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 a React Native Cross-platform project.
* Create a Mobile Center app.
* Integrate the React Native project with the Mobile Center app.
* Track app analytics and events to Mobile Center.

# Prerequisites

* Windows 10 for Android application or Mac OS X for iOS application
* Android Studio for Android application or XCode for iOS application
* A Mobile Center account (<https://mobile.azure.com>)

# Intended Audience

This Quick Start Challenge is intended for developers who are familiar with JS and React JS development.

# Task 1: Create a React Native app

You will need Node, the React Native command line interface, Python2, a JDK, and Android Studio.

You can install Node and Python2 via Chocolatey, a popular package manager for Windows.

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) which can be installed using [Chocolatey](https://chocolatey.org/).

1. Open a Command Prompt as Administrator, then run:

*choco install nodejs.install*

*choco install python2*

*choco install jdk8*

If you have already installed Node on your system, make sure it is version 4 or newer. If you already have a JDK on your system, make sure it is version 8 or newer.

You can find additional installation options on [Node.js's Downloads page](https://nodejs.org/en/download/).

Node comes with npm, which lets you install the React Native command line interface.

1. Run the following command in a Terminal:

*npm install -g react-native-cli*

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.

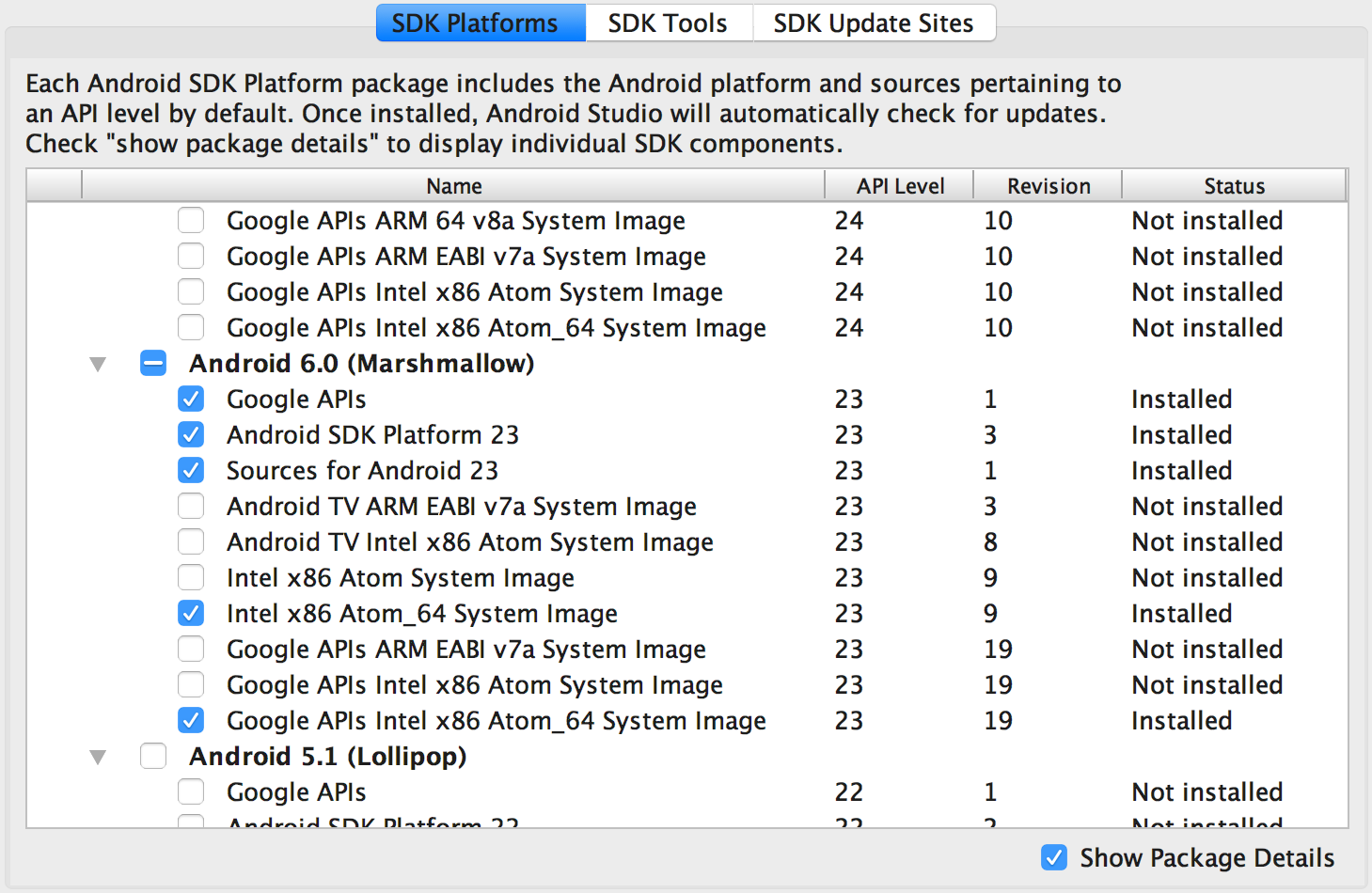
Make sure the boxes next to all of the following are checked:

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

1. Android Studio installs the most recent Android SDK by default. React Native, however, requires the 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*

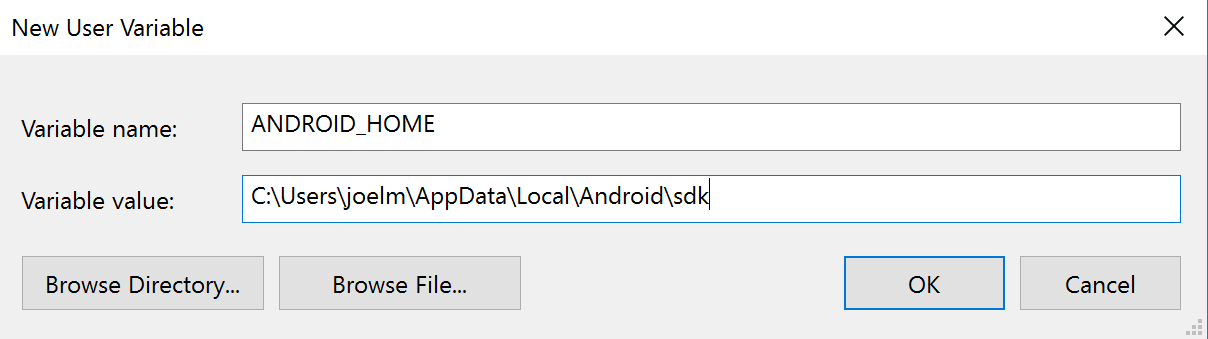


Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build Tools" entry, then make sure that Android SDK Build-Tools 23.0.1 is selected.

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

1. The React Native command line interface requires the ANDROID\_HOME environment variable to be set up.

Go to Control Panel → System and Security → System → Change settings → Advanced System Settings → Environment variables → New, then enter the path to your Android SDK.

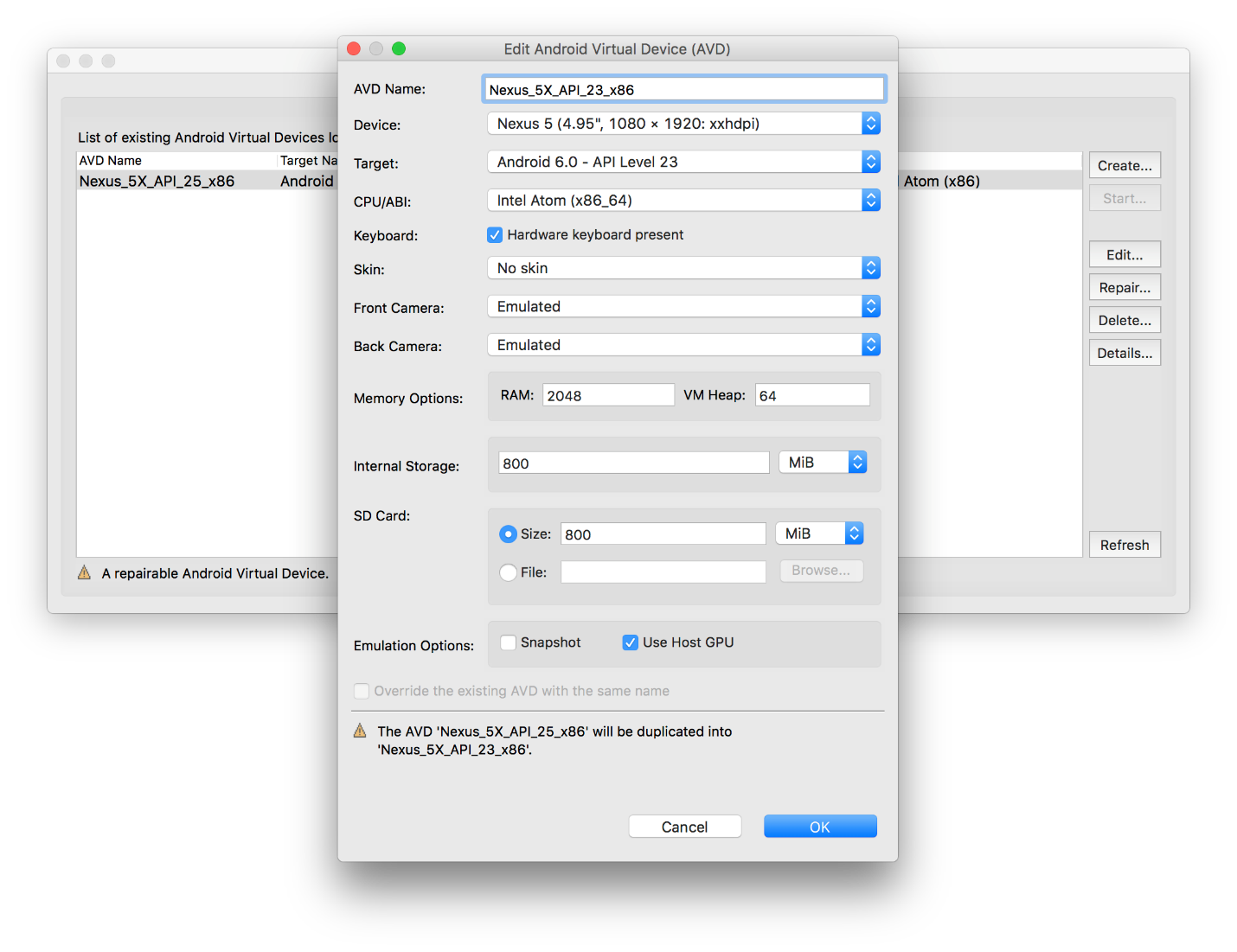


Restart the Command Prompt to apply the new environment variable.

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. Testing your React Native Installation

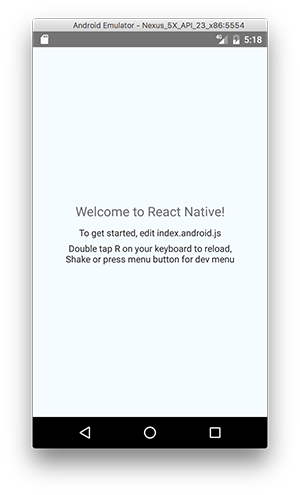
Use the React Native command line interface to generate a new React Native project called "AwesomeProject", then *run react-native run-android* inside the newly created folder:

*react-native init AwesomeProject*

*cd AwesomeProject*

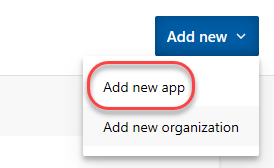
*react-native run-android*

If everything is set up correctly, you should see your new app running in your Android emulator shortly.

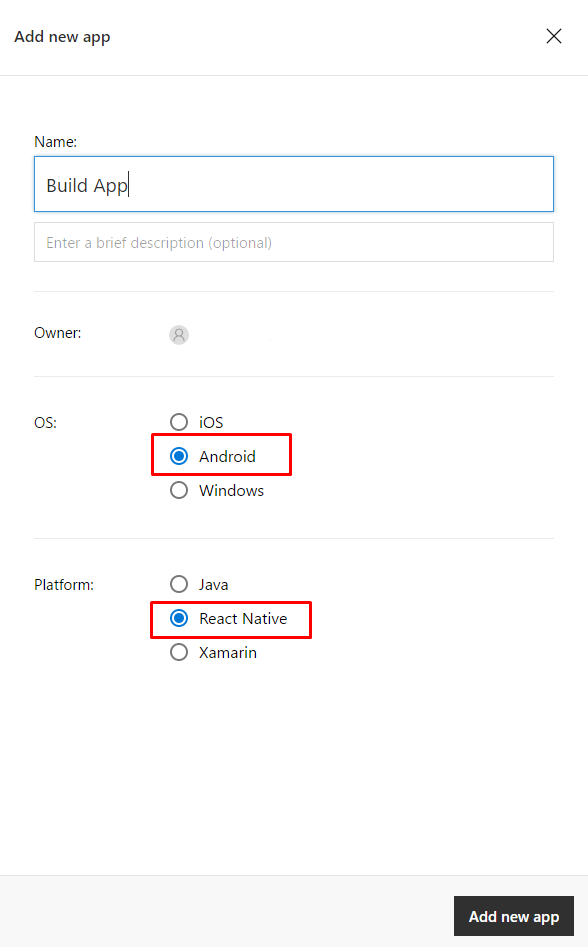


# Task 2: Create the Mobile Center app

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 **React Native**. 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. Note that the first step involves adding npm packages.



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

1. Return to projects’ folder and open **cmd**, then run:

*npm install mobile-center-analytics –save*

*npm install mobile-center-crashes –save*

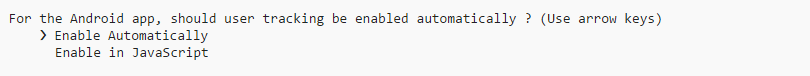
1. Link the plugin to the React Native app using the react-native link command:

*react-native link mobile-center-analytics*

*react-native link mobile-center-crashes*

The SDK will ask for the app secret per platform, which is assigned in Mobile Center. The secret may already be pre-populated, showing up beside the question. If the pre-filled secret is correct, press enter to continue. The SDK will further ask whether or not to send crash reports and user events automatically, which we recommend you do for easy setup. Opt out for greater control over user privacy.





# 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. Immediately after the **MobileCenter** line of code added previously, add the line of code below. It’s a relatively straightforward request to track a text event.

Import Analytics class to index js file:

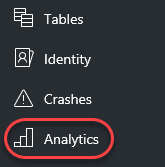
import Analytics from "mobile-center-analytics";

And call it in render function of main component:

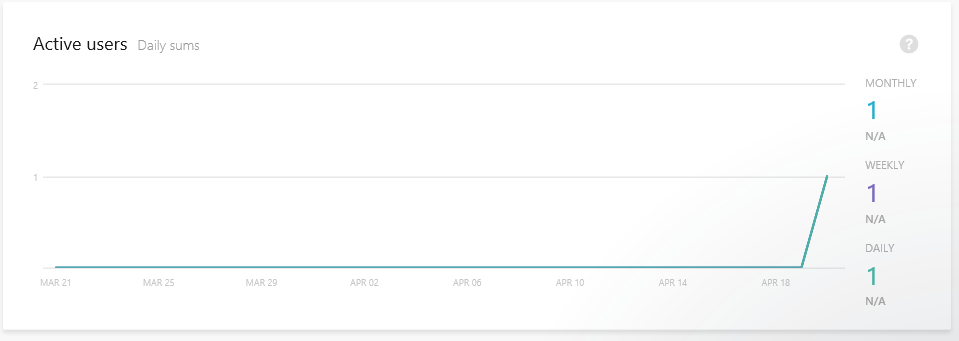
render() {

Analytics.TrackEvent("Session was started");

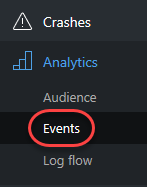
1. Go back to Simulator and press R two times to reload application. Keep in mind that while we’re using Mobile Center for an Android/React Native application, Mobile Center supports every major platform out there. As a result, you should check out the guidance for integrating other platform combinations 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.



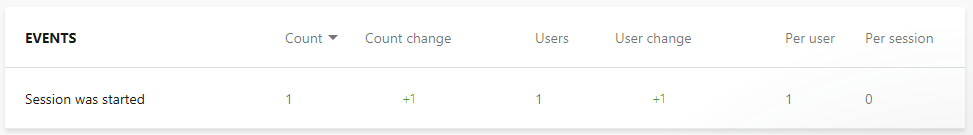
1. By now, the first user session should have been processed for display. If not, try refreshing the window every few seconds until it appears. Note that you can scroll down the page to get insight about the sessions, devices, geographies, and languages of your users, as well as reporting of the breakdown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



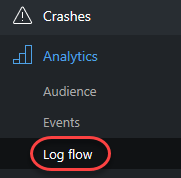
1. Select **Analytics | Events**.



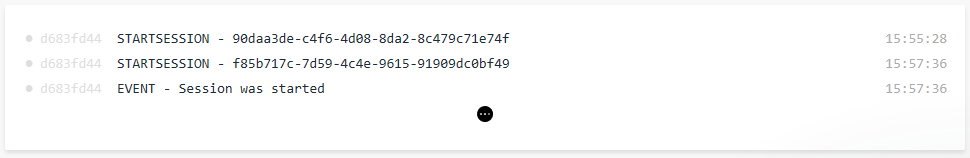
1. If the events haven’t started showing up yet, try refreshing the browser. If you run the application multiple times, then the count should increment each time you do.



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/>

**GitHub SDK**: <https://github.com/Microsoft/mobile-center-sdk-react-native>

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