported by the Mobile SDK	18
Identifying the App	19
Managing Sessions	19
Managing Sessions in iOS or Android Apps	20
Managing Sessions in JavaScript Apps	20
Managing Sessions in Unity or Xamarin Apps	21
Generating Events	21
Creating a Monetization Event	22
Creating a Custom Event	24
Adding Attributes and Metrics	26
Reporting Detailed Data Points	26
Adding Attributes and Metrics in iOS or Android Apps	26
Adding Attributes and Metrics in JavaScript Apps	27
Adding Attributes and Metrics in Unity or Xamarin Apps	27
Submitting Events	28
Submitting Events in iOS Apps	28
Submitting Events in Android Apps	28
Submitting Events in JavaScript Apps	29
Submitting Events in Unity or Xamarin Apps	29
Using the REST API	30
Identifying the App in Mobile Analytics	30
Tying Analytics to Users	30
Managing Sessions	31
Users Engage in Sessions	31
Starting a Session	31
Stopping a Session	33
Generating Events	34
Types of Analytics Events	34
Session Events	34
Defining a Monetization Event	34
Creating a Custom Event	36
Adding Attributes and Metrics	37
Reporting Detailed Data Points	38
Adding Attributes	38
Adding Metrics	38
Providing a Client Context	39
Giving Details of Application Context	39

Submitting Events	39
Event-Based Interaction	39
Signing Requests	40
Reports Overview	41
Toolbar	41
Using the Console Reports	42
Filtering Data Shown by Date Range	42
Selecting the Platform Displayed	42
Overview Tab	43
Active Users Tab	44
Sessions Tab	45
Revenue Tab	46
Retention Tab	47
Custom Events Tab	48
Working with Charts	49
Exporting Analytics	51
Getting Started with Auto Export to Amazon Redshift or Amazon S3	51
Step 1: Go to Manage Apps	51
Step 2: Launch the Auto Export Wizard	51
Step 3: Choose Export to Amazon Redshift + Amazon S3	52
Step 4: Configure an Amazon S3 Bucket	52
Step 5: Configure S3 and S3 to Amazon Redshift Permissions	52
Step 6: Configure Auto Export to Use Amazon Redshift	52
Connecting to Amazon Redshift	53
Managing Auto Export	53
Enabling Auto Export for Additional Apps	54
Disabling Auto Export	54
Exported Event JSON	55
Amazon Redshift Schema for Event Data	59
Example Amazon Redshift Queries	61
Explanation of Infrastructure	63
Amazon Redshift Cluster	63
Virtual Private Cloud	63
EC2 Instance	63
CloudWatch Metrics and Logs	64
Troubleshooting	65
Troubleshooting Default IAM Role Creation	65
Limits	68
REST API Reference	69
HTTP Requests	69
HTTP Header Contents	69
HTTP Body Content	70
PutEvents	71
Requests	71
Responses	76
Examples	76
Document History	77

What Is Amazon Mobile Analytics?

Amazon Mobile Analytics is a service for collecting, visualizing, understanding, and extracting app usage data at scale. Mobile Analytics easily captures both standard device data and custom events and automatically calculates reports on your behalf.



The following reports are available from the Mobile Analytics console:

- Daily Active Users (DAU), Monthly Active Users (MAU), and New Users
- Sticky Factor (DAU divided by MAU)
- Session Count and Average Sessions per Daily Active User
- Average Revenue per Daily Active User (ARPPDAU) and Average Revenue per Daily Paying Active User (ARPPDAU)
- Day 1, 3, and 7 Retention and Week 1, 2, and 3 Retention
- Custom Events

In addition to these aggregated reports, you can also set up your data to be exported automatically to Amazon Redshift and Amazon S3 for further analysis. For more information, see [Auto Export](#).

These reports are provided through six reporting tabs in the console:

- **Overview** – Track nine preselected reports in a simple-to-review dashboard to get a quick idea of engagement: MAU, DAU, New Users, Daily Sessions, Sticky Factor, 1-Day Retention, ARPPDAU, Daily Paying Users, ARPPDAU.
- **Active Users** – Track how many users engage with your app daily and monthly and monitor its stickiness to gauge engagement, appeal, and monetization.

- **Sessions** – Track how often your app is used on a given day and how often each user opens your app during a day.
- **Retention** – Track the rate at which customers come back to your app on a daily and weekly basis.
- **Revenue** – Track in-app revenue trends to identify areas for monetization improvement.
- **Custom events** – Track custom, defined user actions specific to your app.

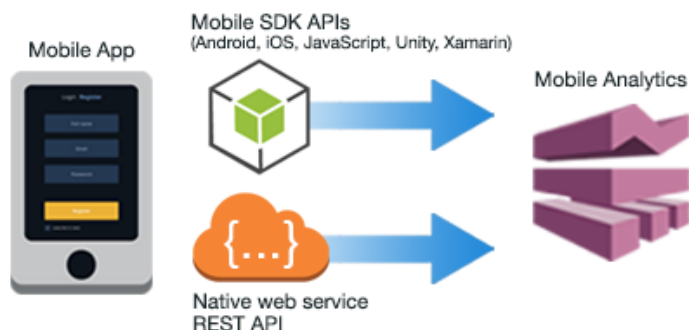
Incorporating Mobile Analytics

Mobile Analytics can be used by developers to capture information, through analytics events created in the application logic, about how their applications are used. Individual bits of data called *attributes* and *metrics* are added to an event created in the application.

After attributes or metrics have been added, the event is then recorded so a copy of the data is preserved in the device's file storage. When an application is sent to the background by the user, recorded events are then submitted to the Mobile Analytics service; this data contributes to the reports available in the console.

There are two ways to incorporate Mobile Analytics into the code of your app:

- The Mobile Analytics web service provides a REST API your app can invoke using HTTP PUT requests. The REST API is platform- and language-independent; it requires a properly formed HTTP request only.
- For mobile app developers, the AWS Mobile SDK provides APIs that enable easier and more robust interaction with Mobile Analytics for apps running on Android, iOS, JavaScript, Unity, or Xamarin.



Using the Mobile SDK

If you want to use Mobile Analytics in an Android, iOS, JavaScript, Unity, or Xamarin application, you will probably want to make API calls through the AWS Mobile SDK. For information about working with Mobile Analytics using the AWS Mobile SDK, see:

- [Android Developer Guide](#)
- [iOS Developer Guide](#)
- [Mobile Analytics SDK for JavaScript](#)
- [Unity Developer Guide](#)
- [Xamarin Developer Guide](#)

Getting Started

Before you can visualize the data in the console at <https://console.aws.amazon.com/mobileanalytics/home>, sign up for an AWS account, create a Cognito identity pool or an AWS Identity and Access Management (IAM) account, and then download the AWS Mobile SDK so that you can integrate it into your app.

Topics

- [Step 1: Sign Up for an AWS Account \(p. 3\)](#)
- [Step 2: Follow the Steps to Add an App \(p. 3\)](#)
- [Step 3: Integrate Mobile Analytics in to Your App \(p. 4\)](#)
- [Step 4: Review the Data in Amazon Mobile Analytics Reports \(p. 4\)](#)

Step 1: Sign Up for an AWS Account

If you already have an AWS account, sign in to your account and go to the next step.

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open <http://aws.amazon.com/> and click **Sign Up**.
2. Follow the on-screen instructions.

Step 2: Follow the Steps to Add an App

1. Open the Amazon Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.
2. If this is the first time you are using Amazon Mobile Analytics, follow the on-screen instructions to add an app.

If you have already added an app, click the settings icon .

3. In **Manage Apps**, choose **Add App**.
4. Follow the instructions.

Step 3: Integrate Mobile Analytics in to Your App

To integrate Mobile Analytics, use the SDK for platforms appropriate to your app:

- [Using Mobile Analytics in an Android app](#)
- [Using Mobile Analytics in an iOS app](#)
- [Using Mobile Analytics in a JavaScript app](#)
- [Using Mobile Analytics in a Unity app](#)
- [Using Mobile Analytics in an Xamarin app](#)

Step 4: Review the Data in Amazon Mobile Analytics Reports

You can see your data in the console at <https://console.aws.amazon.com/mobileanalytics/home> for events reported on iOS and Android platforms. It can take up to an hour for data sent to Mobile Analytics to appear in the console reports.

Adding and Managing Apps

The steps you follow to create an app that uses Mobile Analytics will depend on the platforms you target and the tools and programming languages associated with each. But generally speaking, to use Mobile Analytics, you must do two things:

- Add and set up your app in the console at <https://console.aws.amazon.com/mobileanalytics/home>.
- Add the code to your app that accesses the Mobile Analytics service as determined by the platforms your app will support.

Accessing Mobile Analytics

In order for your app to directly access AWS it must have credentials, which you obtain through Amazon Cognito Identity. An Amazon Cognito identity pool defines user identities used with your account.

When you create your first app, the console creates a Cognito identity pool you can use to authenticate your apps with Mobile Analytics. The console also provides, for each platform, integration code snippets that include the Cognito identity pool ID your apps need to communicate with Mobile Analytics. If you select a different Cognito identity pool to use in your apps, these code snippets will be updated to reflect it.

Topics

- [Adding an App to Mobile Analytics \(p. 5\)](#)
- [Renaming an App \(p. 15\)](#)
- [Viewing Integration Steps \(p. 16\)](#)

Adding an App to Mobile Analytics


To use Mobile Analytics with an app, you must use the console to add the app to the Mobile Analytics service. The steps to add your first app to Mobile Analytics are different from the steps to add additional apps.

If You Have No Existing Identity Pools

Use the following procedure if you are adding your first app in the Mobile Analytics console but do not yet have any Cognito identity pools defined.

To add an app for the first time


1. Open the Amazon Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.



Amazon Mobile Analytics


Amazon Mobile Analytics is a service that lets you simply and cost effectively collect and analyze app usage data.

[Get Started](#)
[Try Product Demo](#)




Custom Data Analysis

Amazon Mobile Analytics Auto Export enables you to set up automatic export of your data to Amazon S3 for use with other data analytics tools such as Amazon Redshift, Amazon Elastic MapReduce (EMR), Extract, Transform and Load (ETL) software, or your own data warehouse.



Own Your Data

Unlike other mobile analytics providers, when you use Amazon Mobile Analytics, we do not report on or share your data with third parties. You can also archive your raw events data to your own Amazon S3 bucket for processing by other tools.



Inexpensive

Amazon Mobile Analytics offers a free tier of 100 million free events per month. Beyond the free tier, costs just \$1.00 per million events per month, and if your app generates more than 1 billion events per month, please contact us.

Additional Resources
[Getting Started Guide](#) | [FAQs](#) | [Pricing](#)

Support
Visit our [Amazon Mobile Analytics Forum](#)

2. Choose **Get Started**.
3. In **App Name**, type a name for your app, and then choose **Create App**.

The screenshot shows the 'Create Mobile Analytics App' wizard in the AWS console. The top navigation bar includes 'AWS', 'Services', and 'Edit'. The user is logged in as 'John Stiles' with a 'Global' region. The breadcrumb trail shows 'Amazon Mobile Analytics' > 'Dashboard' > 'App Management'. The main heading is 'Create Mobile Analytics App'. Below it, the wizard progress is shown: 'Step 1: Add Your App' (active) and 'Step 2: Integrate SDK'. The 'Add Your App' section contains the instruction 'Please enter your app name.' and a text input field labeled 'App Name*' with the placeholder text 'Your App'. A required field asterisk is shown below the input. At the bottom right, there are 'Cancel' and 'Create A' buttons.

4. In the IAM role creation wizard, choose **Allow** to generate an IAM role for the Cognito identity pool.

The screenshot shows the IAM role creation wizard. The top navigation bar is the same as in the previous screenshot. The main heading is 'Amazon Mobile Analytics is requesting permission to create an IAM role'. Below this, the text reads: 'Click "Allow" to create an IAM role that will be used to send your app events to Amazon Mobile Analytics.' There is a 'View Details' link with a right-pointing arrow. At the bottom right, there is a 'Don't Allow' button.

5. In the displayed integration steps, choose the tab that corresponds to the platform targeted by your app. The Cognito identity pool ID generated by the wizard in the previous step appears in the sample code. You can easily copy and paste the code into your app.

AWS Services Edit John Stiles Global

Amazon Mobile Analytics MyFirstApp Dashboard App Management

Integration Steps for MyFirstApp

🍏 iOS Android Unity JavaScript Xamarin

1. Include the SDK

CocoaPods Manual

Update your Podfile to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

2. Initialize Mobile Analytics

Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```


In the application:didFinishLaunchingWithOptions: method in the ApplicationDelegate for your app, use the following code to initialize Mobile Analytics client.

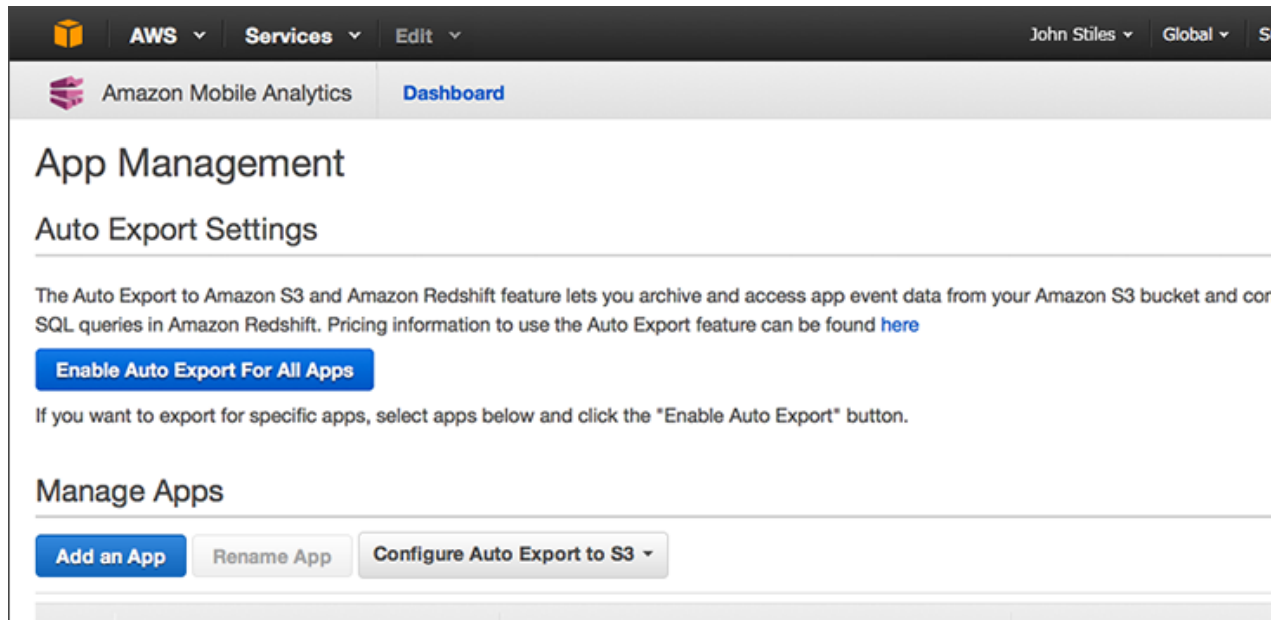
```
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider alloc] initWithRegionType: AWSRegionUSEast1
```

Adding Additional Apps with the Default Cognito Identity Pool

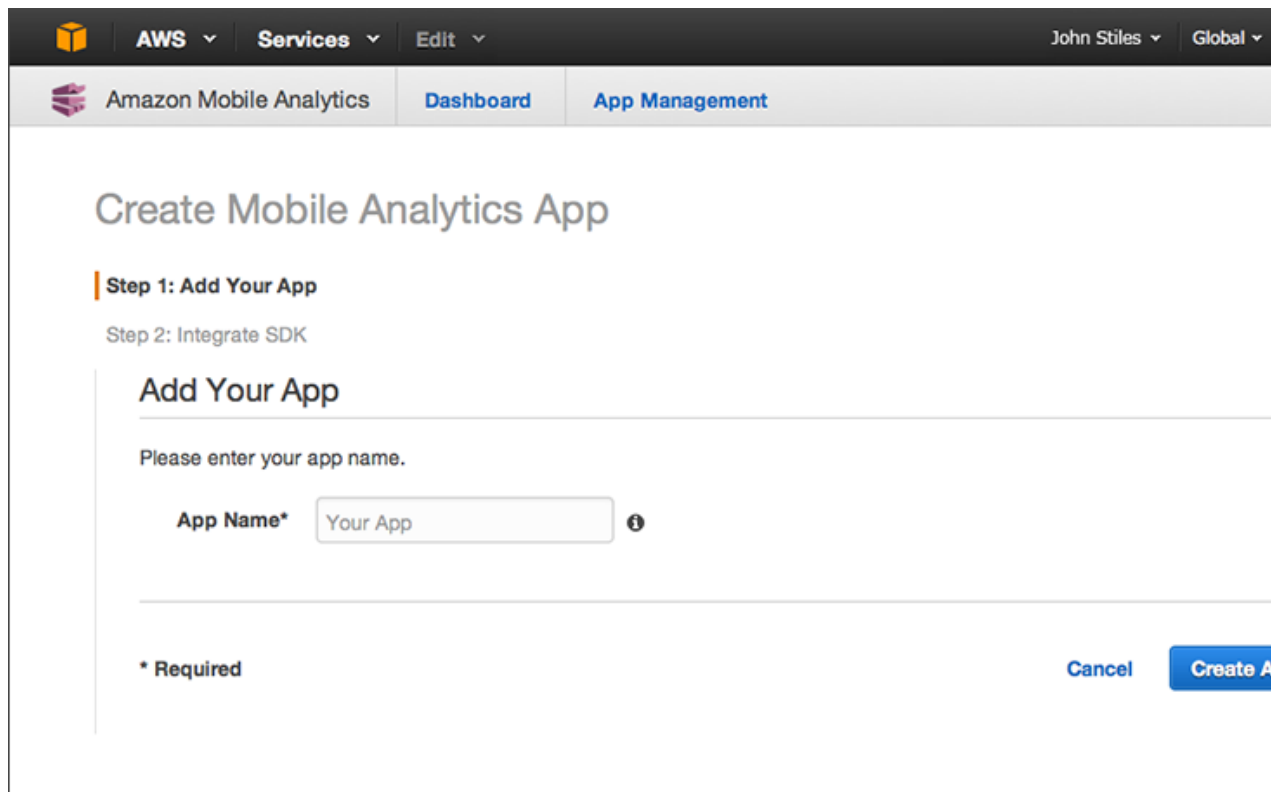
Use the following procedure to add a new app if you have already added your first app and have a default Cognito identity pool.

To add an app if you have existing apps

1. Open the Amazon Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.
2. Click the Settings  icon.
3. In **Manage Apps**, choose **Add an App**.

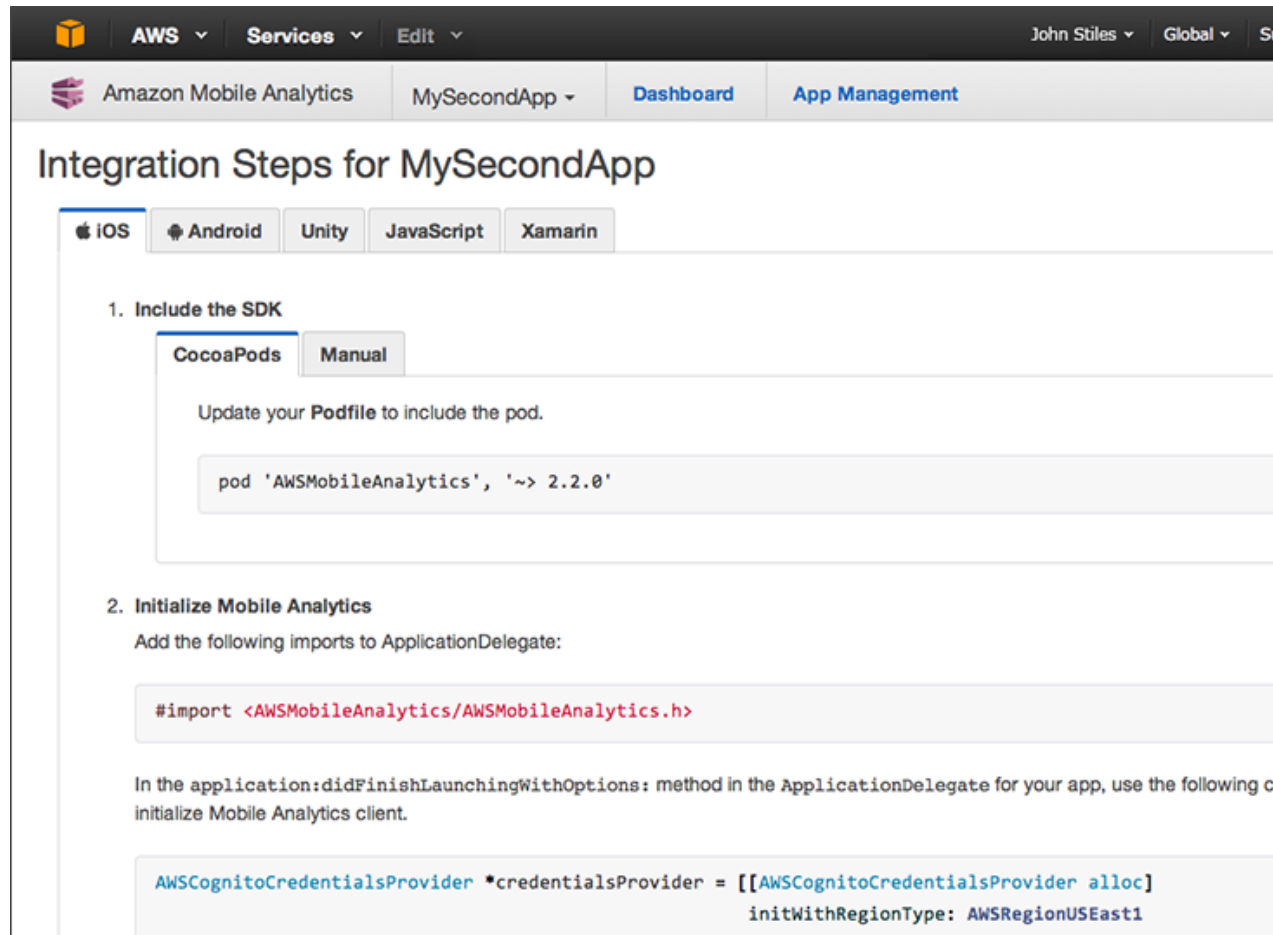


4. In **App Name**, type a name for your app, and then choose **Create App** to add the app using your default identity pool. This name will appear in the console.



5. Choose the tab that corresponds to the platform targeted by your app for details about how to integrate Mobile Analytics. The ID for your default identity pool appears in the sample code. You can easily copy and paste the code into your app. \

Amazon Mobile Analytics User Guide
If You Have Existing Identity Pools but No Default
Cognito Identity Pool



The screenshot shows the Amazon Mobile Analytics console interface. At the top, there are navigation tabs for 'AWS', 'Services', and 'Edit'. The user is logged in as 'John Stiles' and the location is 'Global'. The main header includes 'Amazon Mobile Analytics', 'MySecondApp', 'Dashboard', and 'App Management'. The main content area is titled 'Integration Steps for MySecondApp' and features tabs for different platforms: 'iOS', 'Android', 'Unity', 'JavaScript', and 'Xamarin'. Under the 'iOS' tab, there are two sub-sections: '1. Include the SDK' and '2. Initialize Mobile Analytics'. The '1. Include the SDK' section has two sub-tabs: 'CocoaPods' and 'Manual'. The 'Manual' sub-tab is active, showing instructions to 'Update your Podfile to include the pod.' with a code block:

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

. The '2. Initialize Mobile Analytics' section instructs to 'Add the following imports to AppDelegate:' with a code block:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```


. Below this, it says 'In the application:didFinishLaunchingWithOptions: method in the AppDelegate for your app, use the following code to initialize Mobile Analytics client.' with another code block:

```
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider alloc] initWithRegionType: AWSRegionUSEast1
```

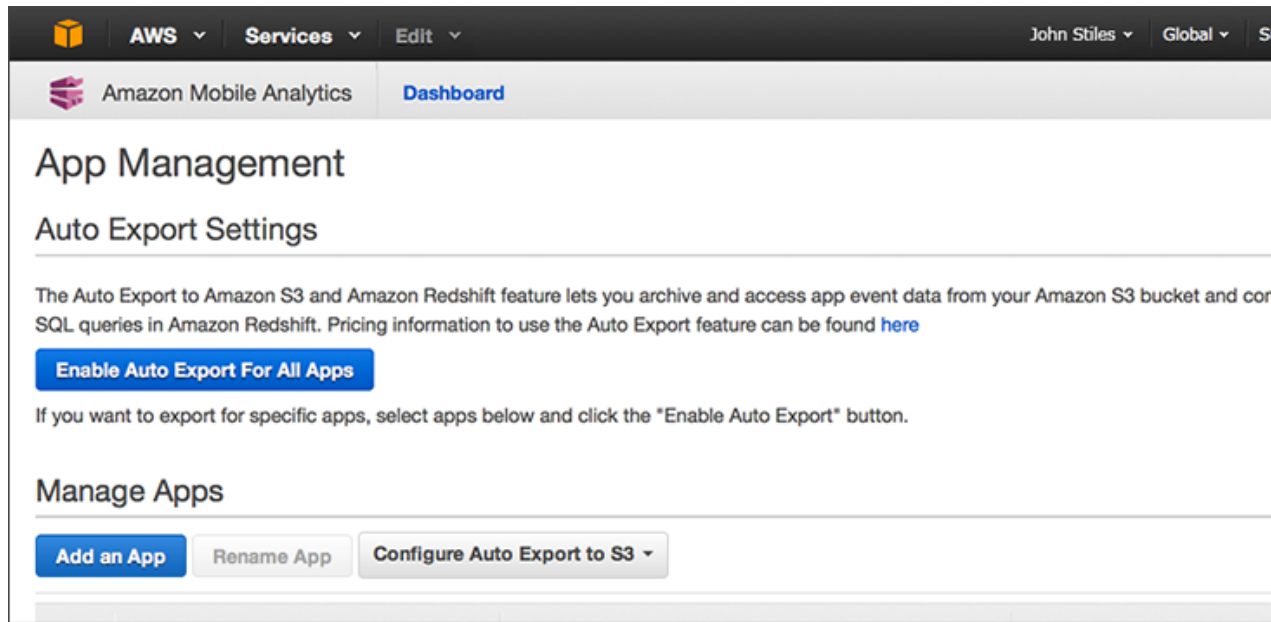
If You Have Existing Identity Pools but No Default Cognito Identity Pool

Use this procedure to add a new app if you used Mobile Analytics before July 6, 2015 or have an Cognito identity pool but have not selected a default .:

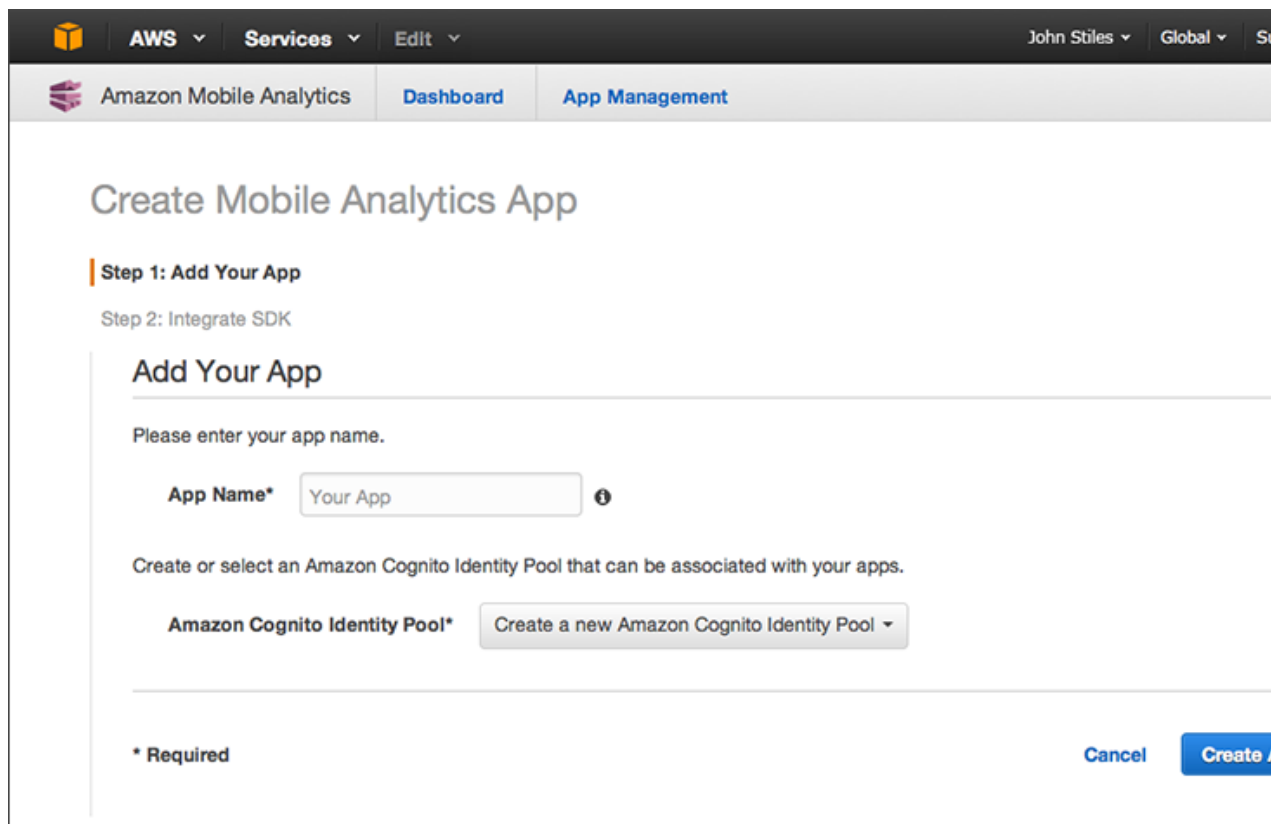
To add an app if you have existing identity pools, but no default identity pool

1. Open the Amazon Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.
2. Click the Settings  icon.
3. In **Manage Apps**, choose **Add an App**.

Amazon Mobile Analytics User Guide
If You Have Existing Identity Pools but No Default
Cognito Identity Pool

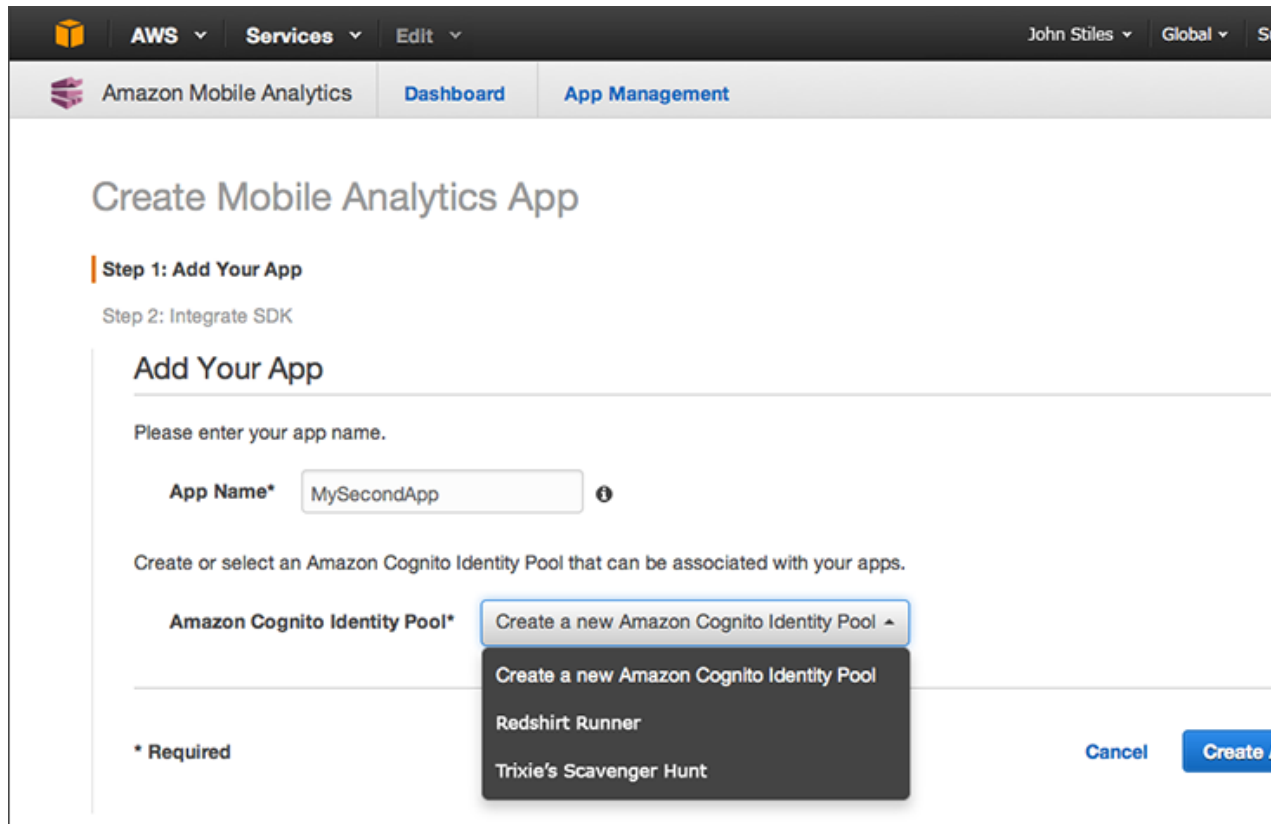


4. In **App Name**, type a name for your app. This name will appear in the console.

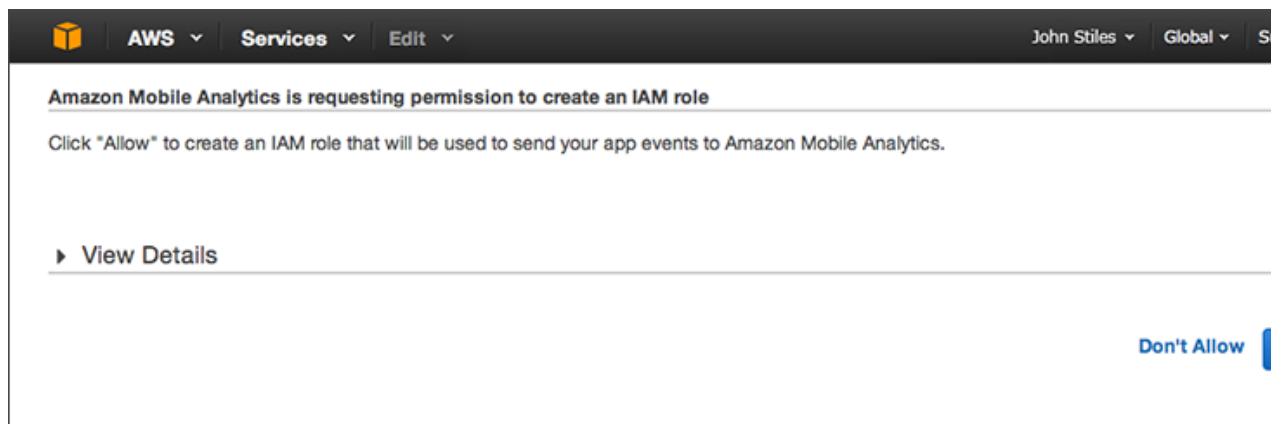


5. To use an existing identity pool, choose it from the **Amazon Cognito Identity Pool** drop-down list. To create a new identity pool, choose **Create a new Amazon Cognito Identity Pool**

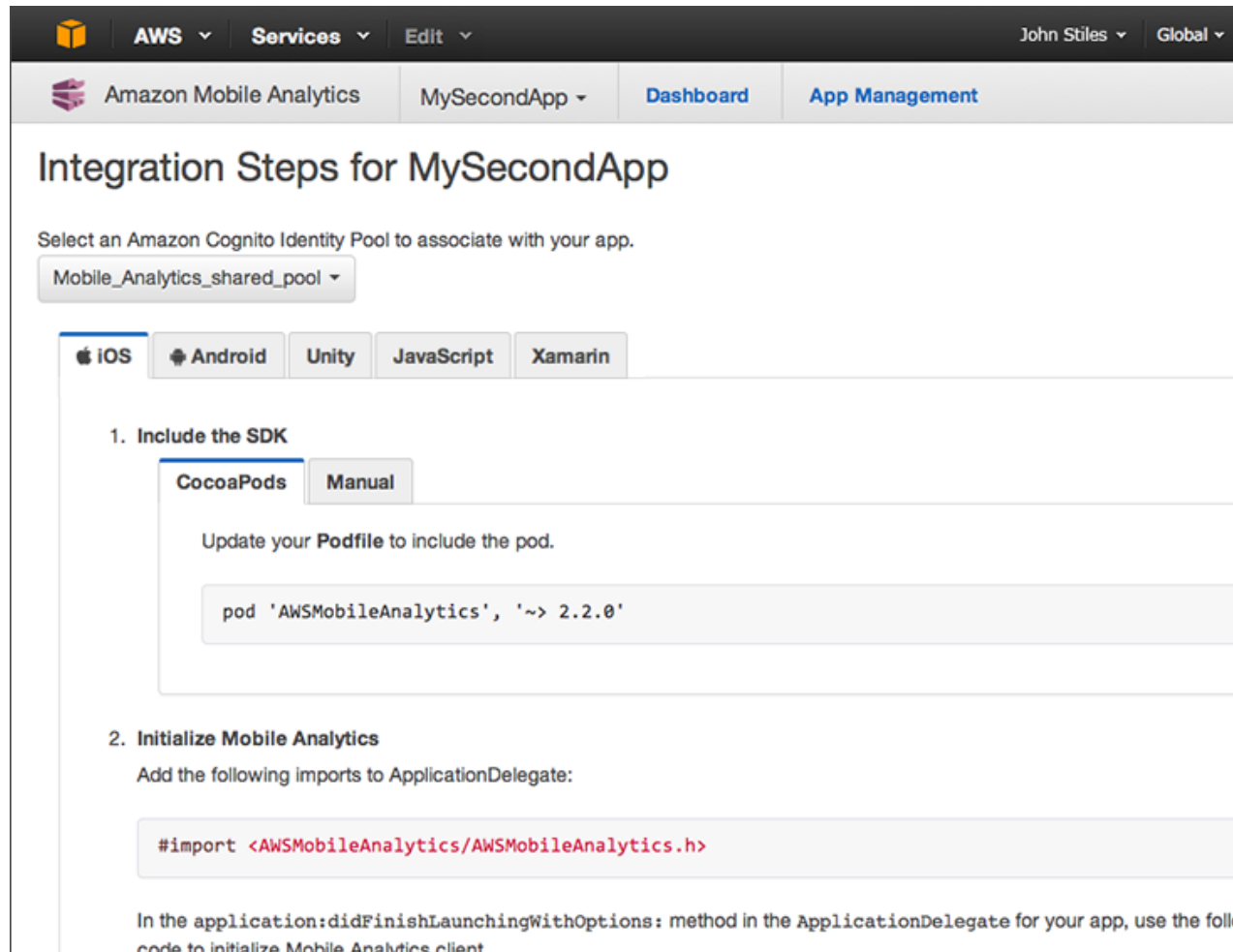
Amazon Mobile Analytics User Guide
If You Have Existing Identity Pools but No Default
Cognito Identity Pool



6. Choose **Create App**.
7. If you chose to create a new identity pool, the IAM role creation wizard will appear. Choose **Allow** to generate an IAM role for the Cognito identity pool.



8. Choose the tab for each platform targeted by your app for instructions and sample code. The ID for the Cognito identity pool you created or chose appears in the sample code. You can easily copy and paste the code into your app.



The screenshot shows the Amazon Mobile Analytics console interface. At the top, there's a navigation bar with 'AWS', 'Services', and 'Edit' menus, and user information 'John Stiles' and 'Global'. Below this, the console title is 'Amazon Mobile Analytics' with a dropdown for 'MySecondApp'. There are tabs for 'Dashboard' and 'App Management'. The main heading is 'Integration Steps for MySecondApp'. Below the heading, it says 'Select an Amazon Cognito Identity Pool to associate with your app.' and a dropdown menu shows 'Mobile_Analytics_shared_pool'. There are tabs for different platforms: 'iOS', 'Android', 'Unity', 'JavaScript', and 'Xamarin'. The 'iOS' tab is selected. Under '1. Include the SDK', there are sub-tabs for 'CocoaPods' and 'Manual'. The 'CocoaPods' sub-tab is active, showing the instruction 'Update your Podfile to include the pod.' and a code block:

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

. Under '2. Initialize Mobile Analytics', it says 'Add the following imports to AppDelegate:' and shows a code block:


```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

. At the bottom, it says 'In the application:didFinishLaunchingWithOptions: method in the AppDelegate for your app, use the following code to initialize Mobile Analytics client.'

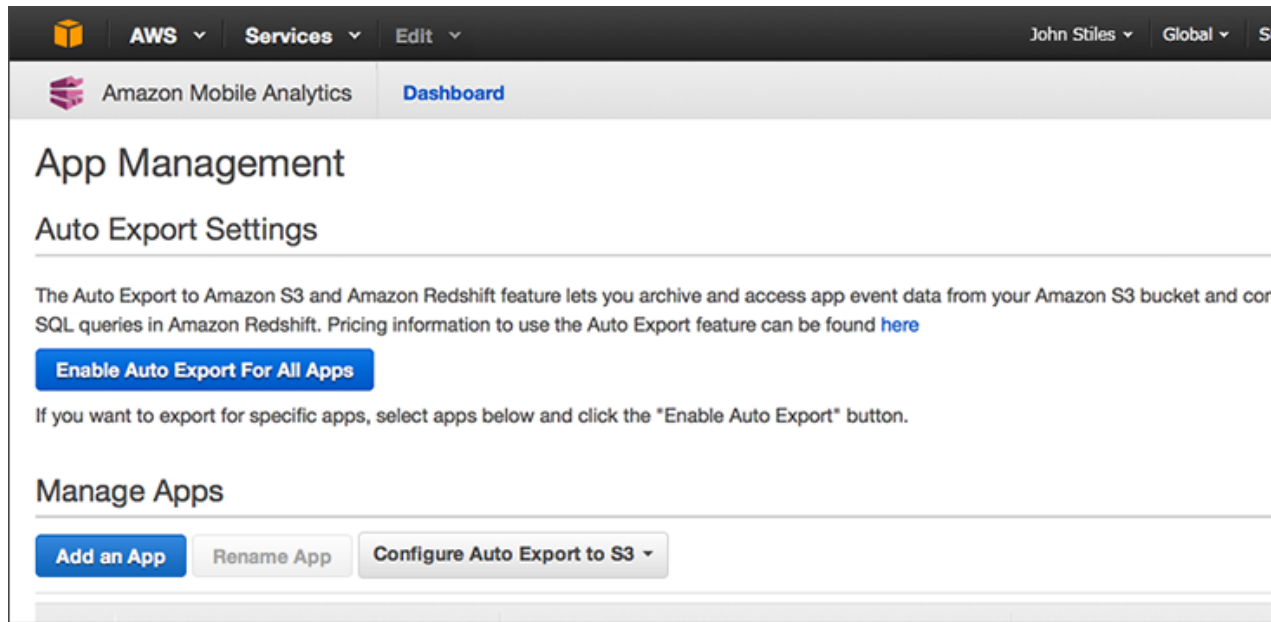
If You Have Existing Identity Pools and a Default Cognito Identity Pool

Use this procedure if you are Amazon Cognito user who would like to start using Mobile Analytics.

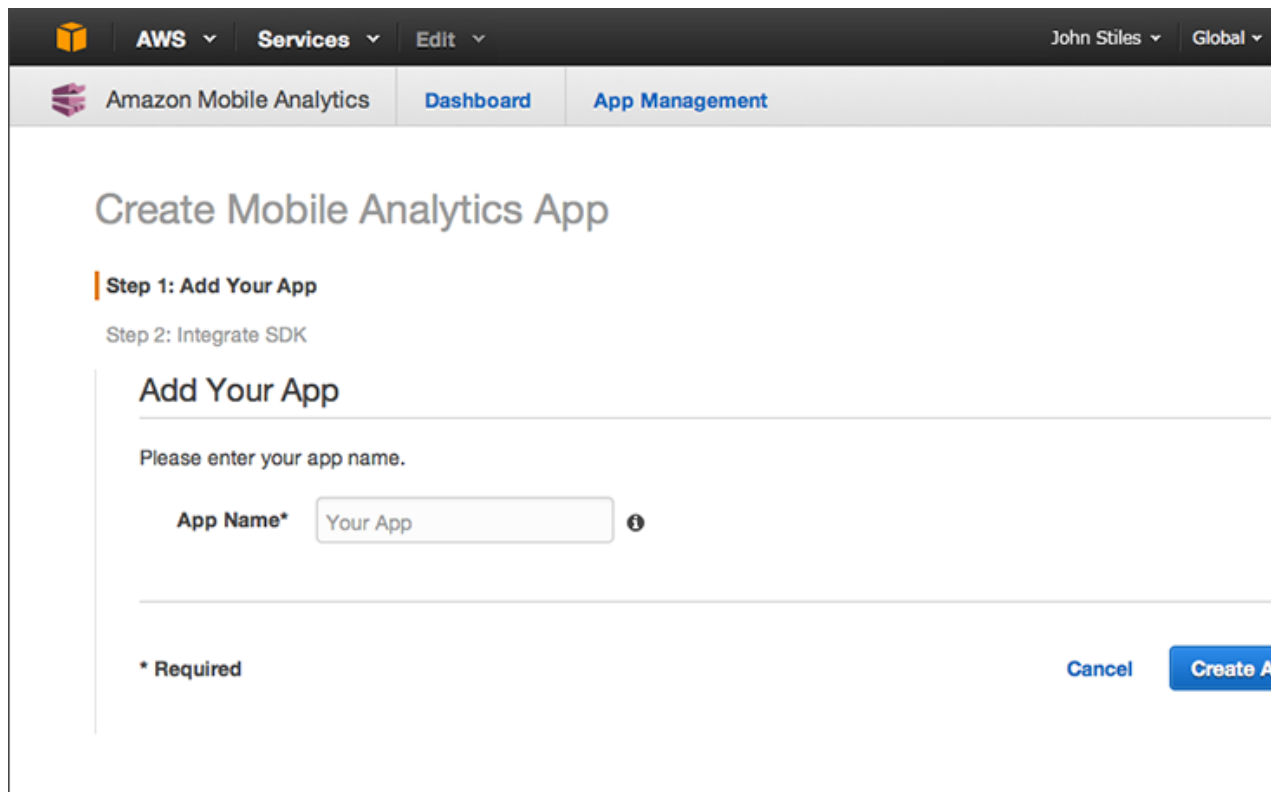
To add an app if you have existing identity pools and a default identity pool

1. Open the Amazon Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.
2. Click the Settings  icon.
3. In **Manage Apps**, choose **Add an App**.

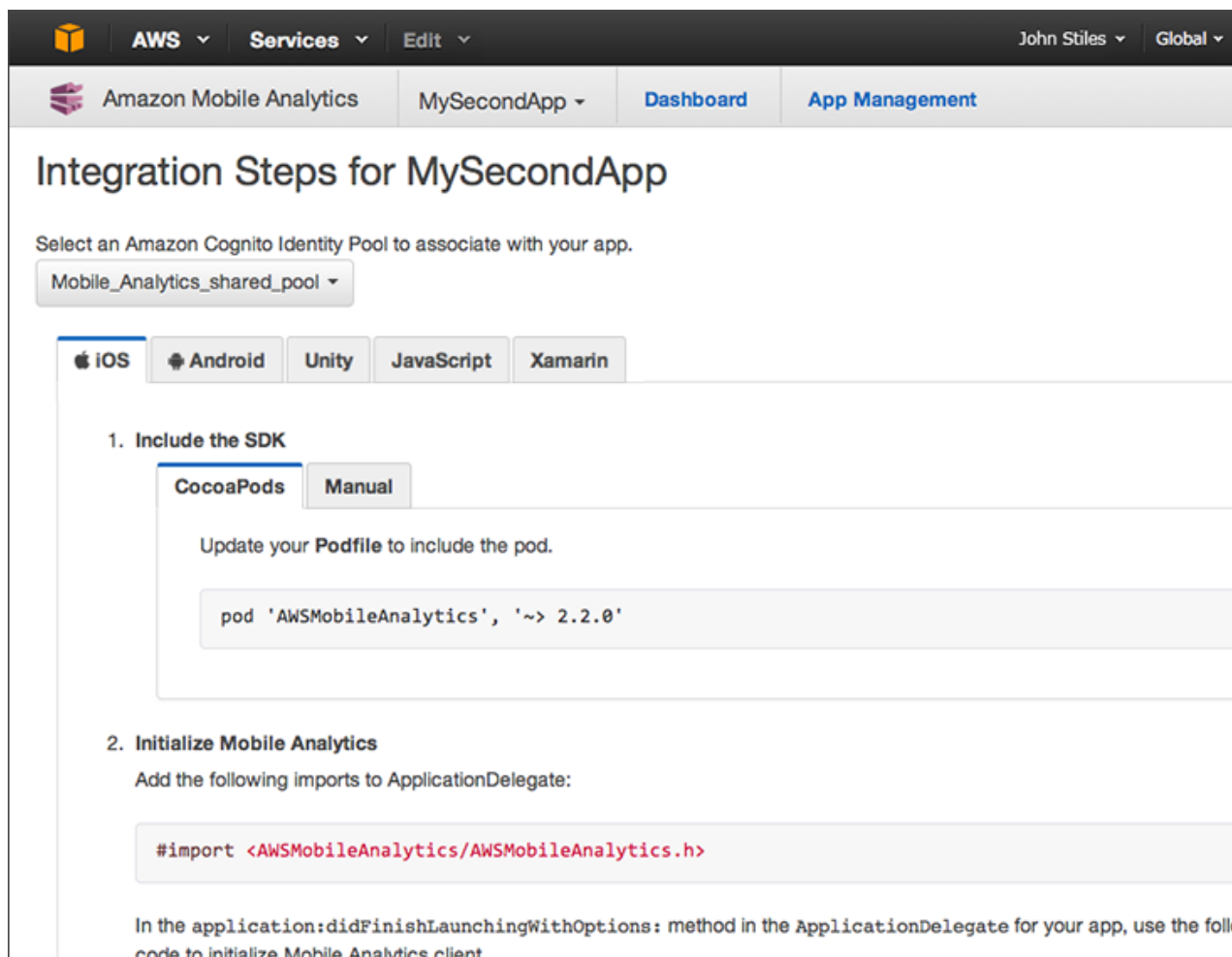
Amazon Mobile Analytics User Guide
If You Have Existing Identity Pools and a Default Cognito Identity Pool



4. In **App Name**, type a name for your app, and then choose **Create App**. This name will appear in the console.



5. Choose the tab for each platform targeted by your app for instructions and sample code to use. The ID for your default Cognito identity pool is automatically associated with the new app, but you can select an other identity pools from the drop-down list. The sample code contains the values appropriate to your app. You can easily copy and paste the values into your app's source code.

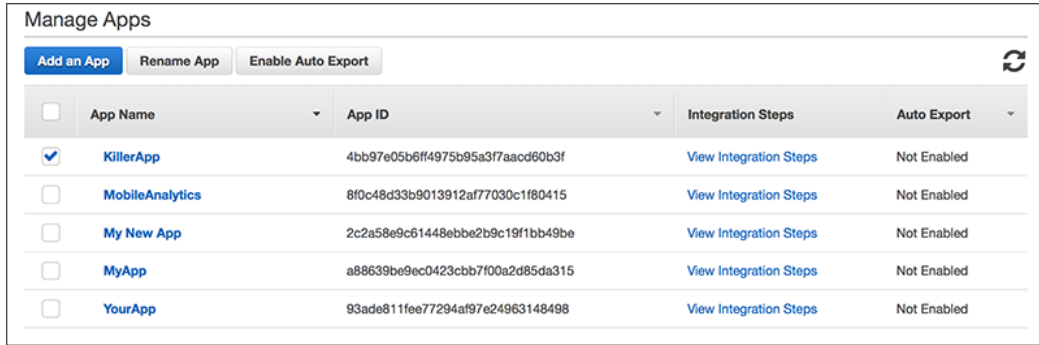


Renaming an App

You can rename an app in the **App Management** section of the console at <https://console.aws.amazon.com/mobileanalytics/home>.

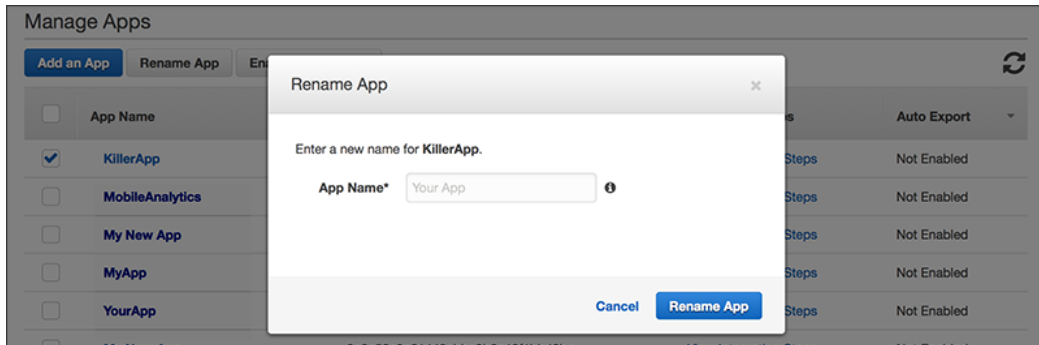
To rename an app

1. In the Mobile Analytics console, go to **App Management**.



<input type="checkbox"/>	App Name	App ID	Integration Steps	Auto Export
<input checked="" type="checkbox"/>	KillerApp	4bb97e05b6ff4975b95a377aacd60b3f	View Integration Steps	Not Enabled
<input type="checkbox"/>	MobileAnalytics	8f0c48d33b9013912af77030c1f80415	View Integration Steps	Not Enabled
<input type="checkbox"/>	My New App	2c2a58e9c61448ebbe2b9c19f1bb49be	View Integration Steps	Not Enabled
<input type="checkbox"/>	MyApp	a88639be9ec0423cbb7f00a2d85da315	View Integration Steps	Not Enabled
<input type="checkbox"/>	YourApp	93ade811fee77294af97e24963148498	View Integration Steps	Not Enabled

2. In the **Manage Apps** list, select the check box next to the app you want to rename.
3. Choose **Rename App**.
4. In the **App Name** box, type the new name for the app, and then choose **Rename App**.



Viewing Integration Steps

Mobile Analytics helps you integrate the service into the source code for your app. It provides blocks of code you can copy and paste into your source code as well as information about where in your app to add it. Where an integration step requires it, the code includes the `appID` to connect the data sent by the app to Mobile Analytics for generating reports and stats.

To view the integration steps for an app

1. From the **Dashboard**, choose **Manage Apps** in the application list in the toolbar.
2. In the **Manage Apps** list, choose **View Integration Steps** next to the app whose details you want to view.

AWS Services Edit John Stiles Global

Amazon Mobile Analytics MySecondApp Dashboard App Management

Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

Mobile_Analytics_shared_pool

iOS Android Unity JavaScript Xamarin

1. Include the SDK

CocoaPods Manual

Update your Podfile to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

2. Initialize Mobile Analytics

Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

In the application:didFinishLaunchingWithOptions: method in the ApplicationDelegate for your app, use the following code to initialize Mobile Analytics client.

3. If you have additional non-default identity pools, you can change the identity pool associated with this app. Simply choose it from the drop-down menu.

AWS Services Edit John Stiles Global

Amazon Mobile Analytics MySecondApp Dashboard App Management

Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

Mobile_Analytics_shared_pool

Mobile_Analytics_shared_pool JavaScript Xamarin

Redshirt Runner

Trixie's Scavenger Hunt

CocoaPods Manual

Update your Podfile to include the pod.

Using Mobile Analytics with the Mobile SDK

To take advantage of Mobile Analytics, your mobile app must incorporate code that generates, records, and submits events to the Mobile Analytics service. This section describes concepts you need to know to incorporate Mobile Analytics into a mobile app.

To integrate Mobile Analytics features that make data collection possible, use the APIs provided by the AWS Mobile SDK for the platforms you plan to support. When working on platforms other than those supported by the AWS Mobile SDK, use the REST API to access Mobile Analytics.

Platforms Supported by the Mobile SDK

Mobile Analytics is supported by the AWS Mobile SDK, providing an API for each of these platforms:

- **iOS:** [Track App Usage Data with Amazon Mobile Analytics](#).
- **Android:** [Amazon Mobile Analytics](#).
- **JavaScript:** [Mobile Analytics SDK for JavaScript](#).
- **Unity:** [Amazon Mobile Analytics](#).
- **Xamarin:** [Amazon Mobile Analytics](#).

If you are working on platforms other than those supported by the AWS Mobile SDK, use the REST API to access the Mobile Analytics service.

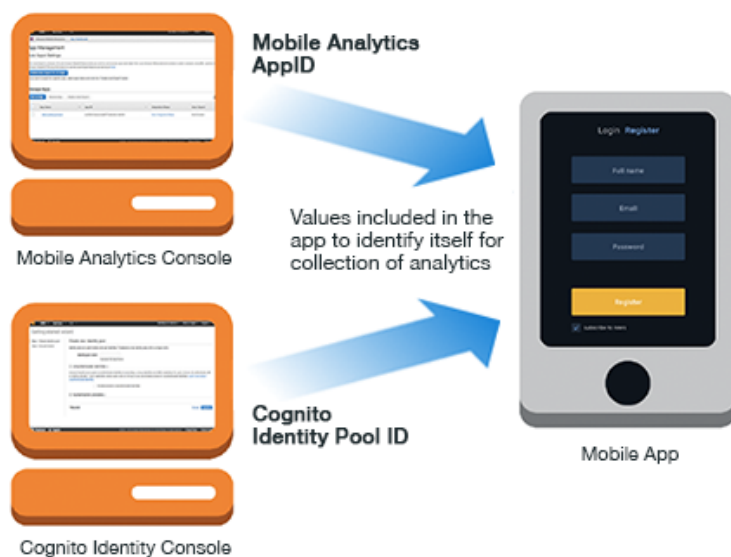
Topics

- [Identifying the App in Mobile Analytics \(p. 19\)](#)
- [Managing Sessions \(p. 19\)](#)
- [Generating Mobile Analytics Events \(p. 21\)](#)
- [Adding Attributes and Metrics \(p. 26\)](#)
- [Submitting Events \(p. 28\)](#)

Identifying the App in Mobile Analytics

Before Mobile Analytics can collect and present any analytic data about your application, you must identify your app to Mobile Analytics by providing the following information:

- The Amazon Cognito identity pool created to authenticate users of your app.
- The AppID created in the Mobile Analytics console when you added your app to Mobile Analytics.



For information about providing the Mobile Analytics `appID` and the Cognito identity pool ID in the source code of your app, see the AWS Mobile SDK for your platform.

- [AWS Mobile SDK Android Developer Guide](#)
- [AWS Mobile SDK iOS Developer Guide](#)
- [Mobile Analytics SDK for JavaScript](#)
- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS Mobile SDK Xamarin Developer Guide](#)

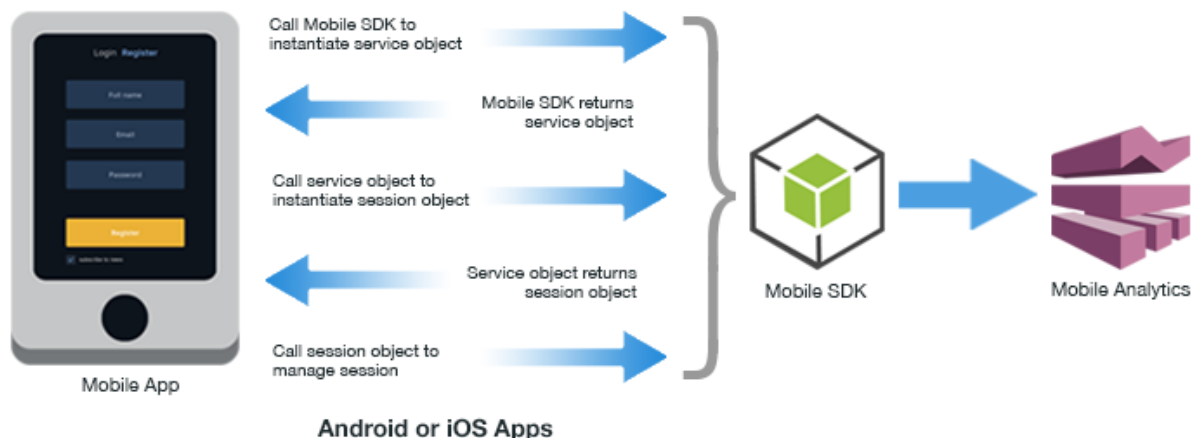
Managing Sessions

Mobile Analytics reports app usage data based on activity captured within sessions. A *session* is a period of time during which the app is active on the device. A session begins when an app is launched or brought to the foreground, and ends when the app is terminated or goes to the background. There is an inactivity period of up to 5 seconds, so a brief interruption, like receiving a text message, does not count as a new session.

Mobile Analytics tracks sessions and events submitted during each session to generate the reports displayed in the console. Total Daily Sessions shows the number of sessions your app has each day. Average Sessions per Daily Active User shows the mean number of sessions per user per day.

Managing Sessions in iOS or Android Apps

Apps that use the AWS SDK for iOS or AWS SDK for Android to access Mobile Analytics start a session when they create the service object for their platform. To manage a running session, the app must additionally create the session object. This session object gives your app access to session management.



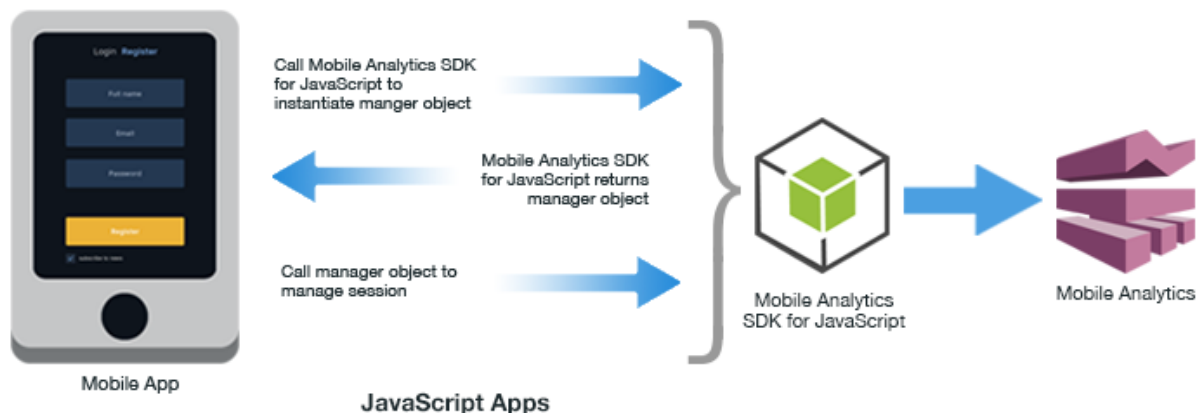
For information about managing sessions in iOS or Android apps, see:

- [AWS Mobile SDK iOS Developer Guide](#)
- [AWS SDK for iOS Reference](#)

- [AWS Mobile SDK Android Developer Guide](#)
- [AWS SDK for Android Reference](#)

Managing Sessions in JavaScript Apps

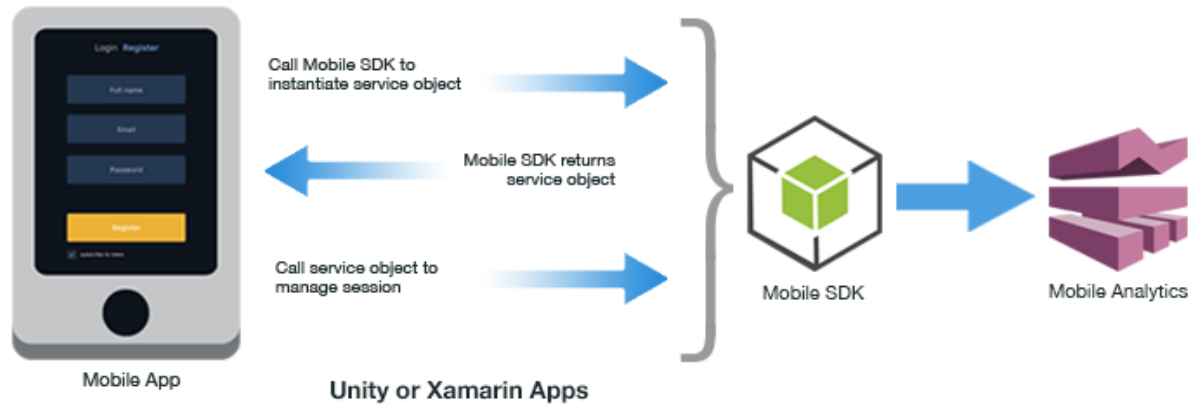
JavaScript apps must include the [AWS Mobile SDK for JavaScript in the Browser](#) as well as the [Amazon Mobile Analytics SDK for JavaScript](#). When your app starts, it must create an `AMA.Manager` object, which starts a session and gives your app access to session management.



For information about managing sessions in JavaScript apps, see the [Amazon Mobile Analytics SDK for JavaScript](#).

Managing Sessions in Unity or Xamarin Apps

When your app starts, it should start a session that Mobile Analytics can then begin to track. Apps that use the AWS Mobile SDK for Unity or Xamarin to access Mobile Analytics automatically start a session when they create the service object. The service object gives your app access to session management.



For information about session management in Unity and Xamarin apps, refer to

- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS SDK for Unity Reference](#)

- [AWS Mobile SDK Xamarin Developer Guide](#)
- [AWS SDK for Xamarin Reference](#)

Generating Mobile Analytics Events

Information about user engagement in your app is sent to the Mobile Analytics service using events. The events needed to produce the basic analytics reports in the console are collected and sent automatically as long as a session is active. There are two types of events your app can generate in addition to those generated automatically:

- Monetization events
- Custom events

Monetization events are specialized events used to report on monetization activities in the app, such as in-app purchases. Custom events are those you create to monitor activities specific to your app, such as completing a level in a game, posting to social media, or setting particular app preferences. An app can have up to 1,500 unique custom events.

Naming Custom Events

When naming custom events, do not begin event names with an underscore (_). Events with names beginning with an underscore are filtered out.

Topics

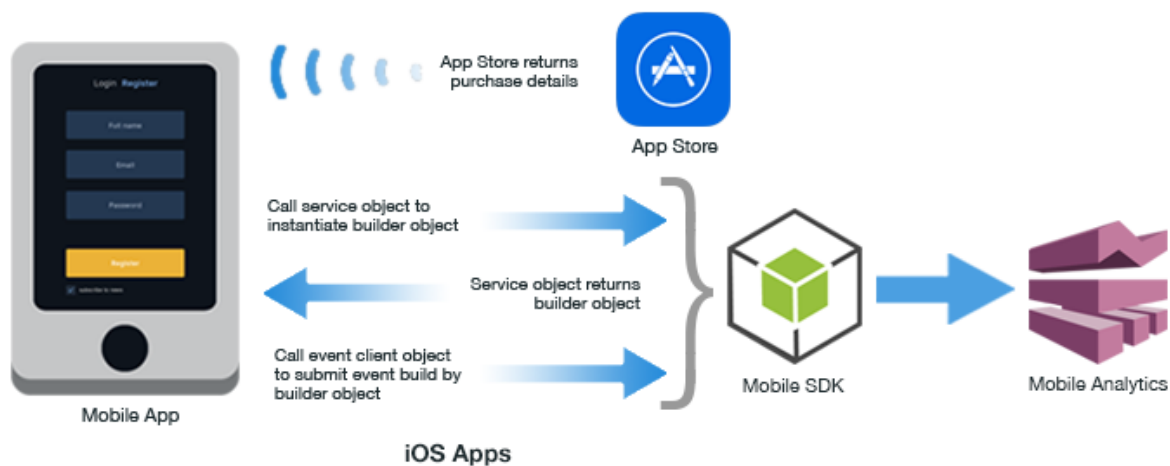
- [Creating a Monetization Event \(p. 22\)](#)
- [Creating a Custom Event \(p. 24\)](#)

Creating a Monetization Event

When a user of your app makes a purchase, the app code handling the purchase sends a monetization event. Data from monetization events are used to enable and populate revenue-focused reports in the console such as Average Revenue Per User (ARPU) and others.

Creating a Monetization Event in iOS Apps

When a user of your app makes an in-app purchase, you can create a monetization event that helps you track the monetary performance of the app. To do this in an iOS app using the AWS Mobile SDK create a builder object to hold data returned by the Apple App Store. You then use the builder object to build a monetization event you [submit with the event client object](#) (p. 24).

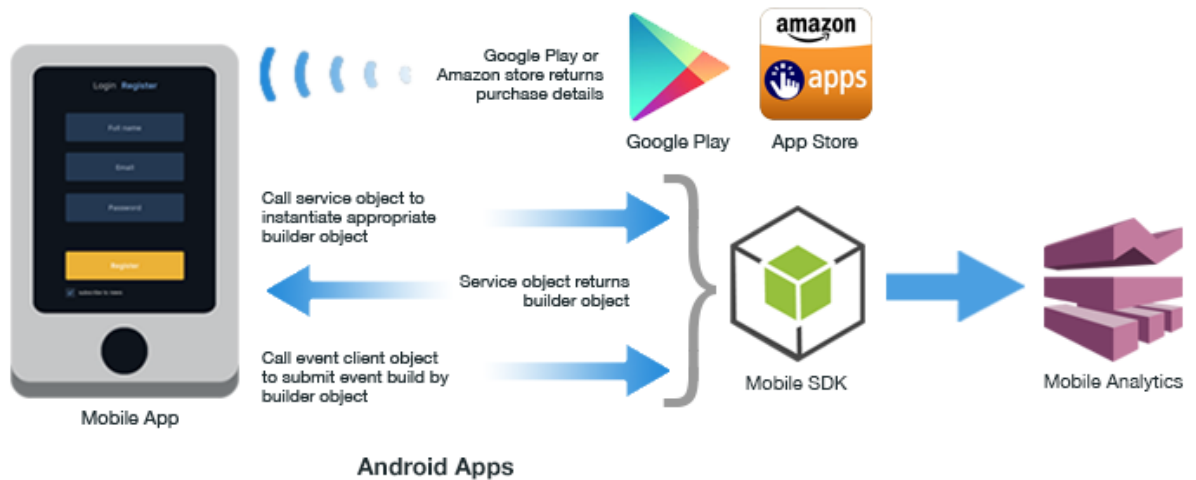


For information about creating monetization events in iOS apps, see:

- [AWS Mobile SDK iOS Developer Guide](#)
- [AWS SDK for iOS Reference](#)

Creating a Monetization Event in Android Apps

When a user makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in an Android app using the AWS Mobile SDK, create a builder object to hold data returned by the online store. There are distinct builder object classes to handle purchases from the Amazon store, the Google Play store, and from an IAP framework not defined by a specific builder. You then use the builder object to build a monetization event you [submit with the event client object](#) (p. 24).

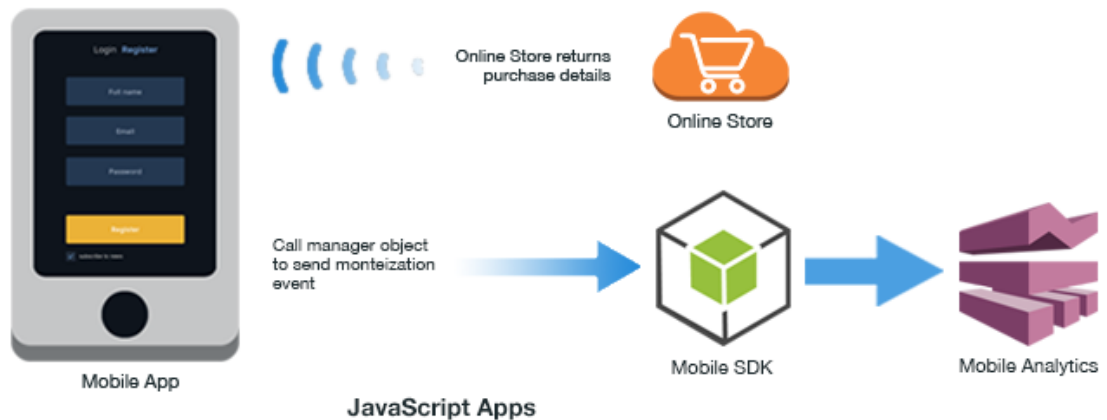


For information about creating monetization events in Android apps, see:

- [AWS Mobile SDK Android Developer Guide](#)
- [AWS SDK for Android Reference](#)

Creating a Monetization Event in JavaScript Apps

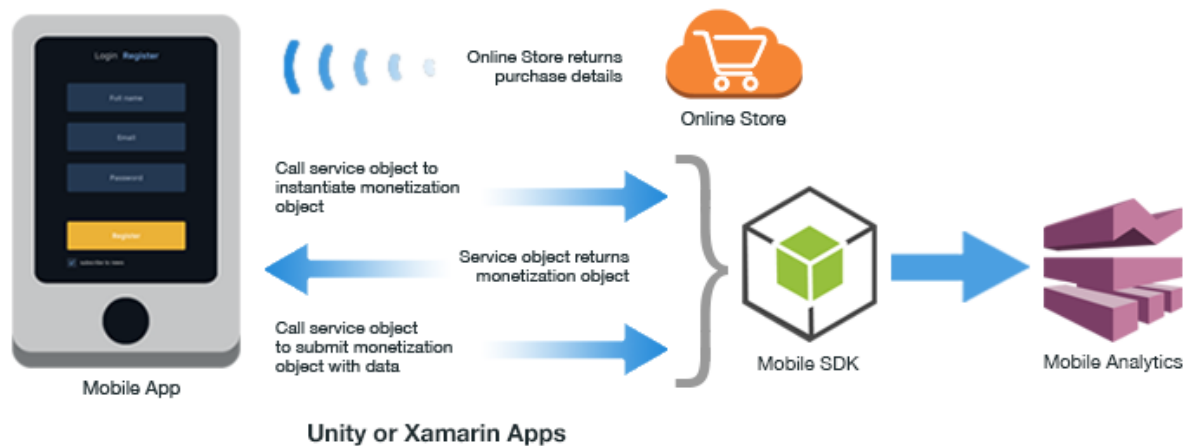
When a user of your app makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in a JavaScript app using the Mobile Analytics SDK for JavaScript, you call a method on the manager object, passing the purchase data as parameters.



For information about creating monetization events in JavaScript apps, see [Mobile Analytics SDK for JavaScript](#).

Creating a Monetization Event in Unity or Xamarin Apps

When a user of your app makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in a Unity app using the AWS Mobile SDK for Unity or a Xamarin app using the AWS Mobile SDK for Xamarin, create a monetization event object to hold data from the online store. Populate the purchase data, and then submit the monetization object to a method on the service object.



For information about creating monetization events in Unity or Xamarin apps, see:

- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS SDK for Unity Reference](#)

- [AWS Mobile SDK Xamarin Developer Guide](#)
- [AWS SDK for Xamarin Reference](#)

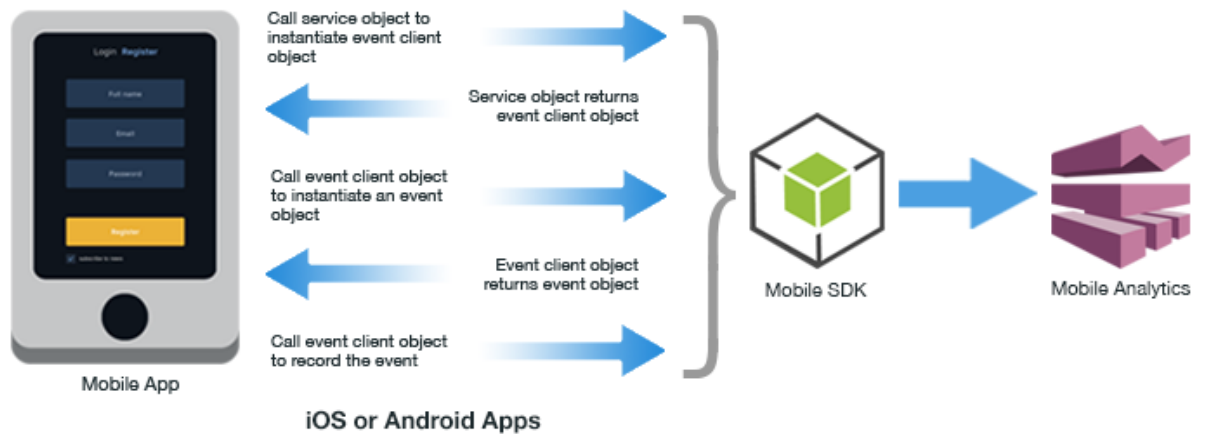
Creating a Custom Event

You assign an `eventType` to your custom event. As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate.

Creating a Custom Event in iOS or Android Apps

Custom events in iOS and Android apps are created and defined using an event object that the app submits to Mobile Analytics. To create these event objects, the app must first create an event client object, and then request individual event objects from the event client.

After the app has used an event client object to obtain a single event object, the app customizes the event object by [adding attributes and metrics \(p. 26\)](#) that specify data values to report. After the event object is customized, the app calls the event client object to record the event.



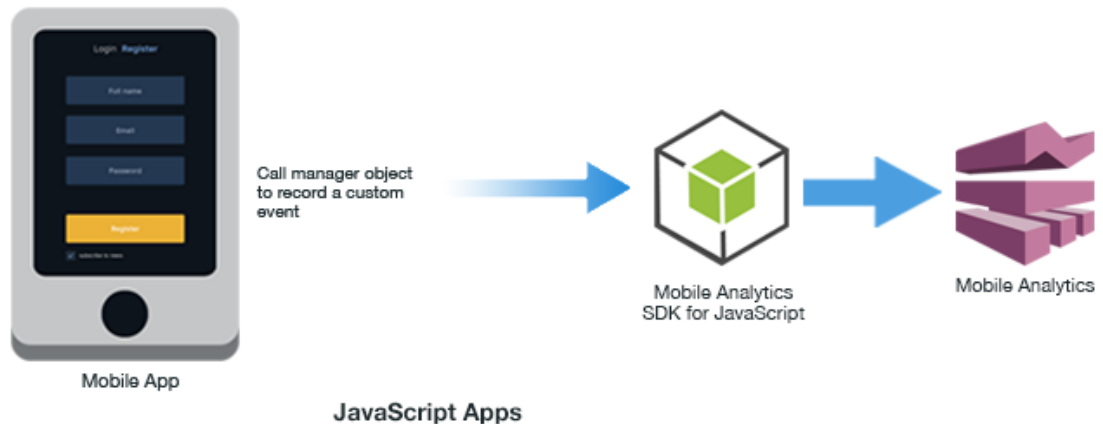
For information about creating custom events in iOS or Android apps, see:

- [AWS Mobile SDK iOS Developer Guide](#)
- [AWS SDK for iOS Reference](#)

- [AWS Mobile SDK Android Developer Guide](#)
- [AWS SDK for Android Reference](#)

Creating a Custom Event in JavaScript Apps

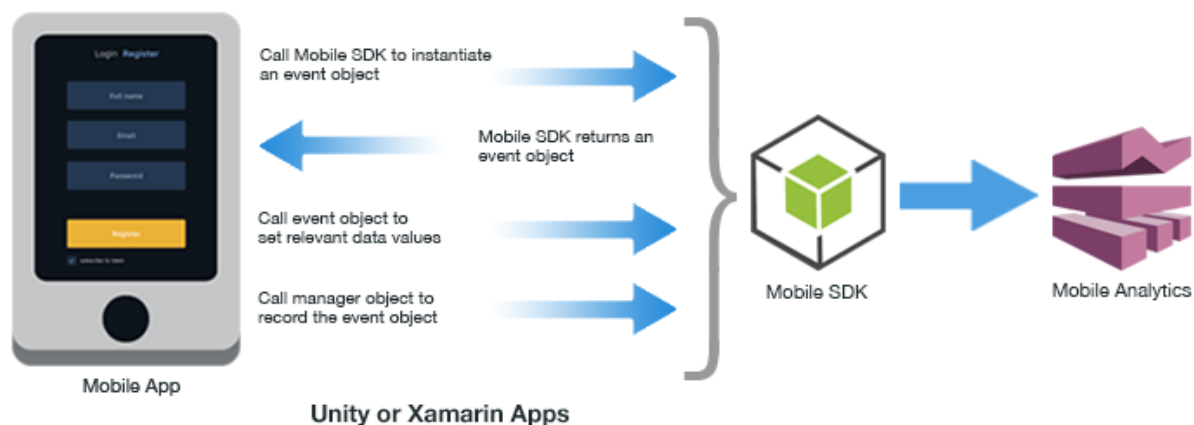
To create a custom a custom event in a JavaScript app, call the manager object to record a custom event, passing the attributes and metrics you want to capture as parameters.



For information about creating custom events in JavaScript apps, see [Mobile Analytics SDK for JavaScript](#).

Creating a Custom Event in Unity or Xamarin Apps

Custom events in Unity or Xamarin apps are created and defined using an event object the app submits to Mobile Analytics. After the app has created an event object, the app customizes the event object by [adding attributes and metrics \(p. 26\)](#) that specify data values to report. After the event object is customized, the app calls the manager object to record the event.



For information about creating custom events in Unity or Xamarin apps, see:

- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS SDK for Unity Reference](#)
- [AWS Mobile SDK Xamarin Developer Guide](#)
- [AWS SDK for Xamarin Reference](#)

Adding Attributes and Metrics

Attributes are data that provides context for the submitted event. For example, a game that submits an event to Mobile Analytics when the player collects a power-up bonus might include an attribute named for the type of bonus collected. You add attributes to an event as a collection of key-value pairs.

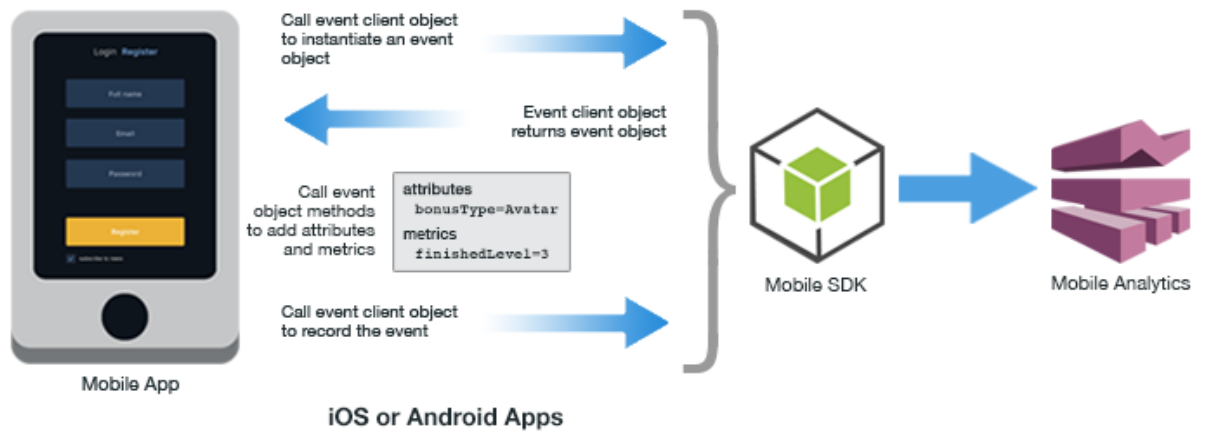
Metrics are data that gives measurable context to the event. For example, a photo-sharing app that submits an event when selected photos are uploaded might include a metric for the total amount of data being uploaded, in megabytes. You add metrics to an event as a collection of key-value pairs.

Reporting Detailed Data Points

Individual data values sent to Mobile Analytics are specified by adding one or more attributes or metrics to an event before you submit it. An event can include any combination of up to 40 total attributes and metrics. As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate. This example demonstrates how to define a custom event.

Adding Attributes and Metrics in iOS or Android Apps

Attributes and metrics are added to events in iOS and Android apps by [creating a custom event object \(p. 24\)](#) and then adding the required key-value pairs using the `addAttribute` or `addMetrics` methods.

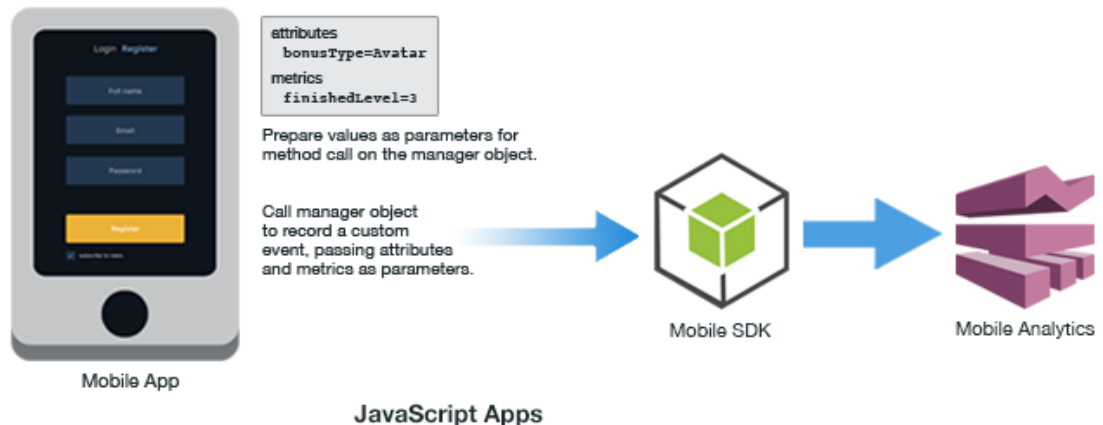


For information about adding attributes and metrics to events in iOS or Android apps, see:

- [AWS Mobile SDK iOS Developer Guide](#)
- [AWS SDK for iOS Reference](#)
- [AWS Mobile SDK Android Developer Guide](#)
- [AWS SDK for Android Reference](#)

Adding Attributes and Metrics in JavaScript Apps

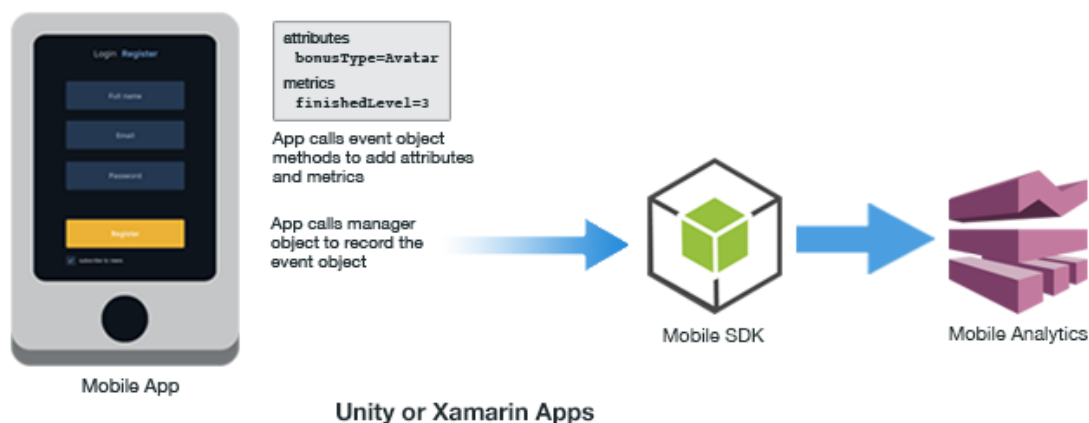
Attributes and metrics are added to [custom events in JavaScript apps \(p. 25\)](#) as parameter values.



For more information, see [Mobile Analytics SDK for JavaScript](#).

Adding Attributes and Metrics in Unity or Xamarin Apps

Attributes and metrics are added to events in Unity or Xamarin apps by creating a custom event object and then adding the required key-value pairs using the `addAttribute` or `addMetrics` methods.



For information about adding attributes and metrics to events in Unity or Xamarin apps, see:

- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS SDK for Unity Reference](#)

- [AWS Mobile SDK Xamarin Developer Guide](#)
- [AWS SDK for Xamarin Reference](#)

Submitting Events

Each app can have up to 1,500 unique custom events, up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

Submitting Events in iOS Apps

When you create event objects in an iOS app, you call the `recordEvent` method on the event client object to record those events in the device's local persistent storage. As long as there is an active session, recorded events are submitted to Mobile Analytics in a background thread. You can call the `submitEvents` method on the event client object to submit events manually. This removes data from the client file store.

For information about submitting events in iOS apps, see:

- [AWS Mobile SDK iOS Developer Guide](#)
- [AWS SDK for iOS Reference](#)

Submitting Events in Android Apps

When you create event objects in an Android app, you call the `recordEvent` method on the event client object to record those events in the device's local persistent storage. As long as there is an active session, recorded events are submitted to Mobile Analytics in a background thread. The client will store a maximum of 5 megabytes of event data. You can call the `submitEvents` method on the event client object to submit events manually. This removes data from the client file store.

For information about submitting events in Android apps, see:

- [AWS Mobile SDK Android Developer Guide](#)

- [AWS SDK for Android Reference](#)

Submitting Events in JavaScript Apps

When you create event objects in a JavaScript app, you call the `recordEvent` method on the manager object to record those events in the device's local persistent storage. You can call the `submitEvents` method on the manager object to submit events manually. This removes data from the client file store. As long as there is an active session, session, events are submitted to Mobile Analytics every 10 seconds.

For information about submitting events in JavaScript apps, see [Mobile Analytics SDK for JavaScript](#).

Submitting Events in Unity or Xamarin Apps

When you create event objects in a Unity or Xamarin app, you call the `recordEvent` method on the manager object to record those events in the device's local persistent storage. All events are submitted to Mobile Analytics in a background thread.

For information about submitting events in Unity or Xamarin apps, see:

- [AWS Mobile SDK Unity Developer Guide](#)
- [AWS SDK for Unity Reference](#)

- [AWS Mobile SDK Xamarin Developer Guide](#)
- [AWS SDK for Xamarin Reference](#)

Using the Mobile Analytics REST API

To take advantage of Mobile Analytics, your app must include code that records and submits events you want reported in the console.

App developers working on platforms supported by the AWS Mobile SDK will likely use the APIs provided for their platforms. You can use the REST API to integrate with Mobile Analytics without incorporating the AWS Mobile SDK in your app. The REST API lets you submit events from a back-end service.

The topics in this section cover key Mobile Analytics concepts, including Mobile Analytics event types and common tasks you perform with the REST API.

Topics

- [Identifying the App in Mobile Analytics \(p. 30\)](#)
- [Managing Sessions \(p. 31\)](#)
- [Generating Mobile Analytics Events \(p. 34\)](#)
- [Adding Attributes and Metrics \(p. 37\)](#)
- [Providing a Client Context \(p. 39\)](#)
- [Submitting Events \(p. 39\)](#)
- [Signing Requests \(p. 40\)](#)

Identifying the App in Mobile Analytics

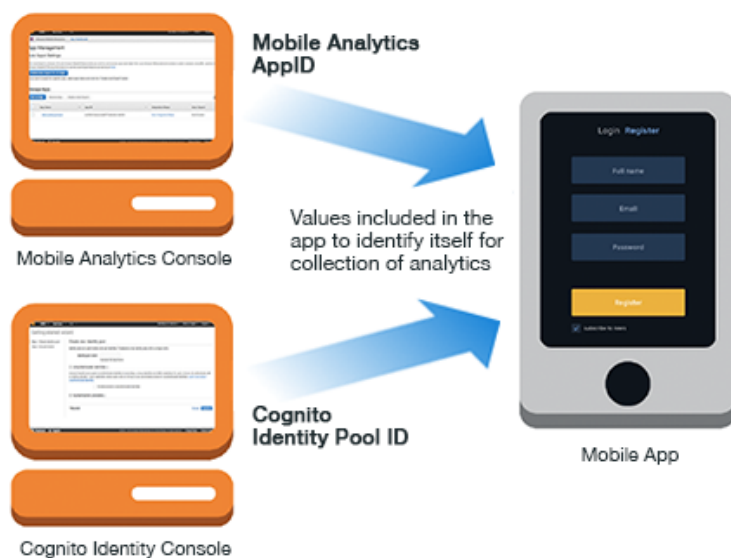
Before you identify your app, you must:

- [Add your app in the Mobile Analytics console. \(p. 3\)](#)
- [Create a Cognito identity pool](#) or use an identity pool created for you by Mobile Analytics.

Tying Analytics to Users

Before Mobile Analytics can collect and present any analytic data about your application, you must identify your application to Mobile Analytics. There are two pieces of information you need so your application can identify itself with Mobile Analytics:

- The Amazon Cognito identity pool created to authenticate users of your application.
- The AppID created in the Mobile Analytics console when you added your app to Mobile Analytics.



When you use the Mobile Analytics REST API, your app provides the AppID as part of the [client context header \(p. 39\)](#) you include when calling the PutEvents action. Your app provides the Cognito identity pool ID as part of initializing Cognito Identity.

Managing Sessions

Before you manage sessions in your app, you must:

- [Add your app in the Mobile Analytics console \(p. 3\)](#)
- [Create a Cognito identity pool](#) or use an identity pool created for you by Mobile Analytics.

Users Engage in Sessions

Mobile Analytics reports app usage data based on activity captured in sessions. A *session* is a period of time during which the application is active on the device. A session begins when an app is launched or brought to the foreground, and ends when the app is terminated or goes to the background. To accommodate brief interruptions, like a text message, an inactivity period of up to 5 seconds does not counted as a new session.

Mobile Analytics tracks sessions and events submitted during each session to generate the reports displayed in the console. Total Daily Sessions shows the number of sessions your app has each day. Average Sessions per Daily Active User shows the mean number of sessions per user per day.

Starting a Session

When your app starts, a session tracked by Mobile Analytics should begin. The code that initializes the app calls the PutEvents action, sending a session start event.



A session start event has an eventType of "_session.start" as shown in the following example.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}}
```

```
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_session.start",
      "session": {
        "startTimestamp": "<ISO 8601 date>",
        "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
        <optional>
      },
      "metrics": {
        <optional>
      }
    }
  ]
}
```

Stopping a Session

When your app closes or is sent to the background, it should end the current session. To end a session, the application code that responds to notification to shut down or pause calls the PutEvents action, sending a session end event.



A session end event has an eventType of "_session.stop" as shown in the following example.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"client_id","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{},"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "eventType": "_session.stop",
  "timestamp": "2014-07-09T03:26:38.719Z",
  "session": {
    "id": "<session id>",
    "duration": 497516,
    "startTimestamp": "2014-07-09T03:17:20.041Z",
    "stopTimestamp": "2014-07-09T03:25:37.557Z"
  },
  "attributes": {},
  "metrics": {}
}
```

Generating Mobile Analytics Events

Before generating analytics events in your app, you must:

- [Add your app in the Mobile Analytics console \(p. 3\)](#)
- [Create a Cognito identity pool](#) or use an identity pool created for you by Mobile Analytics.
- [Start a session \(p. 31\)](#)

Types of Analytics Events

Information about user engagement in your app is sent to the Mobile Analytics service using events. Generally speaking, there are two types of mobile analytics events:

- Standard events
- Custom events

Standard events include those you send to start or stop a session. They also include specialized events used to report on monetization activities in the app, such as in-app purchases. Custom events are those you create to monitor activities specific to your app, such as completing a level in a game, posting to social media, or setting particular app preferences. An app can have up to 1,500 unique custom events.

Session Events

Standard events include those that allow the app to [start and end a session \(p. 31\)](#) in Mobile Analytics. Sessions enable Mobile Analytics to provide the analytics data in the console reports.

To start a session, the app calls the PutEvents action with a session start event. A session start event has an eventType of `"_session.start"`. To end a session, the app sends an HTTP PutEvents request with a session end event. A session end event has an eventType of `"_session.stop"`.

Defining a Monetization Event

When a user of your app makes a purchase, the code that handles the purchase calls the PutEvents action to send a monetization event.



A monetization event has an eventType of `"_monetization.purchase"`, as shown in the following example. It demonstrates how to monetize an event with a price that does not include the currency symbol.

Amazon Mobile Analytics User Guide

Defining a Monetization Event

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{},"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_monetization.purchase",
      "session": {
        "startTimestamp": "<ISO 8601 date>",
        "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
        "_currency": "<ISO 4217 currency code>",
        "_product_id": "<User specified string>"
      },
      "metrics": {
        "_quantity": <Purchase quantity, defaults to 1>,
        "_item_price": <Decimal price>
      }
    }
  ]
}
```

This example demonstrates how to monetize an event with a price that includes the currency symbol. You can use this example with a formatted price.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{},"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_monetization.purchase",
      "session": {
        "startTimestamp": "<ISO 8601 date>",
        "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
        "_currency": "<ISO 4217 currency code>",
        "_product_id": "<User specified string>"
      },
      "metrics": {
        "_quantity": <Purchase quantity, defaults to 1>,
        "_item_price": <Decimal price>
      }
    }
  ]
}
```

```
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_monetization.purchase",
      "session": {
        "startTimestamp": "<ISO 8601 date>",
        "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
        "_item_price_formatted": "<Price prefixed with currency symbol ($1.99)>",
        "_product_id": "<User specified string>"
      },
      "metrics": {
        "_quantity": <Purchase quantity, defaults to 1>
      }
    }
  ]
}
```

Creating a Custom Event

In addition to the standard events, you can also define your own custom events to report on types of interaction specific to your app, such as finishing levels of a game.



You assign an eventType to your custom event. As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate. This example demonstrates how to define a custom event.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mo
```



```
bileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{},"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "sampleEvent",
      "timestamp": "2014-07-09T03:17:20.041Z",
      "session": {
        "id": "<Session id>",
        "startTimestamp": "2014-07-09T03:15:31.041Z"
      },
      "attributes": {},
      "metrics": {}
    },
    {
      "eventType": "otherEvent",
      "timestamp": "2014-07-09T03:17:42.772Z",
      "session": {
        "id": "<Session id>",
        "startTimestamp": "2014-07-09T03:15:31.041Z"
      },
      "attributes": {
        "customAttribute": "someValue"
      },
      "metrics": {
        "count": 1
      }
    }
  ]
}
```

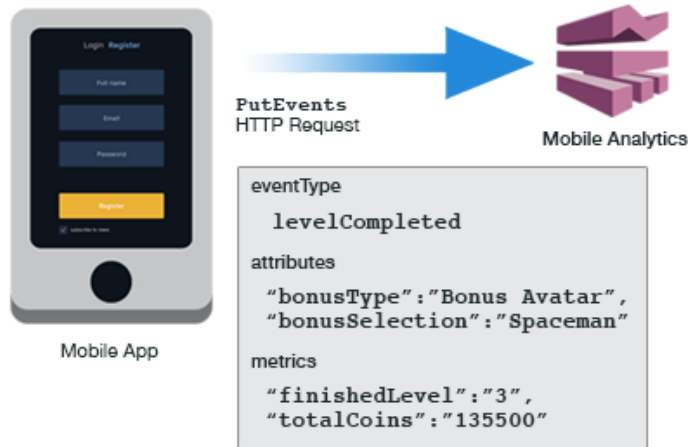
Adding Attributes and Metrics

Before you add attributes or metrics to events in your app, you must:

- [Add your app in the Mobile Analytics console \(p. 3\)](#)
- [Create a Cognito identity pool](#) or use an identity pool created for you by Mobile Analytics.
- [Start a session \(p. 31\)](#)
- [Create an event \(p. 34\)](#)

Reporting Detailed Data Points

You specify individual data values sent to Mobile Analytics by adding one or more attributes or metrics to an event before you submit it. An event can include any combination of up to 40 total attributes and metrics.



As a best practice, we recommend that names given to custom event names be broad and those given to attributes or metrics be specific. For example, using "Item Bought" instead of "Item XYZ" as the custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate.

Adding Attributes

Attributes are data that provides context for the submitted event. For example, a game that submits an event to Mobile Analytics when the player collects a power-up bonus might include an attribute named for the type of bonus collected.

You add attributes to an event as a collection of key-value pairs. When you use the REST API, you add each key-value pair to the attributes section of the event and then submit the event with the PutEvents action. For example:

```
"attributes": {
  "bonusType": "Bonus Avatar",
  "bonusSelection": "Spaceman"
}
```

Adding Metrics

Metrics are data that gives measurable context to the event. For example, a photo-sharing app that submits an event when selected photos are uploaded might include a metric for the total amount of data being uploaded, in megabytes.

You add metrics to an event as a collection of key-value pairs. When you use the REST API, you add each key-value pair to the metrics section of the event and then submit the event with the PutEvents action. For example:

```
"metrics": {  
  "finishedLevel": "3",  
  "totalCoins": "135500"  
}
```

Providing a Client Context

Before you use a client context to submit events from your app, you must:

- [Add your app in the Mobile Analytics console \(p. 3\)](#)
- [Create a Cognito identity pool](#) or use an identity pool created for you by Mobile Analytics.

Giving Details of Application Context

Data in a client context provides information about the client interaction with an application service; it describes the app and environment in which it runs. When your mobile app communicates with Mobile Analytics, it must provide the client context for the application and device. To do this, create a JSON-formatted string with the data to [include as a header \(p. 71\)](#) when using the PutEvents action.

Client context contains at least two sets of properties:

- Client properties
- Environment properties

In addition to these two sets of properties, a client context can also contain custom values required to provide context about the app or environment.

The client context is where you provide the values required by Mobile Analytics to identify your app. In the services section, you must include `mobile_analytics` as a key with the AppID you generated for your app in the Mobile Analytics console as the value.

Submitting Events

Before you submit events from your app, you must:

- [Add your app in the Mobile Analytics console \(p. 3\)](#)
- [Create a Cognito identity pool](#) or used an identity pool created for you by Mobile Analytics.
- [Start a session \(p. 31\)](#)
- [Create an event to submit \(p. 34\)](#)
- [Add relevant attributes and metrics to events \(p. 37\)](#)

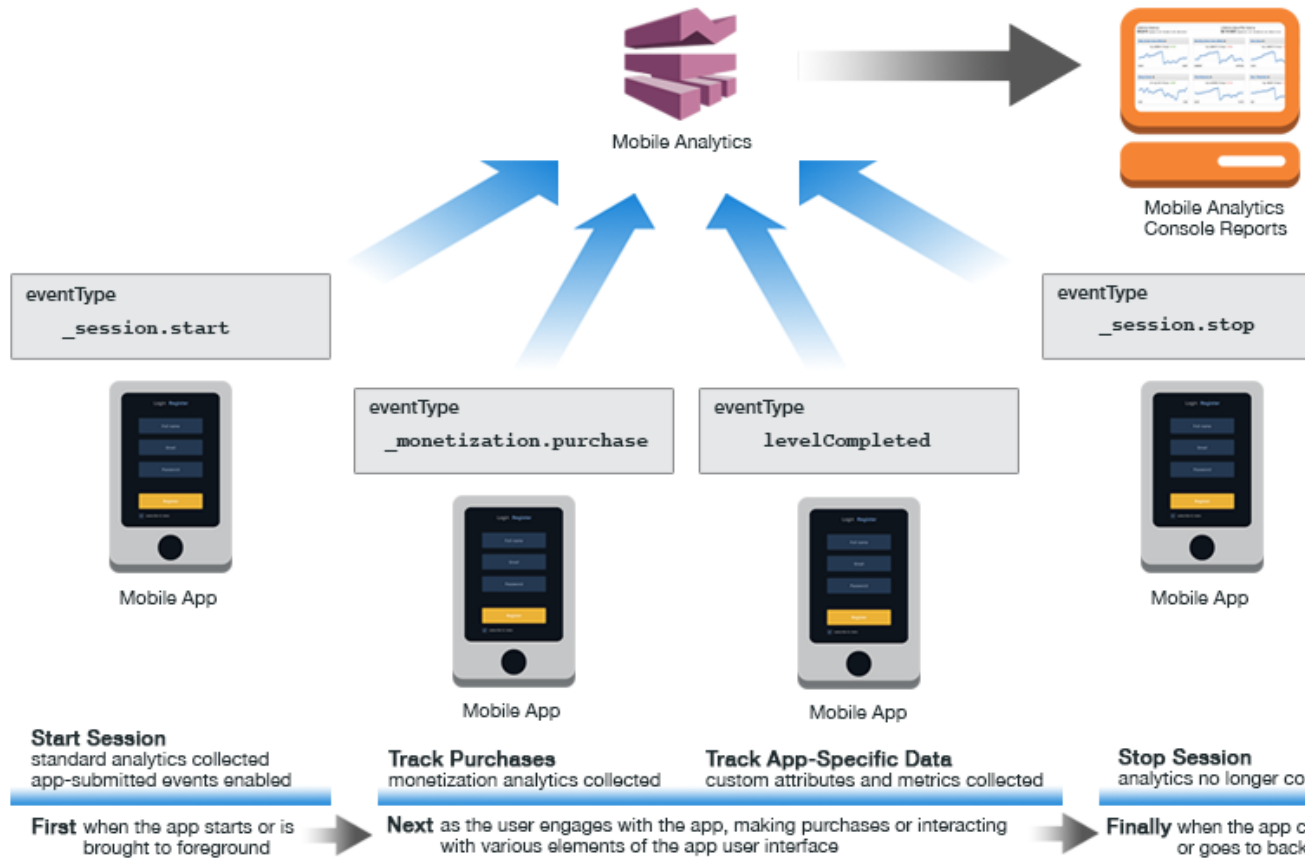
Event-Based Interaction

The interaction between a mobile app and Mobile Analytics takes place through a single PutEvents action in the REST API. You use the PutEvents action to:

- Start and stop sessions to collect the standard analytics displayed in the console reports.

- Send monetization data.
- Collect custom analytics specific to your app.

The following diagram shows how a mobile app sends events to Mobile Analytics at different points in the execution of the app to enable the collection of data used to produce the analytics reports.



You can have up to 1,500 unique custom events per app, up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

Signing Requests

The method to use to sign your request depends on where the request originates. If the request comes from a server, use AWS Identity and Access Management. If the request comes from a mobile device, use Amazon Cognito or IAM.

Both IAM and Amazon Cognito support signature version 4. For more information, see [Signature Version 4 Signing Process](#).

Mobile Analytics Console Reports Overview

You can view or download Mobile Analytics reports from the AWS Management Console. Reports are available for metrics on active users, sessions, retention, in-app revenue, and custom events. Go to the Mobile Analytics console at <https://console.aws.amazon.com/mobileanalytics/home>.

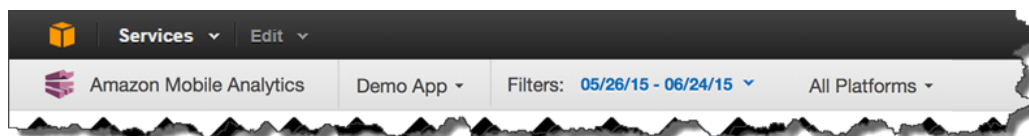
Topics

- [Toolbar \(p. 41\)](#)
- [Using the Console Reports \(p. 42\)](#)
- [Overview Tab \(p. 43\)](#)
- [Active Users Tab \(p. 44\)](#)
- [Sessions Tab \(p. 45\)](#)
- [Revenue Tab \(p. 46\)](#)
- [Retention Tab \(p. 47\)](#)
- [Custom Events Tab \(p. 48\)](#)
- [Working with Charts \(p. 49\)](#)

Toolbar

You can use the controls on the toolbar in the Mobile Analytics console to filter on the following :

- **Application list** – Provides a list of all apps that have submitted data to Mobile Analytics.
- **Date range** – Allows you to specify date ranges for the data to be displayed in the reports. By default, the date range is the last 30 days.
- **Platforms** – Lets you select which platform data to display in the console reports. You can choose to show data from all platforms together or data from a specific platform only.



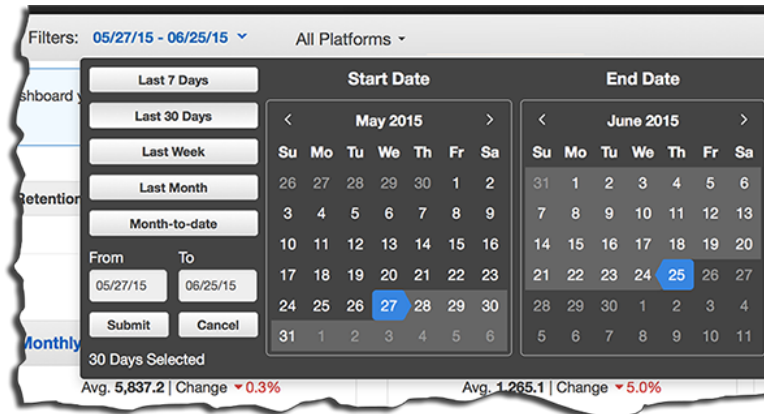
Using the Console Reports

After you have integrated Mobile Analytics into your apps, they will relay data about user engagement to the service. You can view the currently compiled data in the console. Data reported by your apps is incorporated into the compiled data about 60 minutes after it was sent.

To view the current reports for an app, select it from the applications drop-down menu in the toolbar. The reports are organized into a set of tabs, each of which is described in this section.

Filtering Data Shown by Date Range

You can change the date range used to filter the data shown in the reports.

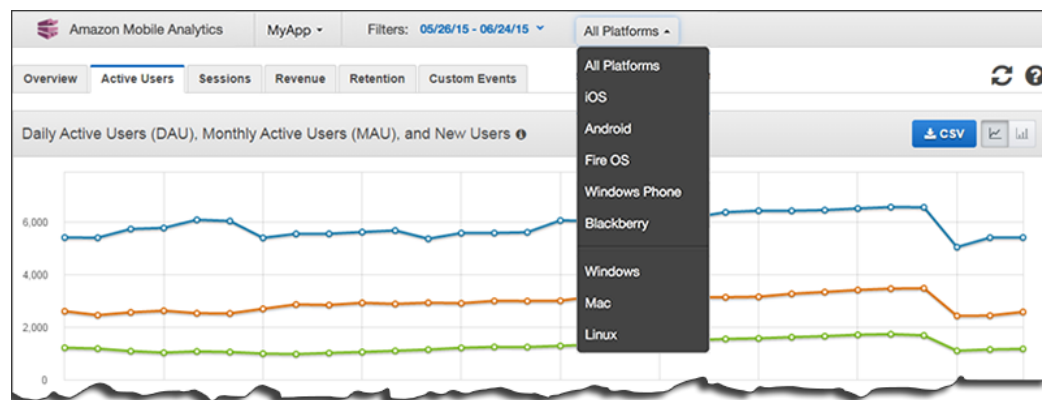


To filter the data displayed in the console

1. Choose the **Filter** menu in the toolbar.
2. Choose the dates you want reflected in the console reports. You can choose one of the preset date range options or enter dates in the **From** and **To** fields.
3. Choose **Submit** to apply the date filter to the console reports.

Selecting the Platform Displayed

You can view the data for all platforms supported by your app or for specific platforms only.



To change the platforms displayed in the reports

1. Choose the **All Platforms** menu in the toolbar.
2. From the drop-down list, choose **All Platforms** to display data reported by the app on all platforms or choose any of the following platforms:
 - iOS
 - Android
 - Fire OS
 - Windows Phone
 - Blackberry
 - Windows (browser-based apps)
 - Mac (browser-based apps)
 - Linux (browser-based apps)

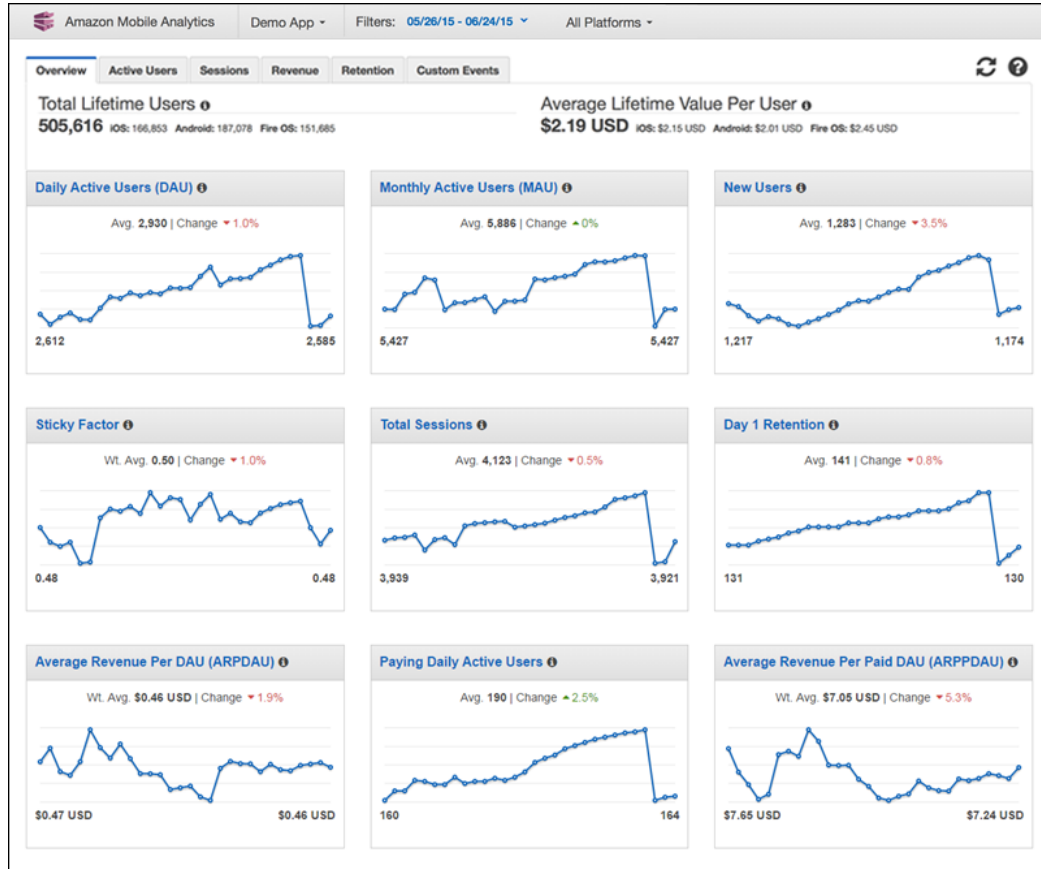
Browser-based apps that run under one of the mobile operating systems are reported under the appropriate operating system.

Overview Tab

The **Overview** tab displays at-a-glance summaries of:

- **Daily Active Users (DAU)** – Users who used your app on a particular day.
- **Monthly Active Users (MAU)** – Users who used your app in the previous 30 days.
- **New Users** – New users who used your app on a particular day.
- **Sticky Factor** – Fraction of monthly users who used your app on a particular day (DAU/MAU).
- **Total Sessions** – Number of times your app was used on a particular day.
- **Day 1 Retention** – Percentage of new users who used your app on a specific day and then again the following day.
- **Average Revenue Per Paid Daily Active User (ARPPDAU)** – Gross revenue for in-app items per daily active user who purchased in-app items.

Amazon Mobile Analytics User Guide Active Users Tab

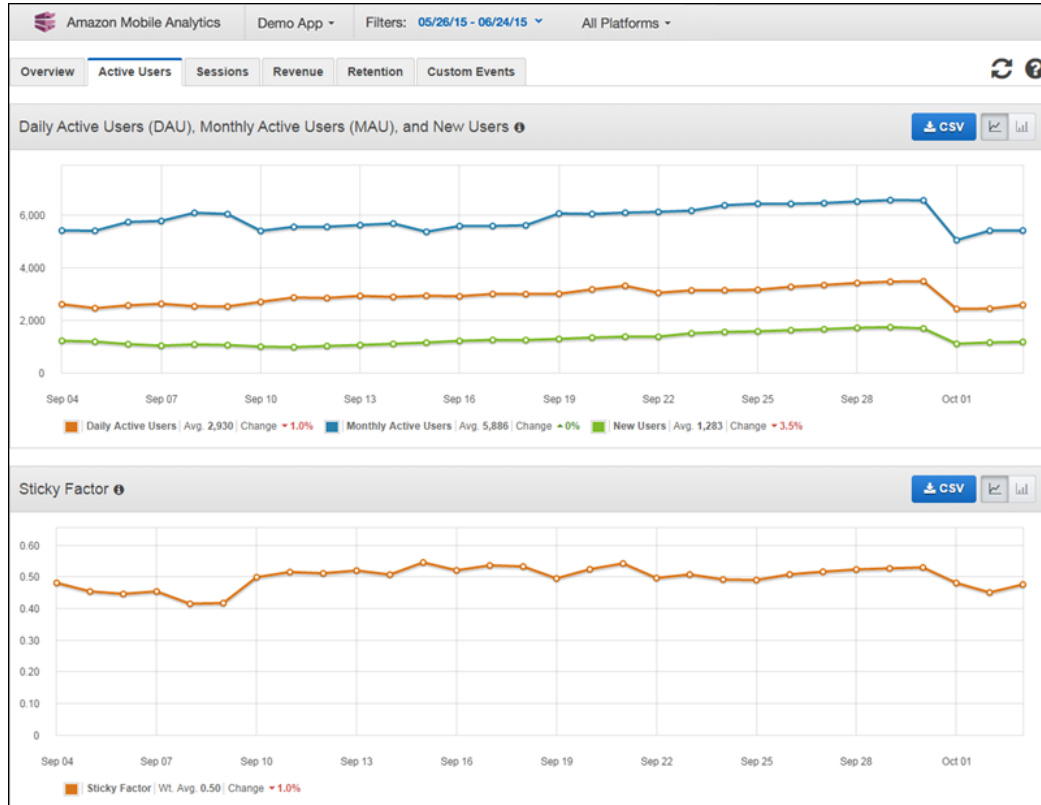


Active Users Tab

The **Active Users** tab displays trend charts for the following:

- **Daily Active Users (DAU)** – Users who used your app on a particular day.
- **Monthly Active Users (MAU)** – Users who used your app in the previous 30 days.
- **New Users** – New users who used your app on a particular day.
- **Sticky Factor** – Fraction of monthly users using your app on a particular day. For example, a Sticky Factor of .25 means that on a particular day, 25% of your users from the previous 30 days used your app.

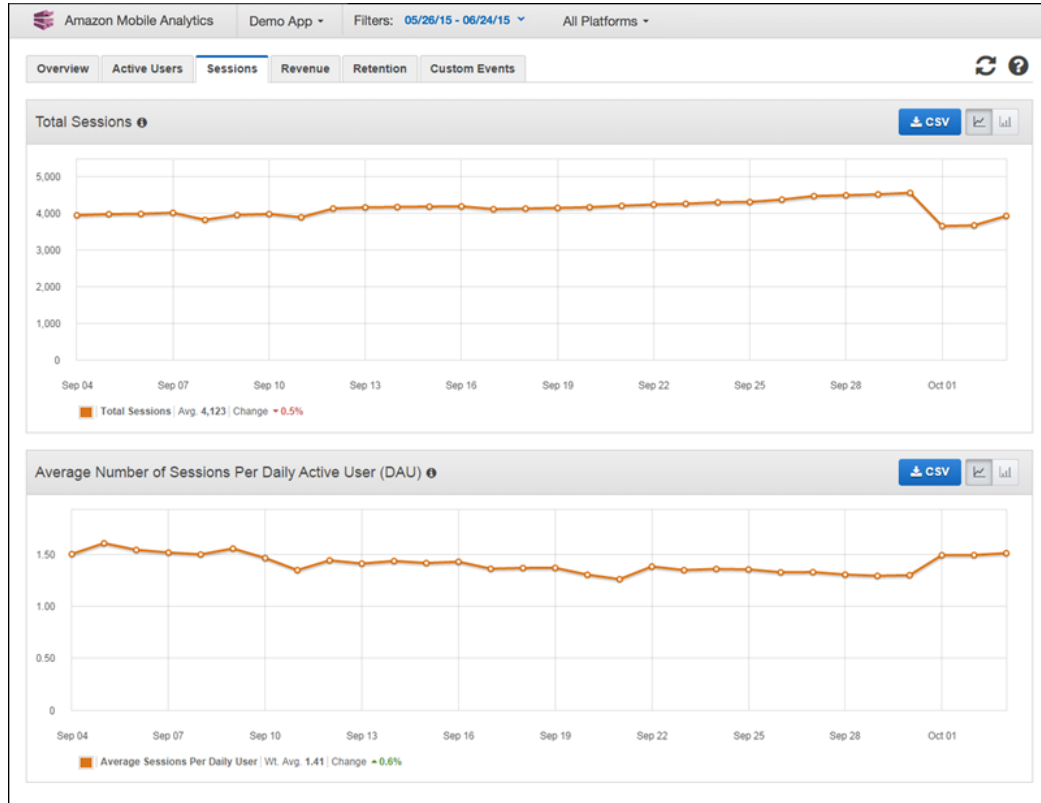
Amazon Mobile Analytics User Guide Sessions Tab



Sessions Tab

The **Sessions** tab displays trend charts for the following:

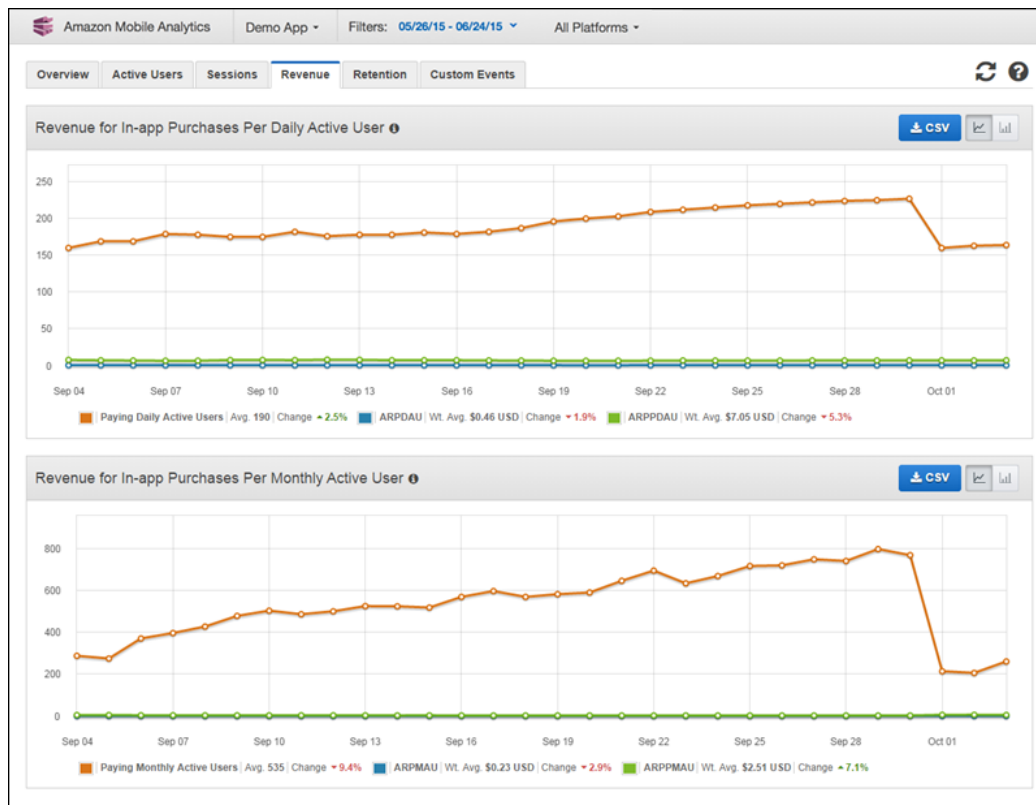
- **Total Sessions** – Number of times your app was used on a particular day.
- **Average Number of Sessions Per Daily Active User** – Average of the number of sessions per daily active user who used your app on a particular day.



Revenue Tab

The **Revenues** tab displays trend charts for the following:

- **Average Revenue Per Daily Active User (ARPPDAU)** – Gross revenue for in-app items per daily active user.
- **Average Revenue Per Paying Daily Active User (ARPPDAU)** – Gross revenue for in-app items per daily active user who purchased in-app items.
- **Paying Daily Active Users** – Users who bought an in-app item on a particular day.



Gross revenue does not include taxes, product returns, reimbursements, subscriptions or revenue from the purchase of the app.

Revenue for in-app purchases per monthly active user:

- **Average Revenue Per Monthly Active User (ARPMUA)** – The gross revenue divided by the number of users who bought an item from your app in the last 30 days.
- **Average Revenue Per Paid Daily Active User (ARPPMAU)** – Gross revenue for in-app items per monthly active user who purchased in-app items.
- **Paying Monthly Active Users** – Users who bought an in-app item in a particular month.

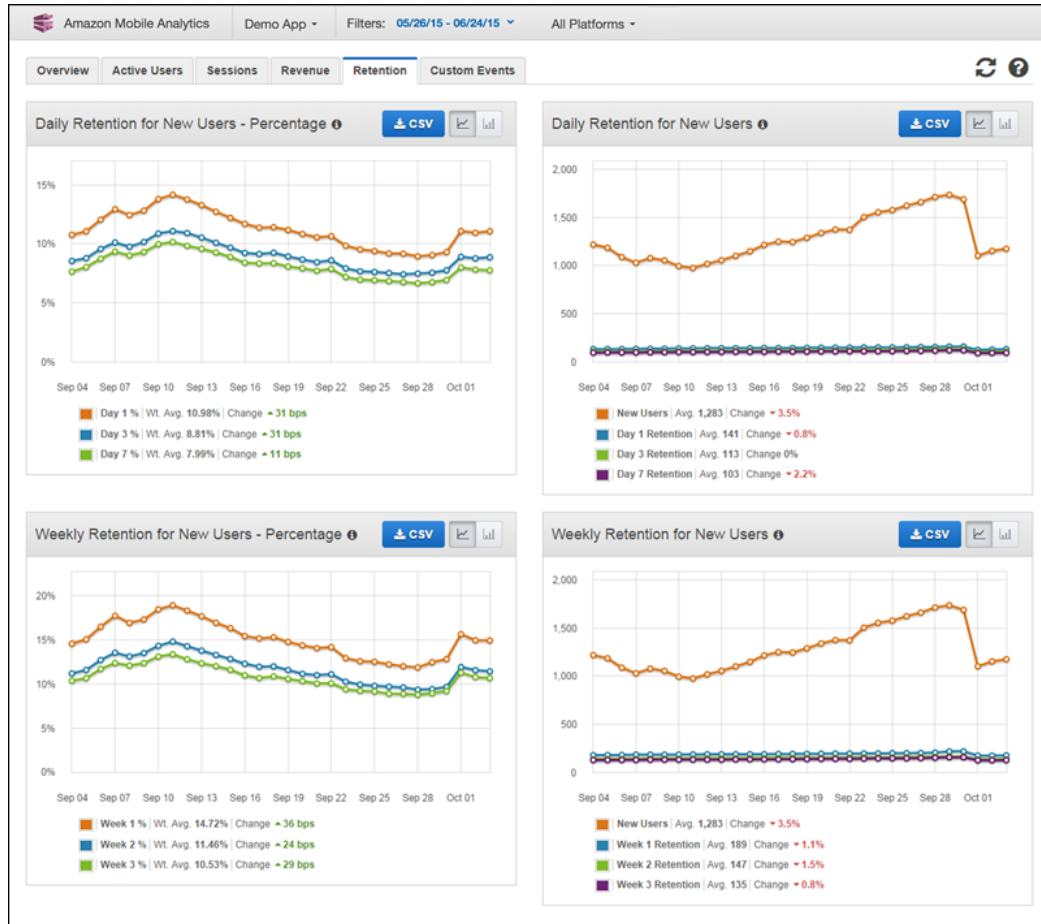
Gross revenue does not include taxes, product returns, reimbursements, subscriptions or revenue from the purchase of the app.

Retention Tab

The **Retention** tab displays trend charts for the following:

- **Daily Retention for New Users—Percentage** – Percentage of new users who used your app on a specific day and then again on the following day, the third day, and the seventh day.
- **Weekly Retention for New Users—Percentage** – Percentage of users who used your app on a specific day and then at least once during the next 7 days inclusive (1-week retention); between 8 and 14 days inclusive (2-week retention); and between 15 and 21 days inclusive (3-week retention).
- **Daily Retention for New Users—Count** – New users who used your app on a specific day and then again on the following day, the third day, and the seventh day. This chart also shows the number of new users who used your app on a particular day.

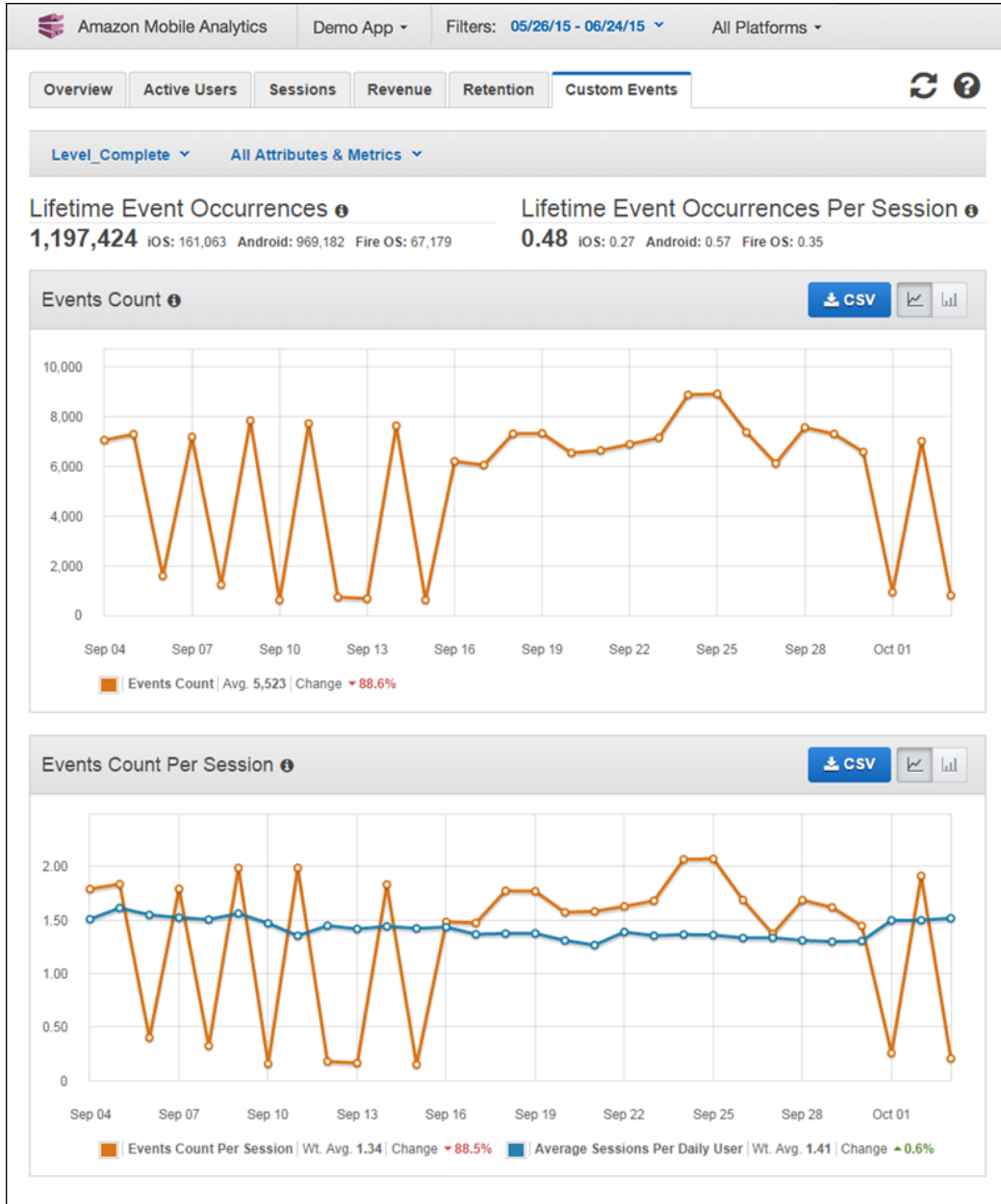
- **Weekly Retention for New Users—Count** – New users who used your app on a specific day and then at least once during the next seven days inclusive (1-week retention); between 8 and 14 days inclusive (2-week retention); and between 15 and 21 days inclusive (3-week retention). This chart also shows the number of new users who used your app on a particular day.



Custom Events Tab



The **Custom Events** tab displays charts for metrics that you define.


- **Lifetime Event Occurrences** – Aggregate number of custom events received for your app to date.
- **Lifetime Event Occurrences Per Session** – Average number of custom events received per session to date.
- **Events Count Per Session** – Average number of custom events received per session.
- **Breakdown of Metric Values** – Average, minimum, and maximum of metric values.



Working with Charts

Many of the tabs on the Mobile Analytics console include controls for displaying or downloading data:

	Click to download the currently displayed data to a comma-separated values file.
	Click to display the current data as a line chart.

	Click to display the current data as a bar chart.
---	---

Amazon Mobile Analytics Auto Export

Amazon Mobile Analytics is a feature that lets you easily collect, visualize, and understand app usage data at scale. Mobile Analytics Auto Export can export event data from your apps to an [Amazon S3](#) bucket or an [Amazon Redshift](#) cluster through an AWS CloudFormation stack created in your account. This data makes it possible for you to analyze app usage in detail and combine this information with other business intelligence data.

Amazon S3 buckets provide secure, durable, highly scalable object storage. Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. AWS CloudFormation provides an easy way to create and manage a collection of related AWS resources.

If you have not already done so, follow the steps in the [Mobile Analytics Getting Started Guide](#) to create an app and generate events.

Getting Started with Auto Export to Amazon Redshift or Amazon S3

The S3 bucket and Amazon Redshift cluster must be in the US East (N. Virginia) region.

Step 1: Go to Manage Apps

From the Amazon Mobile Analytics console, click the app drop-down menu, and choose **Manage Apps**.

Step 2: Launch the Auto Export Wizard

Choose the apps for which you would like to enable Auto Export, and choose the **Enable Auto Export** button. Alternatively, you can select the **Enable Auto Export for All Apps** check box to export all apps.

Step 3: Choose Export to Amazon Redshift + Amazon S3

Under **Amazon S3**, choose the **Get Started** button.

Step 4: Configure an Amazon S3 Bucket

Choose the Create a new S3 bucket button or use the **Select an existing bucket** drop-down list to choose an existing bucket. Regardless of which option you choose, you will own the S3 bucket and will be responsible for all charges incurred. For more information, see [S3 Pricing](#). You can go to the [S3 console](#) at any time to delete the data in your S3 bucket.

Step 5: Configure S3 and S3 to Amazon Redshift Permissions

Mobile Analytics uses AWS Identity and Access Management (IAM) roles to securely access your S3 bucket through a cross account role. Choose the Select/Create Role button. By default, Mobile Analytics creates a new IAM role and policy with write access to the S3 bucket you selected in the previous step. We recommend you use the default setting of creating a new IAM role. If you want to specify a different role or policy, choose **View Details**, and choose the IAM role or policy. If you choose an existing policy, be sure it provides access to the S3 bucket you will be exporting to. Choose the Allow button to allow access to your Amazon S3 bucket.

Choose the **Create Export** button. Upon confirmation, your Auto Export setup will be complete. It can take up to one hour for events to appear in your S3 bucket.

In addition to S3, Auto Export to Amazon Redshift requires an EC2 IAM role. This role is used by the EC2 instance created in your account to read data from S3, and communicate with other AWS resources, such as CloudWatch and Amazon Redshift. Choose the Select/Create Role button. By default, Mobile Analytics creates a new IAM role and policy that allows an EC2 instance in your account to access the S3 bucket you selected in the previous step as well as other AWS resources. We recommend you use the default setting of creating a new IAM role. If you want to specify a different role or policy, choose **View Details** and choose the IAM role or policy. If you choose an existing policy, be sure it provides the following:

- Read and write access to the S3 bucket you will be exporting to.
- Access to "cloudwatch:PutMetricData".
- Access to "logs:*".

Choose the Allow button to use the EC2 role.

Step 6: Configure Auto Export to Use Amazon Redshift

Amazon Mobile Analytics uses CloudFormation to create a new Amazon Redshift cluster and EC2 instance in your account. Enter a password for the "master" user, which will have full administrative permissions to your Amazon Redshift cluster. Keep this password safe. Enter a password for the "eventreader" user, which will have read-only permissions to your Amazon Redshift cluster. Distribute these credentials to anyone who needs read-only access to your data. To prevent accidental changes, we recommend you use the `eventreader` user when you query Amazon Redshift .

If your app records custom event attributes or event metrics, type the name for each attribute or metric, and press **Enter**. Alternatively, you can paste a newline-separated or comma-separated list of values.

The EC2 instance created in your account can record metrics, such as the length of time to load events into Amazon Redshift or whether the load was successful. In addition, the EC2 instance copies all logs to CloudWatch Logs. You can enable [CloudWatch Metrics and Logs](#) by selecting the **Enable CloudWatch Metrics and Logs** check box.

Choosing **Advanced Options** allows you to specify settings for connecting to your Amazon Redshift cluster and EC2 instance:

- **IP access rules** - By default, your Amazon Redshift cluster and EC2 instance will be accessible from your machine only. To make your Amazon Redshift cluster and EC2 instance available to all internet addresses, choose **Accessible from any IP Address**. You can also specify a custom Classless Inter-Domain Routing (CIDR) rule to allow access to addresses from within your organization. For more information about CIDR notation, see [Classless Inter-Domain Routing](#) on Wikipedia.
- **Port number** - The port number used by Amazon Redshift to communicate.
- **Password for the etl_user account used for loading your data from S3 to Amazon Redshift** - This is the write-only password used by the EC2 instance to load data into Amazon Redshift. We recommend you use the generated password.
- **An EC2 key pair used for connecting to the EC2 instance using SSH** - If you want to connect to the EC2 instance by using SSH (for example, to do custom log pulling), you must specify an EC2 key pair. You can create one in the EC2 console. You cannot change the key pair after the EC2 instance has started.

Each of the advanced options has a default value, so you do not need to specify a value unless you want a value different than the default. You can click **Advanced Options** to see the default values.

Choose the Create Export button. It can take up to 15 minutes for your Amazon Redshift cluster to be provisioned.

Note

The EC2 instance will run continuously (and incur charges) regardless of whether new event data is being sent by your app.

Connecting to Amazon Redshift

Amazon Redshift provides multiple ways to connect to a Amazon Redshift cluster, including programmatic access via a JDBC connection, or querying Amazon Redshift directly on your workstation using SQLWorkbench/J. For more information, see [Connecting to a Amazon Redshift Cluster](#).

When logging into your Amazon Redshift cluster, use the Amazon Redshift connection string which is located in the Amazon Redshift console on your cluster's configuration page. For more information, see [Configuring Amazon Redshift Connections](#). Then use the `eventreader` user name and the `eventreader` password you specified in the Auto Export configuration to connect to the cluster.

Managing Auto Export

After you set up Auto Export, you may want to enable export for additional apps or disable Auto Export for an app.

Enabling Auto Export for Additional Apps

After you have set up Auto Export, you can have data from other apps automatically exported. Go to the App Management page, choose the apps to enable, choose the **Configure Auto Export to S3** or the **Configure Auto Export to Amazon Redshift + S3** drop-down list (depending on how you configured Auto Export), and choose **Enable Auto Export**. The apps will be added to your Auto Export configuration and data will begin flowing to your S3 bucket or Amazon Redshift cluster.

Note

Only one Auto Export configuration can be used at one time. If you chose the Amazon Redshift + Amazon S3 option, all of your apps configured to use Auto Export will write event data to Amazon Redshift and Amazon S3.

Disabling Auto Export

You can disable Auto Export for one or more apps. From the App Management page, choose the apps to disable, choose the **Configure Auto Export to S3** or the **Configure Auto Export to Redshift + S3** drop-down list (depending on how you configured Auto Export), and choose **Disable Auto Export**. After you've confirmed the action, data for the selected apps will stop flowing to your S3 bucket. Disabling Auto Export will not delete your S3 bucket, the data contained in it, your Amazon Redshift cluster, or your EC2 instance. You will continue to incur charges until you delete or terminate each resource.

To remove your Amazon Redshift cluster and EC2 instance so that you no longer incur charges, go to the [CloudFormation console](#) and delete the stack Mobile Analytics created on your behalf. This operation will delete all resources and cannot be reversed.

To remove data in S3, go to the [S3 console](#) and remove it manually.

Exported Event JSON

Example

The following JSON shows the data written for a hypothetical event that contains all elements and attributes. Not all elements are valid for all event types.

```
{
  "event_type": "hypothetical_event",
  "event_timestamp": 1415390144920,
  "arrival_timestamp": 1415390151679,
  "event_version": "3.0",
  "application": {
    "app_id": "a376fb15e8d9460fab172014a1236e25",
    "package_name": "com.amazon.exampleapp",
    "version_name": "1.0",
    "version_code": "1",
    "title": "Example App",
    "cognito_identity_pool_id": "us-east-1:156b8012-aaaa-4da9-9c44-a39198aebf7b",

    "sdk": {
      "name": "aws-sdk-android",
      "version": "2.1.1"
    }
  },
  "client": {
    "client_id": "47b63bb3-aaaa-bbbb-9ab7-2df4f1bccc82",
    "cognito_id": "us-east-1:4cbe503d-dd92-4224-8193-f3ba95cbec8f"
  },
  "device": {
    "model": "KFTT",
    "make": "Amazon",
    "platform": {
      "name": "ANDROID",
      "version": "4.0.3"
    },
    "locale": {
      "code": "en_US",
      "language": "en",
      "country": "US"
    }
  },
  "session": {
    "session_id": "f1bccc82-20141107-193640206",
    "start_timestamp": 1415389000206,
    "stop_timestamp": 1415389000206,
  },
  "monetization": {
    "transaction": {
      "transaction_id": "ABCDABCD-ABCD-ABCD-ABCD-ABCDABCDABCD",
      "store": "Amazon",
      "item_id": "com.yourcompany.exampleitem",
      "quantity": 1,
      "price": {
        "reported_price": "$0.99",
        "amount": 0.99,
        "currency": {
          "code": "USD",
          "symbol": "$"
        }
      }
    }
  }
}
```

Amazon Mobile Analytics User Guide Exported Event JSON

```
    }  
  },  
  "attributes": {  
    "levelName": "Level 5",  
    "playerClass": "Wizard"  
  },  
  "metrics": {  
    "score": 100400,  
    "timeToCompleteMinutes": 3.5  
  }  
}
```

JSON Event Schema Data

Attribute	Description	
amount	The numerical price value. Used only in a monetization event.	
app_id	The app ID that was used to identify the app. This ID was assigned on the App Management page of the console.	
arrival_timestamp	The time when the event was received by Amazon Mobile Analytics servers, expressed as an epoch time with milliseconds.	
client_id	Used to correlate events from the same app instance (app on a specific device).	
code	The locale code.	
code	Currency code for the purchase. Used only in a monetization event.	
cognito_id	If Amazon Cognito was used, this is the Cognito client ID. If not using Amazon Cognito, this attribute will not exist.	
cognito_identity-pool_id	If Amazon Cognito was used, this is the Cognito identity pool ID. If not using Amazon Cognito, this attribute will not exist.	
country	The country part of the locale.	
event_timestamp	The time the event occurred, expressed as an epoch time with milliseconds.	

Amazon Mobile Analytics User Guide
Exported Event JSON

Attribute	Description	
event_type	The name that was specified when the event was created. Events such as <code>_session.start</code> and <code>_session.stop</code> are generated by the system.	
event_version	Version of the schema of the event.	
item_id	An identifier for the item purchased. Used only found in a monetization event.	
language	The language part of the locale.	
make	The make of the device.	
model	The model of the device.	
name	The name of the Amazon Mobile Analytics SDK used by the app.	
name	If the Amazon Mobile SDK was used, this is the Android or iPhone OS () or the value you specified when using the PutEvents API.	
package_name	The package name.	
quantity	The quantity of items purchased. Used only in a monetization event.	
reported_price	A string representation of the price reported by the purchasing library. Used only in a monetization event.	
session_id	An ID for all events that occurred in the same session.	
start_timestamp	Time when the session began, expressed as an epoch timestamp with milliseconds. Used only on <code>_session.start</code> events.	
stop_timestamp	Time when the session ended, expressed as an epoch timestamp with milliseconds. Used only on <code>_session.stop</code> events.	
store	The name of the store where the transaction took place. Used only in a monetization event.	

Attribute	Description	
symbol	Currency symbol for the purchase. Used only in a monetization event.	
title	The title of the app.	
transaction_id	An identifier for the transaction. Used only in a monetization event.	
version	The platform version.	
version	The version of the Amazon Mobile Analytics SDK used by the app.	
version_code	The version code.	
version_name	The version name.	

Amazon Redshift Schema for Event Data

A new schema called AWSMA is created to contain the tables and views to access the data. The AWSMA.v_event view should be the view used to access events. You can sign in to your Amazon Redshift cluster using the `eventreader` user and use the password configured in the Auto Export feature. This user will have full select access to the AWSMA.v_event view.

JSON Amazon Redshift Schema for Event Data

Column Name	Data Type	Description
event_type	varchar(256)	The name that was specified when the event was recorded in the SDK or the name of a system event that was recorded by the SDK (for example, <code>_session.start</code> , <code>_session.stop</code>).
event_timestamp	timestamp	The UTC timestamp (of the device) when the event occurred.
arrival_timestamp	timestamp	The UTC timestamp (of the server) when the event was received by the AWS server.
event_version	char(12)	The version of the event structure (typically 3.0).
application_app_id	varchar(64)	The ID used to identify the app.
application_package_name	varchar(256)	The package name of the app.
application_version_name	varchar(256)	The version name associated with the app.

Amazon Mobile Analytics User Guide
Amazon Redshift Schema for Event Data

Column Name	Data Type	Description
application_version_code	varchar(256)	The version code associated with the app.
application_title	varchar(256)	The title of the app.
application_cognito_identity_pool_id	varchar(64)	This is populated if you are using Amazon Cognito authentication, it is the Amazon Cognito identity pool ID.
application_sdk_name	varchar(256)	The AWS SDK that sent the event.
application_sdk_version	varchar(256)	The version of the AWS SDK that sent the event.
client_id	varchar(64)	An ID used to identify other events sent from the same app instance. If you used the AWS Mobile SDK to record events, all events from the same app on a device will have the same client_id. This is a GUID string. You can use this value to sample events from a random set of users.
client_cognito_id	varchar(64)	The Cognito client ID, this column is populated if you are using Amazon Cognito authentication. This is useful to correlate different users if you are using non-anonymous users.
device_model	varchar(256)	The device model.
device_make	varchar(256)	The device make.
device_platform_name	varchar(256)	The device platform (for example, iPhoneOS or Android).
device_platform_version	varchar(256)	The device platform version.
device_locale_code	varchar(256)	The locale of the device (for example, en_US).
device_locale_language	varchar(64)	The language part of the locale (for example, "en").
device_locale_country	varchar(64)	The country part of the locale (for example, "US").
session_id	varchar(64)	A GUID string used to identify events that came from the same app and device in the same session.
session_start_timestamp	timestamp	The timestamp when the session started.

Column Name	Data Type	Description
session_end_timestamp	timestamp	The timestamp when the session stopped. Populated for _session.stop events only.
monetization_transaction_id	varchar(64)	An identifier for the transaction.
monetization_transaction_store	varchar(64)	The name of the app store used for the transaction.
monetization_transaction_item_id	varchar(64)	An identifier for the item purchased.
monetization_transaction_quantity	FLOAT8	The quantity of items purchased.
monetization_transaction_price_reported	varchar(64)	A string representation of the price reported by the purchasing library. This is different for each store.
monetization_transaction_price_amount	FLOAT8	The numerical price value.
monetization_transaction_price_currency_code	varchar(16)	The currency code for the purchase.
monetization_transaction_price_currency_symbol	varchar(32)	The currency symbol for the purchase.
a_levelName	varchar(4000)	All custom attributes start with "a_".
a_playerClass	varchar(4000)	All custom attributes start with "a_".
m_score	FLOAT8	All custom metrics start with "m_".
m_timeToCompleteMinutes	FLOAT8	All custom metrics start with "m_".

Example Amazon Redshift Queries

The “users” columns in the following queries count distinct client_cognito_id values only when users are authenticated with Amazon Cognito. For unauthenticated use cases, use the client_id for app-device counts.

Top item purchases, by store (last 30 days):

```
SELECT
  application_app_id AS"app id",
  monetization_transaction_item_id AS"item id",
  monetization_transaction_store AS"store",
  COUNT(DISTINCTclient_id) AS"devices",
  COUNT(DISTINCTclient_cognito_id) AS"users",
  SUM(monetization_transaction_quantity) AS"quantity",
  SUM(monetization_transaction_price_amount) "amount (Apple only)",
  monetization_transaction_price_currency_code AS"currency (Apple only)"
```

```
FROM
AWSMA.v_event
WHERE
event_type = '_monetization.purchase'AND
event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
GROUPBY
"app id",
"item id",
"currency (Apple only)",
"store"
ORDERBY
"app id"ASC,
"item id"ASC,
"quantity"DESC,
"store",
"devices"DESC
;
```

30 day active users and devices, by device language code:

```
SELECT
application_app_id AS"app id",
COUNT(DISTINCTclient_id) AS"devices",
COUNT(DISTINCTclient_cognito_id) AS"users",
device_locale_language AS"language"
FROM
AWSMA.v_event
WHERE
event_type = '_session.start'AND
event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
GROUPBY
"app id",
"language"
ORDERBY
"app id"ASC,
"devices"DESC,
"language"
;
```

30 day active users and devices, by app version:

```
SELECT
application_app_id AS"app id",
device_platform_name AS"platform",
application_version_name AS"version name",
application_version_code AS"version code",
COUNT(DISTINCTclient_id) AS"devices",
COUNT(DISTINCTclient_cognito_id) AS"users"
FROM
AWSMA.v_event
WHERE
event_type = '_session.start'AND
event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
GROUPBY
"app id",
"platform",
```

```
"version name",  
"version code"  
ORDERBY  
"app id"ASC,  
"platform"ASC,  
"devices"DESC,  
"version name"DESC,  
"version code"DESC  
;
```

Explanation of Infrastructure

The Mobile Analytics and Amazon Redshift combination includes the following components:

- An Amazon Redshift cluster that holds all Auto Export events. It can start small and scale up to petabytes of information. For more information, see [Amazon Redshift](#).
- A VPC to isolate network resources to control access. There is no additional charge for using Amazon Virtual Private Cloud (Amazon VPC) , aside from the standard Amazon EC2 usage charges. For more information, see [Amazon VPC](#).
- A t2.micro EC2 instance to facilitate the transfer of data from your S3 subscription bucket into Amazon Redshift. For more information, see [Amazon EC2](#).
- Cloudwatch Logs and metrics for information about the health of the data transfer to Amazon Redshift.
- Python scripts that run on your EC2 instance and allow you to interact with the Amazon Redshift cluster.

Amazon Redshift Cluster

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze all of your data using your existing business intelligence tools. For more information, see [Amazon Redshift](#).

Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. For more information, see [Amazon Virtual Private Cloud](#).

EC2 Instance

When the EC2 instance is provisioned, it downloads the Mobile Analytics Python scripts. After the EC2 instance starts, these scripts connect to the Amazon Redshift cluster and set up the users and tables required to process Mobile Analytics events. The scripts are invoked every hour to load all new event data into your Amazon Redshift cluster.

You can connect to the console using SSH with the EC2 keyname specified in the advanced settings:
ssh -i your-keyname.pem ec2-user@public-dns-of-instance

Note

Your pem file must be set to read-only permissions. Run the following command to ensure the file will be accepted: **chmod 400 your-keyname.pem**.

For more information about EC2 instances, see [Amazon EC2](#).

CloudWatch Metrics and Logs

The EC2 instance records CloudWatch metrics and logs for debugging and monitoring of the EC2 instance. For more information, see [Amazon CloudWatch](#). To view metrics about the execution of the EC2 load events operation:

1. Go to the [CloudWatch console](#).
2. On the left navigation pane, choose Custom Metrics.
3. Choose the LoadEventsDuration metric.

To view logs about the invocation of the EC2 instance:

1. Go to the [CloudWatch console](#).
2. Choose **Logs**.
3. Choose the **LogGroup** for the EC2 instance.
4. Choose the application log stream.

Troubleshooting

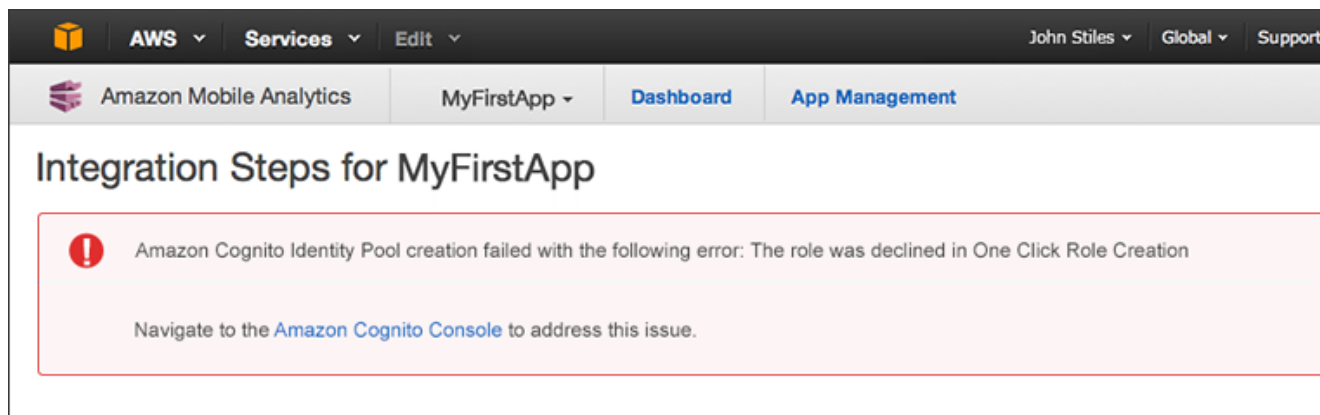
If you encounter issues or difficulties when working with Mobile Analytics, consult the topics in this section.

Topics

- [Troubleshooting Default IAM Role Creation \(p. 65\)](#)

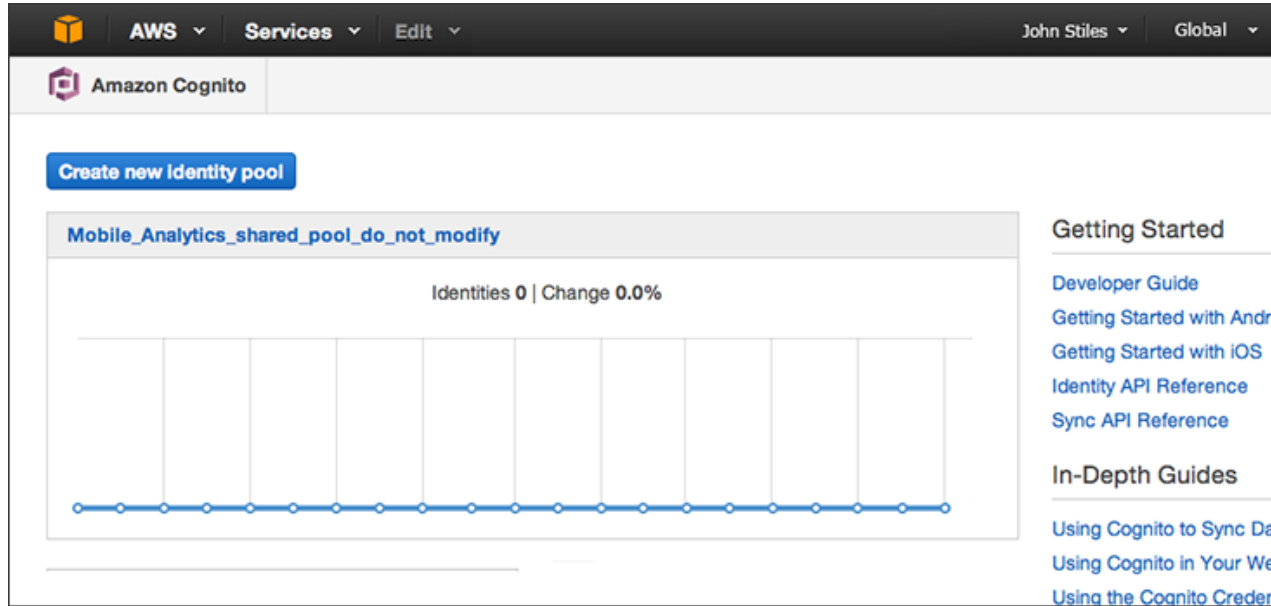
Troubleshooting Default IAM Role Creation

The following error message is displayed if Mobile Analytics is unable to create a default IAM role and Cognito identity pool when you add an app:

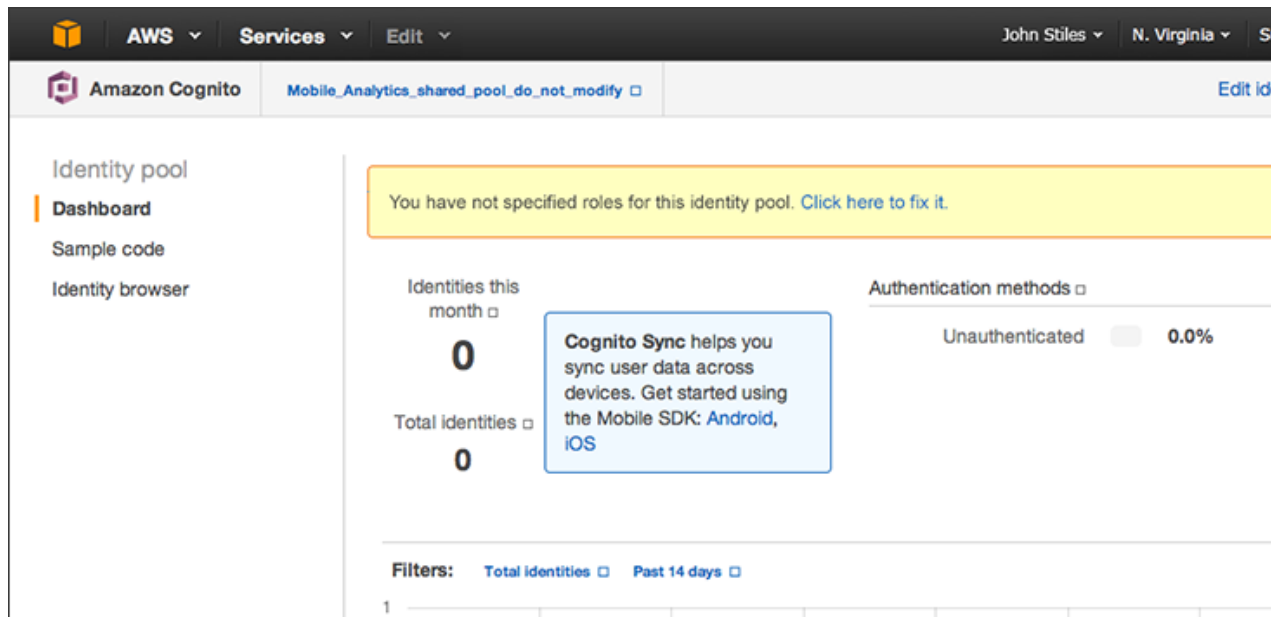


To correct a failed attempt to create a default IAM role and Cognito identity pool

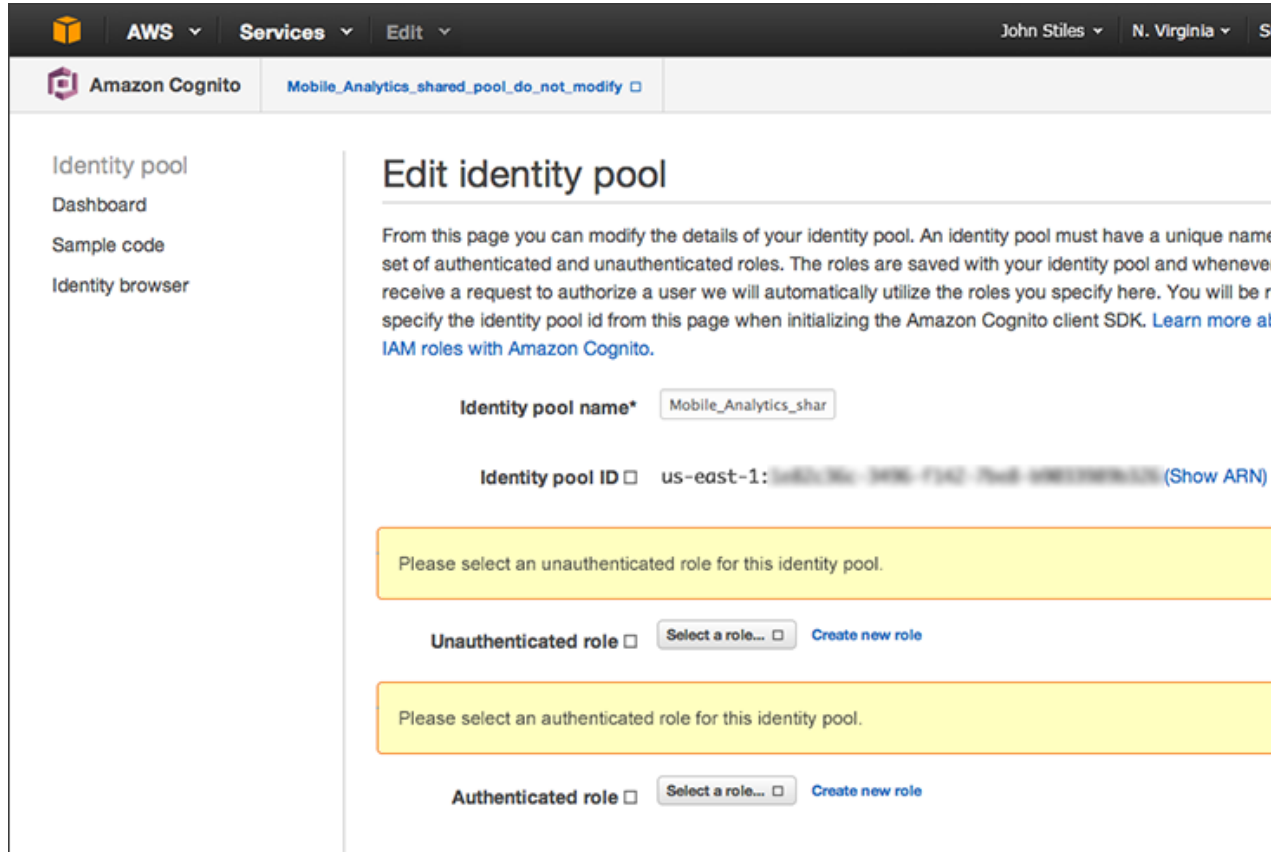
1. Open the Amazon Cognito console.
2. Click **Mobile_Analytics_shared_pool_do_not_modify**.



3. Choose **Click here to fix it.**



4. For **Unauthenticated role** and **Authenticated role**, choose an existing role or create a new role.



5. Return to the Mobile Analytics console and the integration steps for your app.

Limits

The following is a list of limits for a Mobile Analytics custom event.

Capacity	Limit
Total number of custom events per application.	1,500
Total number of attributes and metrics per event.	40
Number of characters in the key of an attribute or metric.	50
Number of characters in the value of an attribute.	200
Number of apps per AWS account.	100

REST API Reference

This is the Amazon Mobile Analytics REST API reference. It contains examples of the following events, requests, and operations.

Topics

- [Making HTTP Requests to Mobile Analytics \(p. 69\)](#)
- [PutEvents \(p. 71\)](#)
- [Examples \(p. 76\)](#)

Making HTTP Requests to Mobile Analytics

If you don't use the [AWS Mobile SDK](#), you can perform Mobile Analytics operations over HTTP using the POST request method. The POST method requires you to specify the operation in the header of the request and provide the data for the operation in JSON format in the body of the request.

HTTP Header Contents

Mobile Analytics requires the following information in the header of an HTTP request:

Host

The Mobile Analytics endpoint. This value must be
`https://mobileanalytics.us-east-1.amazonaws.com`

X-Amz-Date

The date. Must be specified in ISO 8601 standard format, in UTC time. For example:

`20130315T092054Z`

Authorization

The set of authorization parameters that AWS uses to ensure the validity and authenticity of the request. For more information, see [Signature Version 4 Signing Process](#).

User Agent

Information about the user agent originating the request.

X-Amz-Client-Context

Information about the client interacting with Mobile Analytics. Data in a client context describes the app and the environment in which it runs. For details about the contents of the client context, see [PutEvents \(p. 71\)](#).

X-Amz-Security-Token

If you sign your request using temporary security credentials, you must include the corresponding security token in your request by adding the X-Amz-Security-Token header.

For information on signing requests using temporary security credentials in your REST API requests, see [Signing and Authenticating REST Requests](#).

Content-Type

Specifies JSON and the version. For example, Content-Type: application/json

Content-Length

The payload size in bytes.

HTTP Header Example

The following is an example header for an HTTP request for Mobile Analytics.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>","app_version_code":"<app_version_code>","app_package_name":"<app_package_name>"},"custom":{"env":{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_version>","locale":"<locale>"}}}
```

HTTP Body Content

The body of an HTTP request contains the data for the operation specified in the header of the HTTP request. The data must be formatted according to the [JSON data schema](#) for Mobile Analytics. For the PutEvents operation, the body content of the HTTP request consists of an array of one or more events.

HTTP Body Example

The following is an example of the body for an HTTP request for Mobile Analytics.

```
{
  "events": [
    {
      "eventType": "<Event type>",
      "timestamp": "<ISO 8601 date>",
      "session": {
        "id": "<Session id>",
        "startTimestamp": "<ISO 8601 date>"
      },
      "attributes": {
```

```
        "<Optional string name>": "<Optional string value>",  
        ...  
        "<Optional string name>": "<Optional string value>"  
    },  
    "metrics": {  
        "<Optional string name>": <Optional numeric value>,  
        ...  
        "<Optional string name>": <Optional numeric value>  
    }  
  },  
  ...  
]  
}
```

PutEvents

The PutEvents operation records one or more events. You can have up to 1,500 unique custom events per app, any combination of up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

Topics

- [Requests \(p. 71\)](#)
- [Responses \(p. 76\)](#)

Requests

Client Context Header

Syntax

```
x-amz-Client-Context: {  
    "client": {  
        "client_id": "<client_id>",  
        "app_title": "<app_title>",  
        "app_version_name": "<app_version_name>",  
        "app_version_code": "<app_version_code>",  
        "app_package_name": "<app_package_name>"  
    },  
    "custom": {},  
    "env": {  
        "platform": "<platform>",  
        "model": "<model>",  
        "make": "<make>",  
        "platform_version": "<platform_version>",  
        "locale": "<locale>"  
    },  
    "services": {  
        "mobile_analytics": {  
            "app_id": "<mobile_ana
```

```
lytics_app_id>"  
  
    }  
}
```

Description

The operation takes the following request header.

x-amz-client-context

The request header.

client

Name-value pairs that describes the client application.

client_id

A unique identifier representing this installation instance of your app.

Type: String

Default: None

Required: Yes

app_title

The title of your app. For example, **My App**.

Type: String

Default: None

Required: Yes

app_version_name

The version of your app. For example, **v2.0**.

Type: String

Default: None

Required: No

app_version_code

The version code for your app. For example, **3**.

Type: String

Default: None

Required: No

app_package_name

The name of your package. For example, **com.example.my_app**.

Type: String

Default: None

Required: No

custom

User defined name-value pairs that describe this installation of the application.

Type: Map

Default: None

Required: No

env

Name-value pair that describes the device that runs the event.

platform

The operating system of the device. For example, **iphoneos**.

Type: String

Valid values: **iphoneos**, **android**, **windowsphone**, **blackberry**, **macos**, **windows**, **linux**

Default: None

Required: Yes

model

The model of the device. For example, **Nexus**.

Type: String

Default: None

Required: No

make

The manufacturer of the device. For example, **samsung**.

Type: String

Default: None

Required: No

platform_version

The version of the operating system of the device. For example, **4.0.4**.

Type: String

Default: None

Required: No

locale

The locale of the device. For example, **en_US**.

Type: String

Default: None

Required: No

services

Name-value pair that contains service specific sections.

mobile_analytics

Name-value pair that describes service specific attributes.

app_id

The value obtained from the Mobile Analytics console to record data to.

Type: String

Default: None

Required: Yes

Request Body

Syntax

```
{
  "events": [
    {
      "eventType": "<Event type>",
      "timestamp": "<ISO 8601 date>",
      "session": {
        "id": "<Session id>",
        "startTimestamp": "<ISO 8601 date>"
      },
      "attributes": {
        "<Optional string name>": "<Optional string value>",
        "...",
        "<Optional string name>": "<Optional string value>"
      },
      "metrics": {
        "<Optional string name>": <Optional numeric value>,
        "...",
        "<Optional string name>": <Optional numeric value>
      }
    },
    ...
  ]
}
```

Description

This operation takes the following request content.

Events

An array of JSON objects representing a batch of unique event occurrences in your app. Each JSON object in the array consists of the following:

eventType

A name signifying an event that occurred in your app. This is used for grouping and aggregating like events together for reporting purposes.

Type: String

Default: None

Required: Yes

timestamp

The time the event occurred in ISO 8601 standard date time format. For example, `2014-06-30T19:07:47.885Z`

Type: String

Constraints: Must follow ISO 8601 format

Default: None

Required: Yes

attributes

A collection of key-value pairs that give additional context to the event. The key-value pairs are specified by the developer.

This collection can be empty or the attribute object can be omitted.

Type: JSON object of key-value pairs (String:String)

Constraints: Key can be up to 50 characters or less and the value can be up to 200 characters.

Default: None

Required: No

metrics

A collection of key-value pairs that gives additional measurable context to the event. This key has the following key-values pairs. The pairs specified by the developer.

This collection can be empty or the attribute object can be omitted.

Type: JSON object of key value pairs (String:Number)

Constraints: Key can be up to 50 characters.

Default: None

Required: No

session

Describes the session. Session information is required on ALL events. This key has the following key-value pairs.

id

A unique identifier for the session.

Type: String

Default: None

Required: Yes

startTimestamp

The time the event occurred in ISO 8601 standard date time format. For example, `2014-06-30T19:07:47.885Z`

Type: String

Constraints: Must follow ISO 8601 format

Default: None

Required: Yes

version

Describes the version. This value must always be `v2.0`.

Type: String

Constraints: Must always be `v2.0`.

Default: None

Required: Yes

JSON Example

```
{ "events": [
  {
    "metrics": {
      "Score": 12345,
      "TimeInLevel": 64
    },
    "session": {
      "id" : "<session id>",
      "startTimestamp" : "2014-06-30T19:07:47.885Z"
    },
    "attributes": {
      "LevelName": "Level1",
      "CharacterClass": "Warrior",
      "Successful": "True"
    },
    "eventType": "LevelComplete",
    "version": "v2.0",
    "timestamp": "2014-06-30T19:07:47.885Z"
  }
]}
```

Responses

Syntax

```
HTTP/1.1 202
x-amzn-RequestId: <A request id>
Content-Type: application/json
```

Response Headers

This operation has the following response codes.

202 Accepted

The request has been accepted for processing, however the events have not been fully processed.

400 Bad Request

The x-amz-client-context header is missing or invalid.

OR

The event payload is missing or invalid.

403 Forbidden

The request is not authorized to perform this action.

413 Request Entity Too Large

The payload is too large. The payload cannot exceed 1024 KB.

Examples

For examples of typical tasks performed with the PutEvents operation, see [Using the REST API \(p. 30\)](#).

Document History

The following table describes the important changes to the documentation since the last release of Mobile Analytics.

- **API version:** 2014-06-05
- **Latest documentation update:** September 4, 2015

Change	Description	Date Changed
Documentation Improvements	The User Guide has been expanded and reorganized in parts. Examples of using the REST API have been moved from the reference to a series of topics that describe them in more detail.	September 4, 2015
Xamarin Support Added	The AWS Mobile SDK adds support for Xamarin, including Mobile Analytics. The Mobile Analytics User Guide now includes links to the Xamarin SDK Developer Guide regarding Mobile Analytics. Displayed integration steps now include sample code and steps for integration with a Xamarin app.	July 29, 2015
Streamlined New App Workflow	The process for creating a new app now includes a one-click wizard for generating the necessary IAM role and Cognito identity pool for users with no existing identity pools to use. Users with existing identity pools have the ability to create a new identity pool or use one of their existing identity pools with new apps. For more information, see Adding an App to Mobile Analytics. (p. 5)	July 14, 2015

Change	Description	Date Changed
New Supported Platforms	The Mobile Analytics console now displays reports on an expanded set of platforms. You can now filter the events that are compiled into the console reports to display the following platform choices: All Platforms, iOS, Android, Fire OS, Windows Phone, Blackberry, Windows, Mac, and Linux. Screen shots have been updated.	June 30, 2015
Auto Export	Added a topic on how to auto export app data to an Amazon S3 bucket. For more information, see Amazon Mobile Analytics Auto Export .	December 17, 2014