# MobileFirst Quality Assurance for Cordova applications

## Overview

This tutorial is a continuation of the IBM Mobile Quality Assurance setup step. If you haven't done it yet, please do so now (../overview/).

This tutorial explains how to install and configure the Mobile Quality Assurance client using the Apache Cordova hybrid SDK, and how to enable apps to send back bug reports and feedback.

Use the command line to set up and instrument Mobile Quality Assurance. You must download and install the latest version of Apache Cordova and its required co-requisites before you begin this procedure.

### Jump to:


- Creating a simple hybrid app using Apache Cordova
- Installing the libraries
- Configuring how MQA communicates with your app
- Testing your app
- Further discovery


## Creating a simple hybrid app using Apache Cordova

You must have a mobile app to instrument MQA. If you do not have a mobile app, complete the following steps to create one:

1. Open a command prompt and navigate to the directory where you want your new app folder to be created:

   ```
   cd MyApps
   ```

2. Enter the following command to create a folder called *hello*, assign an app identifier of *com.example.hello*, and name the app *HelloWorld*:

   ```
   cordova create hello com.example.hello HelloWorld
   ```

3. Navigate to the folder that you just created for your app.

   ```
   cd hello
   ```

4. Add required files for each of the platforms on which your app runs.

   ```
   cordova platform add ios
   cordova platform add android
   ```

## Installing the libraries

The libraries support both preproduction mode or production mode. Use preproduction mode when you want internal testers to report bugs and feedback with detailed information and have apps under test automatically report crashes.

Complete these steps to install the libraries:

1. Download and extract the Cordova SDK library from Device SDKs for Mobile Quality Assurance for Bluemix (http://www.ibm.com/support/docview.wss?uid=swg27044490).
2. Navigate to the directory that contains your app project files.

   ```
   cd hello
   ```

- Add the Mobile Quality Assurance plug-in to your Cordova project by completing the following steps:
  1. Enter the following command to install the MQA plugin:

     ```
     cordova plugin add plugin_location
     ```

     Where *plugin_location* is the path to the directory of the extracted Mobile Quality Assurance plugin that you downloaded.
  2. Enter the following command to install the required device plugin:

     ```
     cordova plugin add cordova-plugin-device
     ```

  3. If your app has an Android runtime that is earlier than 4.0, rename the `project_name\app_name\platforms\android\custom_rules.xml` file to `custom_rules_bak.xml`.

## Configuring how MQA communicates with your app

For your app to send data to Mobile Quality Assurance, you must have an application key from the MQA service in Bluemix. For more information about application settings, see Registering apps. If you already have an account, you can find your application key by clicking the settings icon (gear) and then selecting App Settings.

### Configure MQA to start a new Mobile Quality Assurance session each time your app starts in the JavaScript file

1. Open the index.js file that is in the `your_project_name\www\js` directory.
2. Place the following function and required parameters inside the `onDeviceReady()` function before the end brace of the function:

```
MQA.startNewSession({
  mode: "QA",
  //or mode: "MARKET" for production mode
  android: {
     appKey: "your_MQA_Android_appKey" ,
     notificationsEnabled: true
 },
  ios: {
     appKey: "your_MQA_iOS_appKey" ,
     screenShotsFromGallery: true,
       }
},
   {
  success: function () {console.log("Session Started successfully");},
  error: function (string) { console.log("Session error" + string);}
});
```

> To learn more about starting a Mobile Quality Assurance session, including details about the different session configuration options, see Starting Cordova quality assurance sessions (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/t_start_cordova_session.html).

3. Build and run your app.

```
cordova build
cordova run
```

## Testing your app

After the environment is set up, you can load your app on your device and start testing it. Complete the following steps to report a bug or provide feedback during testing:

1. Shake your mobile device lightly.
2. Select **Report a bug** or **Give feedback**.
3. Enter the required information and select **Send** to send the information.
4. View the bug reports and feedback in your Bluemix MQA instance where you retrieved the AppKey.

For more information about reporting bugs, see Reporting bugs (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/t_ReportingBugs.html). For more information about submitting feedback, see User feedback (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/c_UserFeedback.html).

## Further discovery

After you are comfortable with the basics of setting up MQA with your app, you can read content in IBM Knowledge Center to learn and explore some of the other things that you can do. For example:

- Managing users and testers (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/t_ManagingTesters.html). You can manage the roles and access of your team as they apply to your app and MQA instance.
- Distributing and managing builds (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/t_DistributingBuilds.html). You can distribute test builds and production builds to the appropriate audience.
- Managing reports (http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/t_ManagingReports.html). You can automatically distribute test builds and production builds to your audience.
- User sentiment analytics

(http://www.ibm.com/support/knowledgecenter/SSJML5_6.0.0/com.ibm.mqa.uau.saas.doc/topics/UserSentiment.html). You can use MQA to analyze and monitor user comments and information about your app. You can also compare the analysis of those comments with the analysis of the user feedback from similar apps.