Use Mobile Center to quickly get the insights that your app users won't tell you

# Overview

Mobile Center brings together multiple services, commonly used by mobile developers, into a single, integrated product. You can build, test, distribute, and monitor your mobile apps and easily add backend cloud services to scale your app to millions of users on demand.

# Objectives

* Create an Android Native project.
* Create a Mobile Center app.
* Integrate the Android Native project with the Mobile Center app.
* Track app analytics and events to Mobile Center.

# Prerequisites

* Windows 10 for Android application
* Android Studio for Android application
* A Mobile Center account (<https://mobile.azure.com>)

# Intended Audience

This Quick Start Challenge is intended for developers who are familiar Android development in Java.

# Task 1: Create an Android Native app

You will need a JDK, and Android Studio.

Android Studio, which we will install next, requires a recent version of the [Java SE Development Kit (JDK)](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html).

1. [Download and install Android Studio](https://developer.android.com/studio/index.html)

Choose Custom installation when prompted by the Setup Wizard, and proceed to the next step.

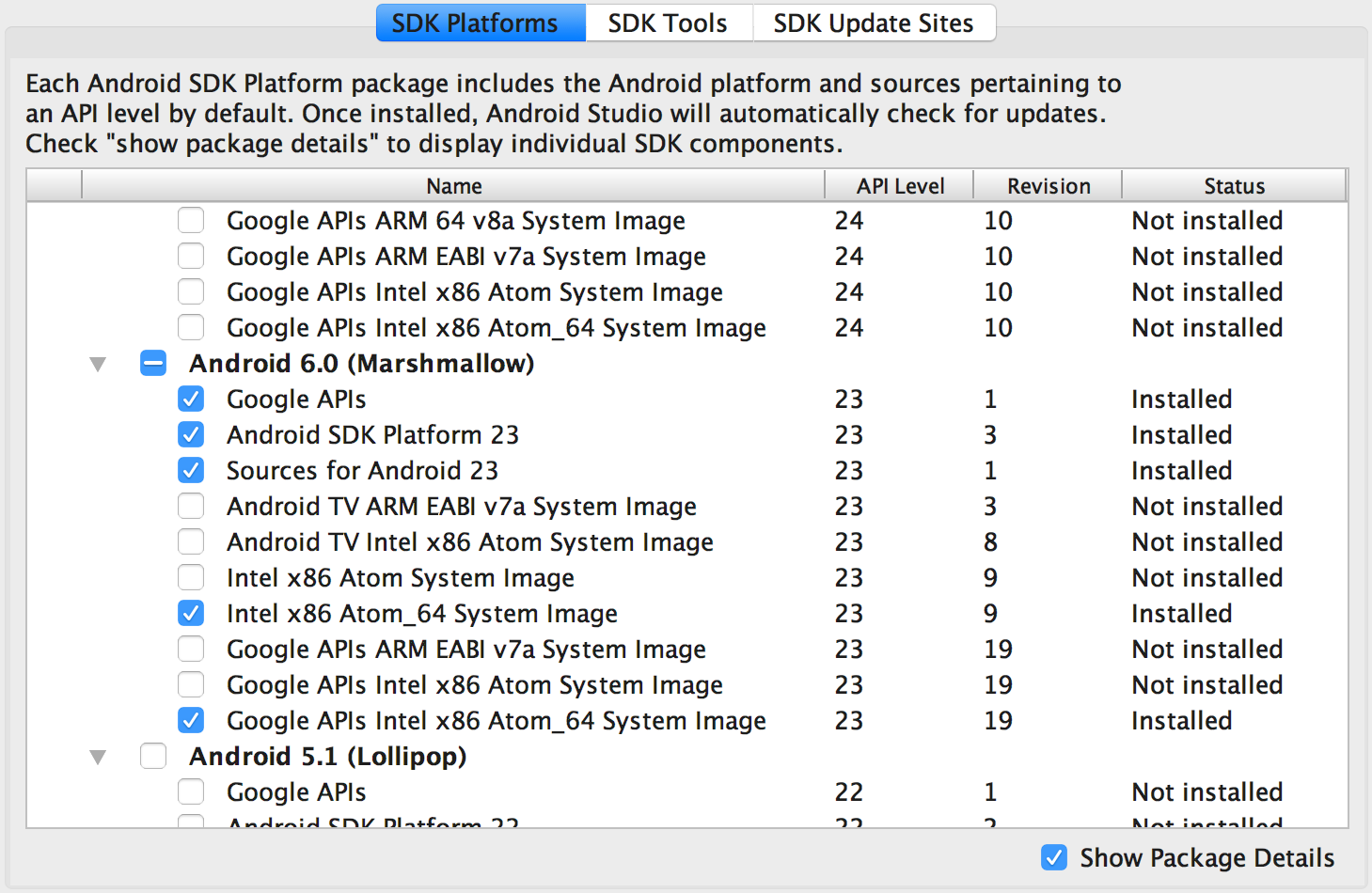
Make sure the boxes next to all of the following are checked:

* *Android SDK*
* *Android SDK Platform*
* *Performance (Intel ® HAXM)*
* *Android Virtual Device*

1. Android Studio installs the most recent Android SDK by default. App was developed using Android 6.0 (Marshmallow) SDK. To install it, launch the SDK Manager, click on "Configure" > "SDK Manager" in the "Welcome to Android Studio" screen.

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the Android 6.0 (Marshmallow) entry, then make sure the following items are all checked:

* *Google APIs*
* *Android SDK Platform 23*
* *Intel x86 Atom\_64 System Image*
* *Google APIs Intel x86 Atom\_64 System Image*
* *Android SDK Manager*



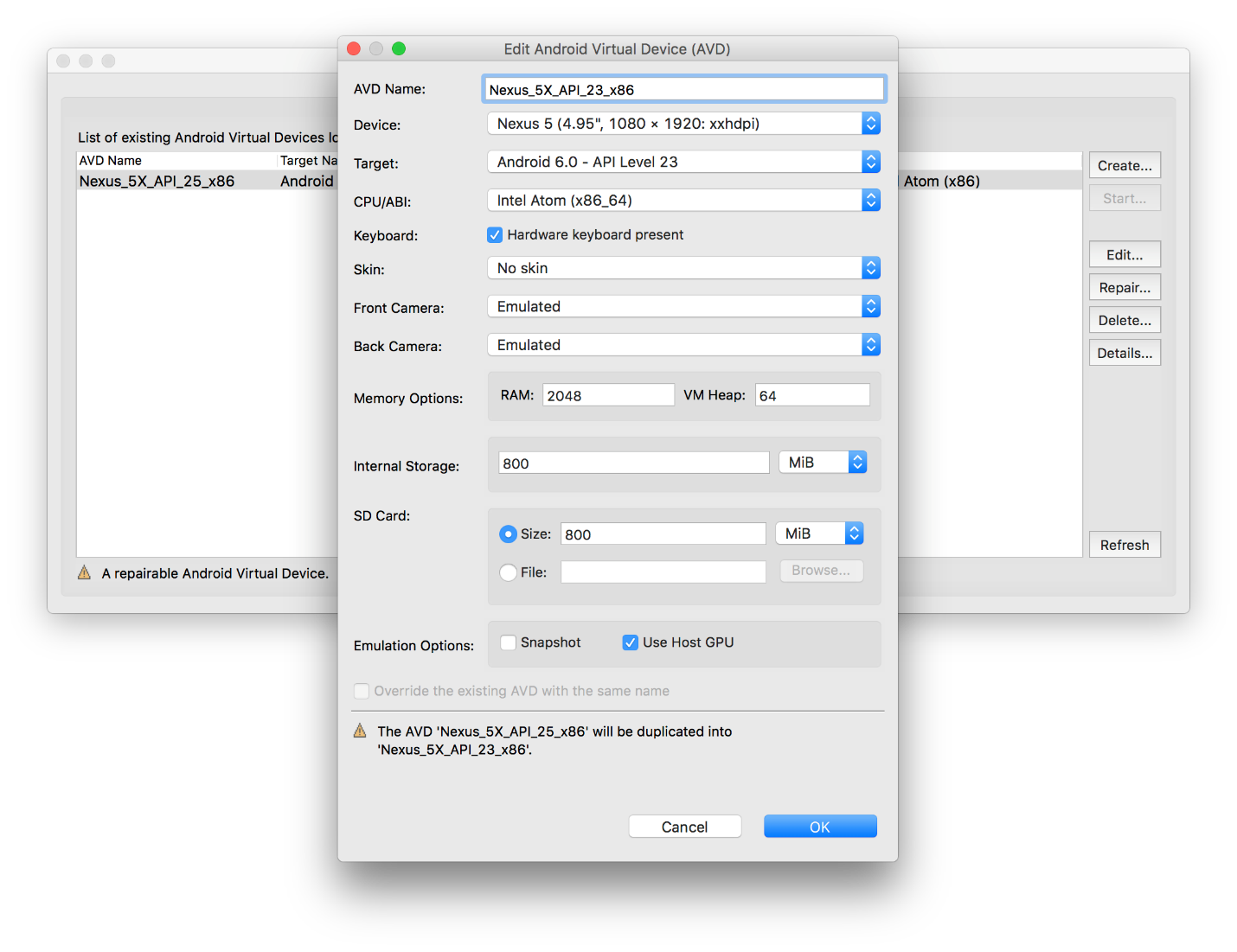
Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build Tools" entry, then make sure that Android SDK Build-Tools 23.0.1 is selected.

Finally, click "Apply" to download and install the Android SDK and related build tools.

1. Starting the Android Virtual Device

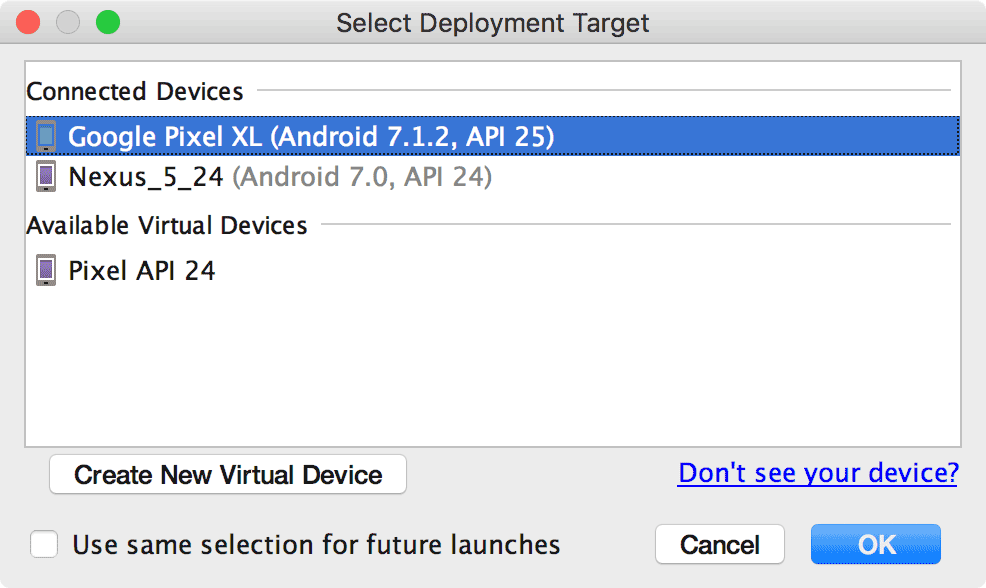
You can see the list of available AVDs by opening the "AVD Manager" from within Android Studio.

Once in the "AVD Manager", select your AVD and click "Edit...". Choose "Android 6.0 - API Level 23" under Device, and "Intel Atom (x86\_64)" under CPU/ABI. Click OK, then select your new AVD and click "Start...", and finally, "Launch".



1. Build and run the app

To build and run your app, select Run > Run in the menu bar (or click Run https://developer.android.com/studio/images/buttons/toolbar-run.png in the toolbar). If it's the first time running the app, Android Studio asks you to select a deployment target as shown in the figure below. Simply select a device to install and run your app.

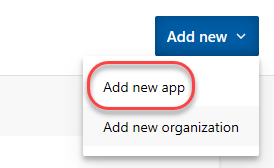


If the dialog says, "No USB devices or running emulators detected," then you need to [set up and connect your device](https://developer.android.com/studio/run/device.html) or launch an emulator by clicking a device listed under Available Virtual Devices.

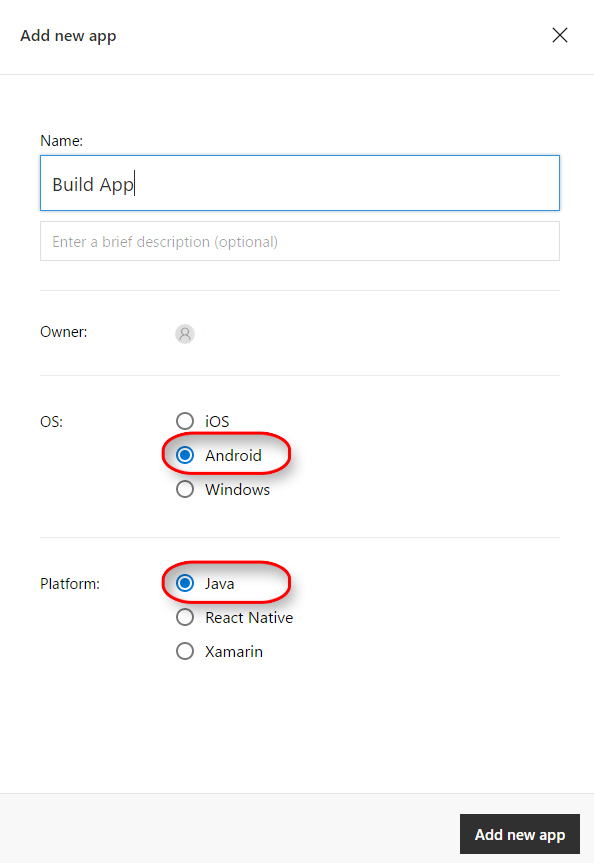
You can also deploy your app in debug mode by clicking Debug https://developer.android.com/studio/images/buttons/toolbar-debug.png. Running your app in debug mode allows you to set breakpoints in your code, examine variables and evaluate expressions at run time, and run debugging tools.

# Task 2: Create the Mobile Center app

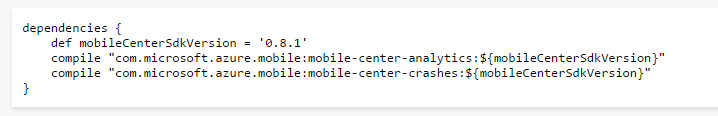
1. Log in to your Mobile Center account at <https://mobile.azure.com>.
2. From the top right corner, select **Add new | Add new app**.



1. Enter a **Name** of **“Build App”** and set it to be for **Android** using **Java**. Click **Add new app**.



1. Once the new app has been created, there will be a set of instructions for integrating the app with Mobile Center.



# Task 3: Integrate the Android Native app with Mobile Center

1. Add the SDK to the project:

dependencies {

def mobileCenterSdkVersion = '0.8.1'

compile "com.microsoft.azure.mobile:mobile-center-analytics:${mobileCenterSdkVersion}"

compile "com.microsoft.azure.mobile:mobile-center-crashes:${mobileCenterSdkVersion}"

}

1. Start the SDK:

Open your app’s main activity class and add the following import statements.

import com.microsoft.azure.mobile.MobileCenter;

import com.microsoft.azure.mobile.analytics.Analytics;

import com.microsoft.azure.mobile.crashes.Crashes;

Look for *onCreate* callback in the same file and add the following.

MobileCenter.start(getApplication(), "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

Analytics.class, Crashes.class);

# Task 4: Adding event tracking to the app

1. It may take a while for the analytics to begin to appear in Mobile Center, so we’ll move on to the next task and add in some event tracking.
2. You can track your own custom events with up to five properties to know what's happening in your app, understand user actions, and see the aggregates in the Mobile Center portal.

Once you have started the SDK, use the *trackEvent()* method to track your events with properties. You can send up to 200 distinct event names. Also, note that there is a maximum of 256 characters supported per event name and 64 characters per event property name and event property value.

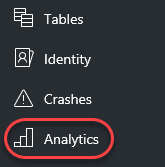
Map<String, String> properties = new HashMap<>();

properties.put("Category", "Music");

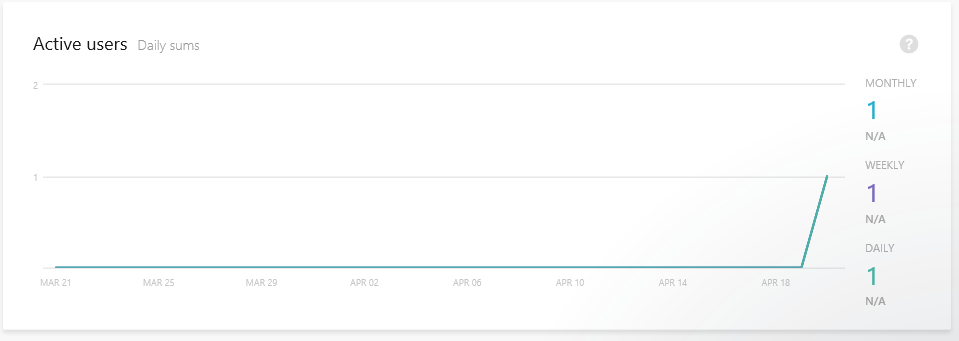
properties.put("FileName", "favorite.avi");

Analytics.trackEvent("Video clicked", properties);

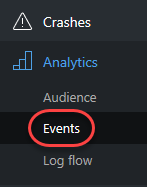
1. Go back to Simulator and press R two times to reload application. Keep in mind that while we’re using Mobile Center for an Android application, Mobile Center supports every major platform out there. As a result, you should check out the guidance for integrating other platform combinations with the Mobile Center events guidance at <https://docs.microsoft.com/en-us/mobile-center/analytics/understand-events>.
2. Return to the Mobile Center browser window. Select the **Analytics** tab.



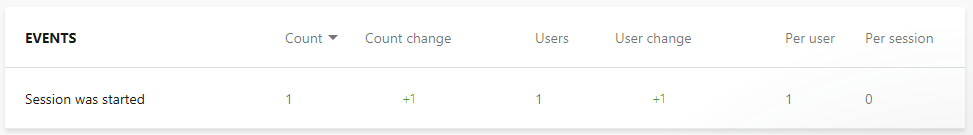
1. By now, the first user session should have been processed for display. If not, try refreshing the window every few seconds until it appears. Note that you can scroll down the page to get insight about the sessions, devices, geographies, and languages of your users, as well as reporting of the breakdown of devices per version.



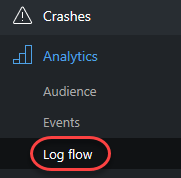
1. Select **Analytics | Events**.



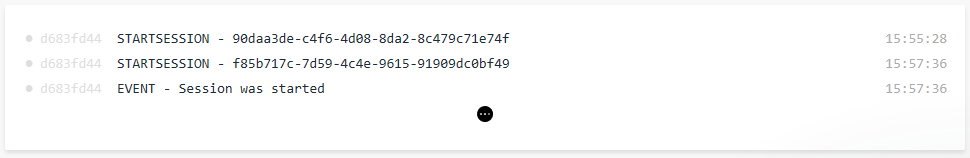
1. If the events haven’t started showing up yet, try refreshing the browser. If you run the application multiple times, then the count should increment each time you do.



1. Select **Analytics | Log flow**.



1. Log Flow will show you in real time what is being received by the backend. While the transmission may be delayed by a few seconds, this is a great place to get immediate notifications about sessions tarts, crashes, and events.



# Summary

Congratulations on completing this Quick Start Challenge! In this lab, you’ve learned how to use Mobile Center to instrument and monitor your mobile apps in development and production.

# Additional Resources

If you are interested in learning more about this topic, you can refer to the following resources:

**Documentation**: <https://docs.microsoft.com/en-us/mobile-center/>

**Team blog**: <https://blogs.msdn.microsoft.com/visualstudio/tag/visual-studio-mobile-center/>

**Android Studio User Guide**: <https://developer.android.com/studio/intro/index.html>