Push code updates to your Apache Cordova apps, instantly!

# Overview

# CodePush is a cloud service that enables Cordova and React Native developers to deploy mobile app updates directly to their users’ devices without having re-build a binary and/or re-distribute it through any public app stores. During this exercise, we’ll build a CodePush enabled Cordova app and leverage the CodePush CLI to create and deploy instant updates to the Cordova app.

**Note**: For simplicity purposes, this code challenge focuses on developing Cordova apps on the Android emulator. However, CodePush can also be used on iOS apps and even React Native apps.

# Objectives

* Leverage the CodePush CLI to create a CodePush account
* Register an app with CodePush
* Enable CodePush on a Cordova app

# Prerequisites

* Visual Studio 2015
* NodeJS
* CodePush CLI

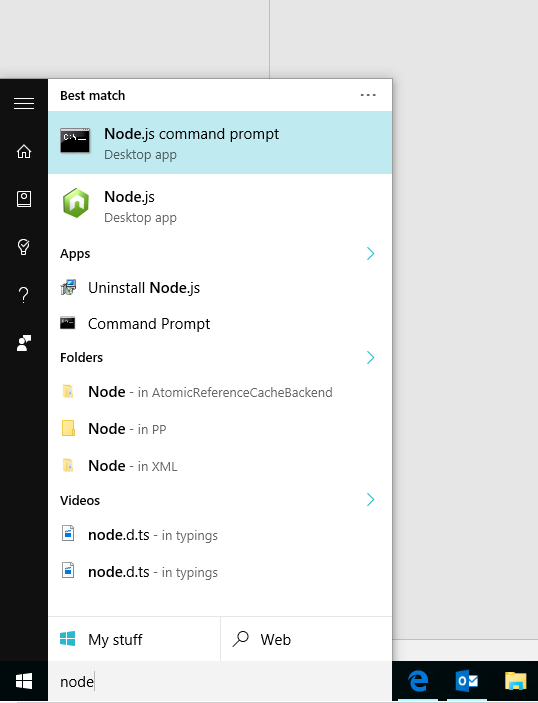
# Intended Audience

This Quick Start Challenge is intended for developer’s familiar with CLI tools, JavaScript and Visual Studio. Prior experience with Cordova, NodeJS or CodePush is needed but the experience will be easier to follow when familiar with them.

# Step 1: Create a CodePush account

Let’s start by creating an account using the NodeJS based CodePush CLI.

First, launch the NodeJS CLI from the Start menu:

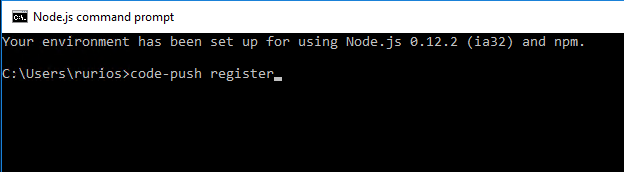


Ensure you are running the latest version of the CodePush CLI by running the following command:

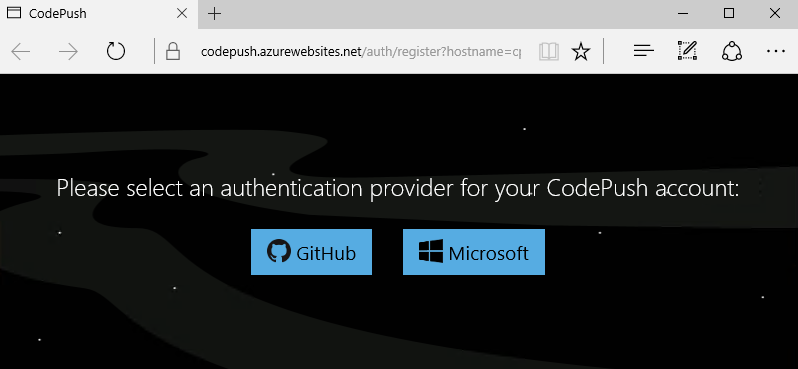
$ npm install -g code-push-cli

Now create a CodePush account by typing:

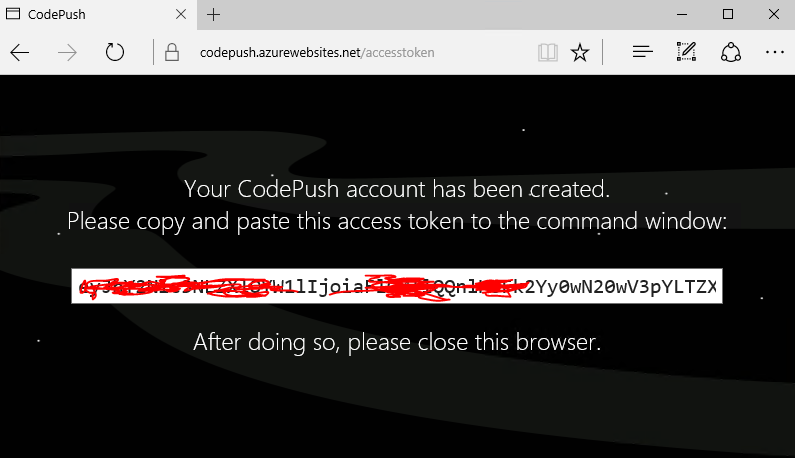
$ code-push register



This will open a new page allowing you create a CodePush account using either a GitHub or a Microsoft account.

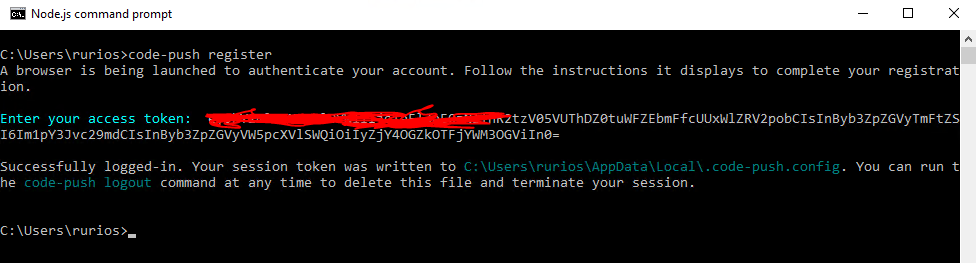


Authenticating with your provider will create the account and provide you with an access token.



Aksdjfkldfuasdjnfkasdfq93-04[ppiadsfadsfjkajsdfklldfuasdjn04ripokjfd90i3-p2z1c()o

Copy the token into the CLI and you’ll successfully log into your CodePush account.



Aksdjfkldfuasdjnfkasdfq93-04[ppiadsfadsfjkajsdfklldfuasdjn04ripokjfd90i3-po

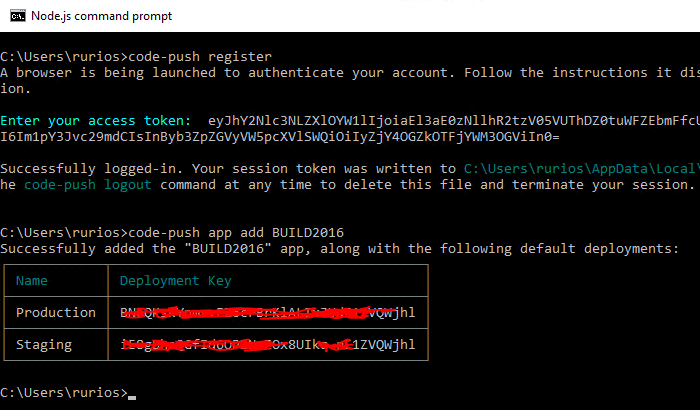
po

Aksdjfkldfuasdjnfkasdfq93-04[ppiofdklasf93-04ripokjfd90i3-po

# Step 2: Register a new app with the CodePush service

Now that the account has been setup, let’s register the app we want to use with CodePush by typing the following command:

$ code-push app add BUILD2016

Notice you are adding an app and calling it “BUILD2016

Aksdjfkld#$@123=-123dfd<>12fuasdjnfkasdfq93-04[ppiadsfadsf

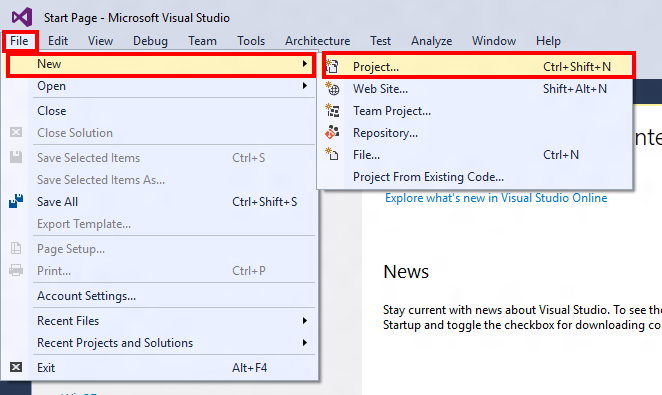
%^$12fu 90a12xcsdjfkld#$@123=-123<>asdfq93-04[ppiadsfadsf

Once the app is created, you’ll get two deployment keys: Production & Staging.

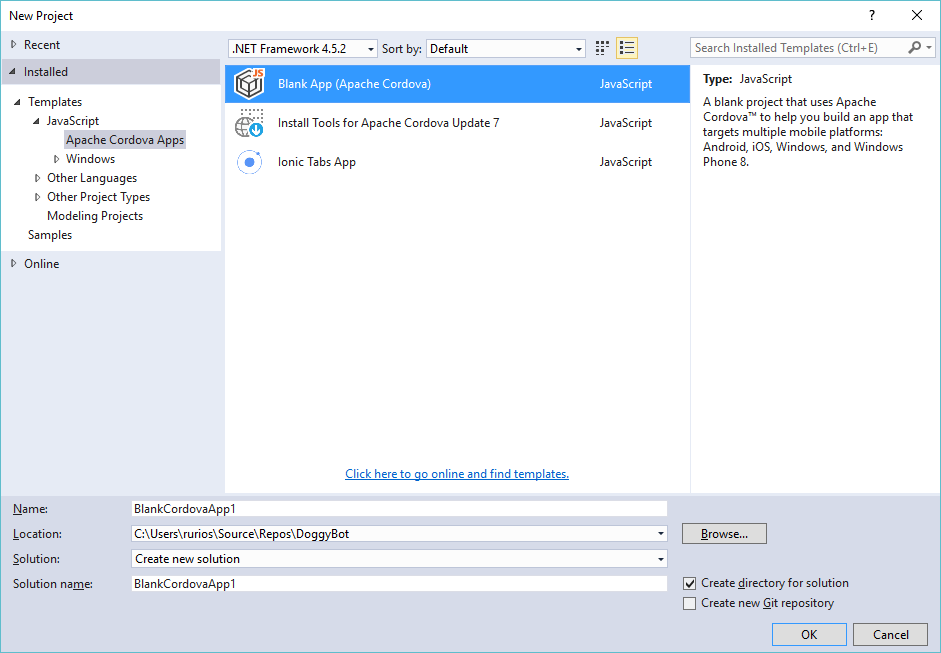
Keep the Production key handy (you can copy by selecting the text and pressing enter), as it will be used to configure the CodePush service on a Cordova app you’ll create in the next step.

# Step 3: Create a new Cordova project in Visual Studio

First, launch **Visual Studio 2015** from the Start Menu. Then, create a new project:



Select **Installed → Templates → JavaScript → Apache Cordova Apps → Blank App** for your project template:

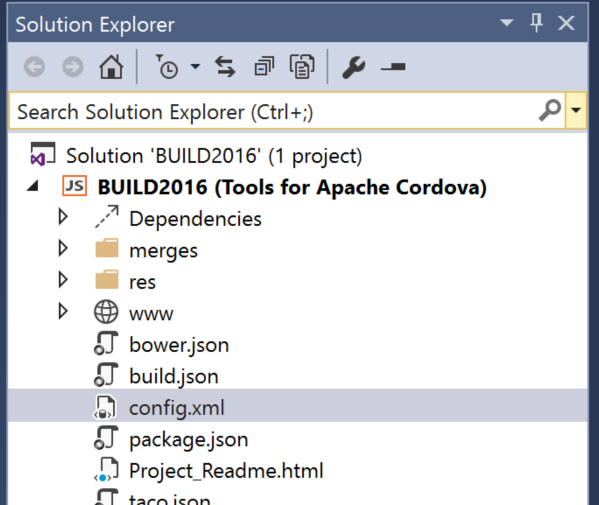


Give the project a name (e.g. BUILD2016) and create it by pressing Ok.

# Step 4: Install the CodePush plugin

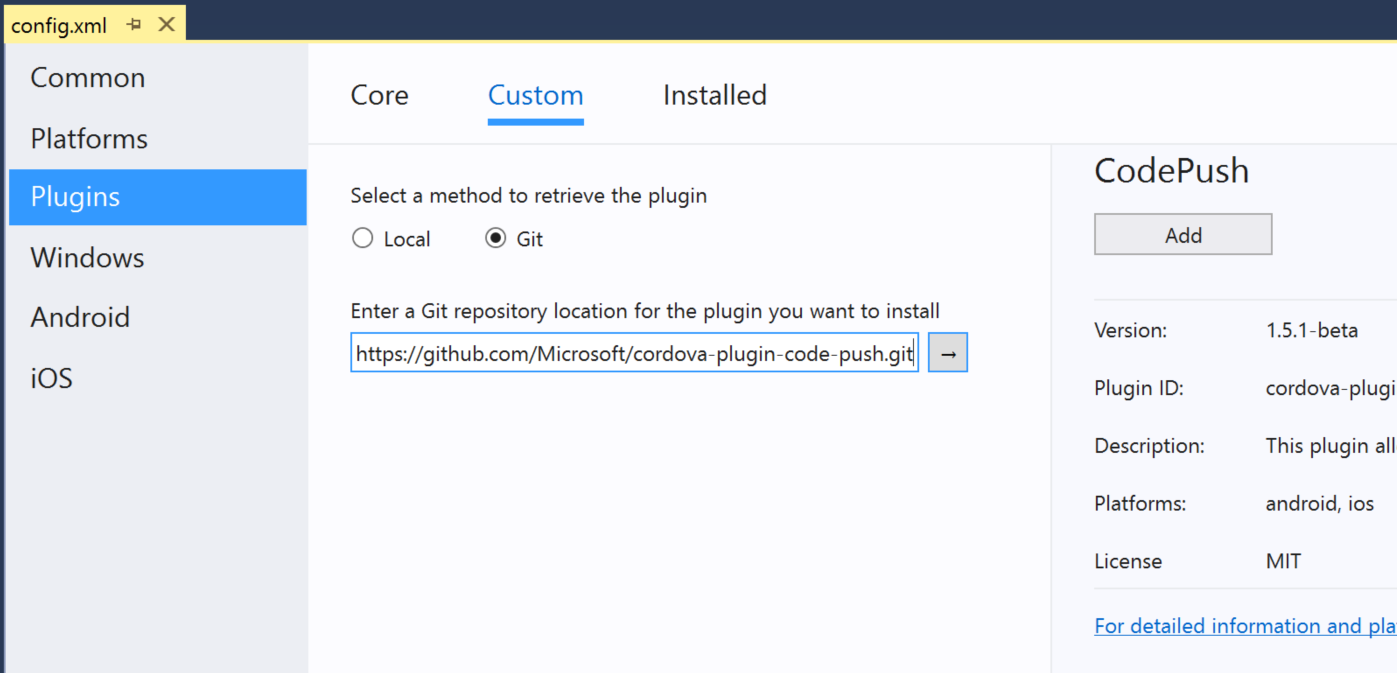
After the project is created, you’ll need to add the CodePush plugin. You can do this via the NodeJS CLI or through the Visual Studio interface.

To install it from Visual Studio, you’ll need to open the plugin interface. You can reach it by double clicking the *config.xml* file

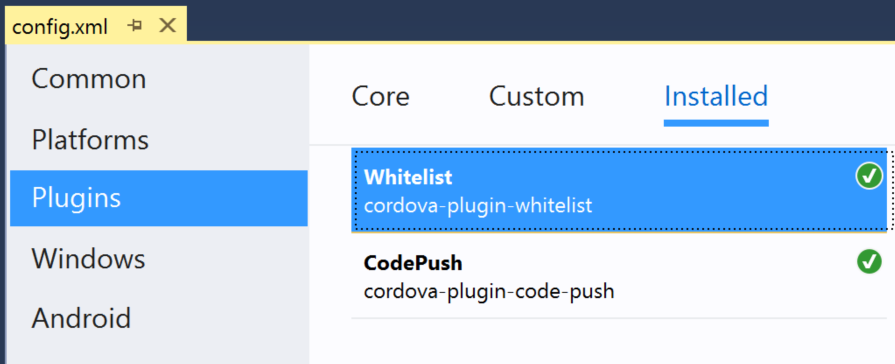


Once the config.xml viewer opens, select **Plugins → Custom → Git,** type in the location for the CodePush plugin repository, *press enter and then press**the* ***Add*** *button****.***

https://github.com/Microsoft/cordova-plugin-code-push.git



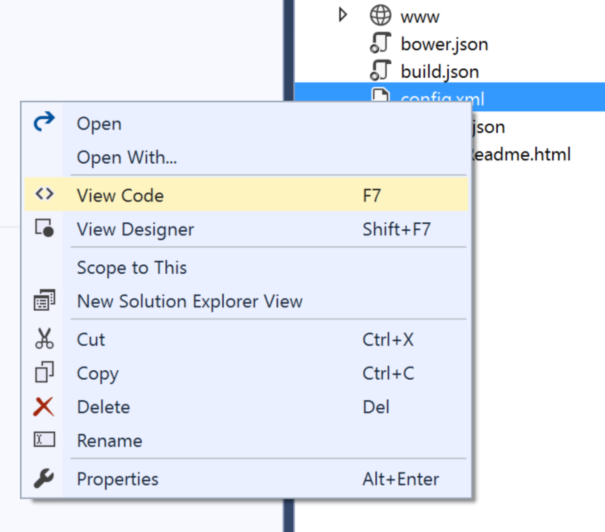
You can verify the installation was successful by visiting the **Installed** tab.



# Step 5: Configure the CodePush plugin

To configure the service, you’ll first need to add the deployment key that was provided by the CodePush CLI on step 2.

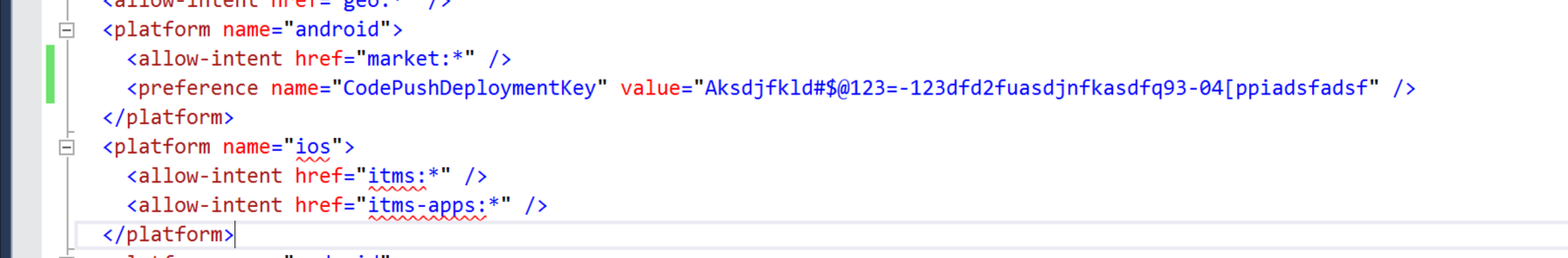
Go to *config.xml* again but this time, open the context menu by right clicking the file and select “View Code”.



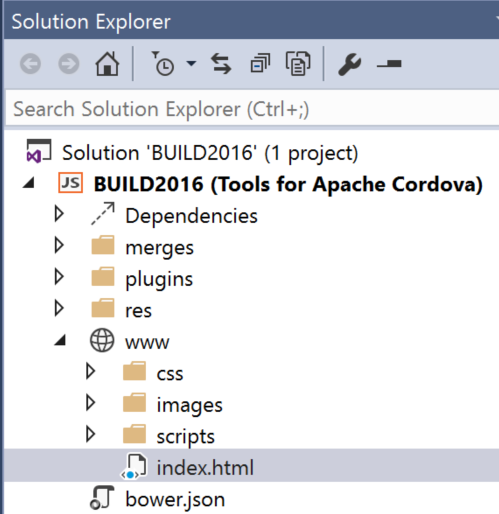
Once the code view is opened, copy the code below inside the Android platform (we’ll only test Android on this Challenge but iOS is also supported).

<preference name="CodePushDeploymentKey" value="YOUR-ANDROID-DEPLOYMENT-KEY" />

**Note**: make sure to replace “YOUR-ANDROID-DEPLOYMENT-KEY” with your deployment key.



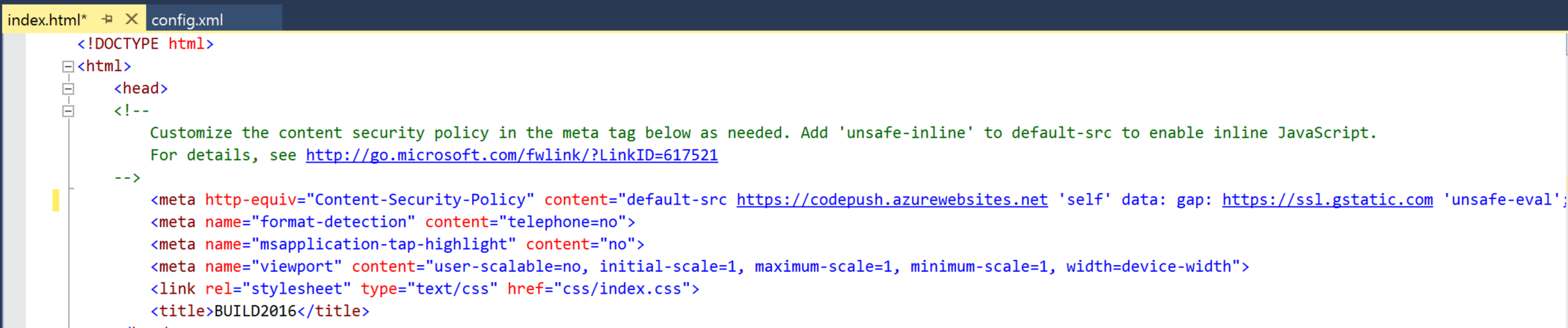
Once the key is applied, find and open the *index.html* page.



Now add *https://codepush.azurewebsites.net* to the app’s Content-Security-Policy meta tag.

For the purposes of this exercise, you can replace the existing CSP (Content Security Policy) meta tag with the one below:

<meta http-equiv="Content-Security-Policy" content="default-src https://codepush.azurewebsites.net 'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src \*" />

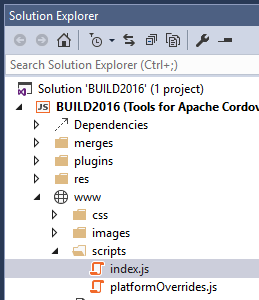


# Step 6: Configure the CodePush update experience

With the CodePush plugin installed and configured, the only thing left is to add the necessary code to configure the end user’s update experience. CodePush provides Cordova developers with multiple options to configure the end users’ update experience, but for the purposes of this code challenge, we’ll only cover “active mode” updates. You can learn more about other options by visiting the CodePush [Documentation](https://microsoft-my.sharepoint.com/personal/rurios_microsoft_com/Documents/CodePush/microsoft.github.io/code-push/docs/cordova.html#link-3).

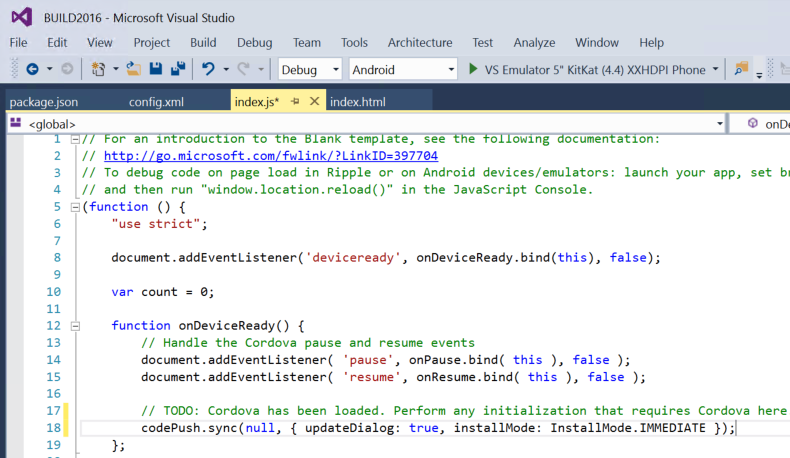
Active mode updates, are simply automatic updates that require user interaction to be completed. When an update is available, the app will automatically prompt the user for permission before downloading and installing the update. Once the user accepts, the update is silently downloaded and immediately applied.

To set up an active mode update, find and open the *index.js file*.



Find the onDeviceReady event and add the following code:

codePush.sync(null, {updateDialog: true, installMode: InstallMode.IMMEDIATE});

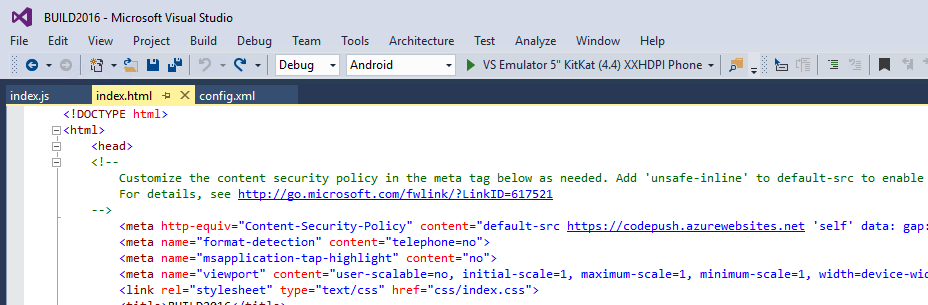


That’s it! Your app is ready to start receiving CodePush updates!

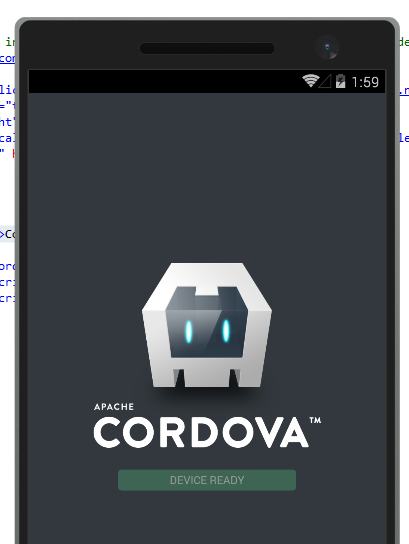
If you are curious about the code you added, the *codePush.sync()* call verifies if an update is available and proceeds to download and install it based on the parameters used. The null parameter is replacing an error callback (which was not implemented for simplicity’s sake). Finally, because of the InstallMode. IMMEDIATE flag, the update will be applied immediately after the app is restarted.

# Step 7: Create a code update and upload it to CodePush

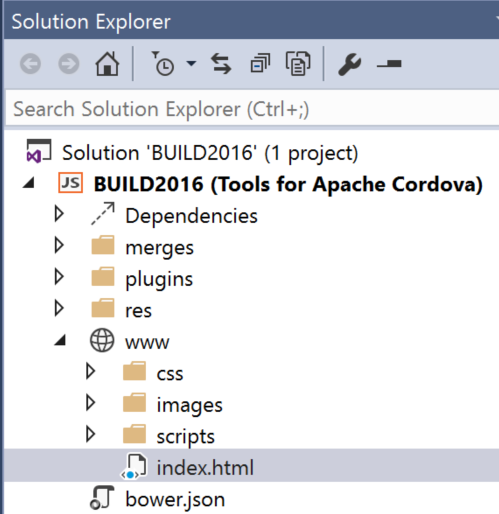
Before moving forward, test everything is working as expected by deploying the app to the Android emulator.   
  
Select Android as the “solution platform” and either press F5 or the “Play” button.



If you followed the instructions, your app should look as follows:



Open *index.html* page to change the HTML markup and create an updated version of the app.



Replace the contents of the div.app from:

<div class="app">

<p id="deviceready" class="event">Connecting to Device</p>

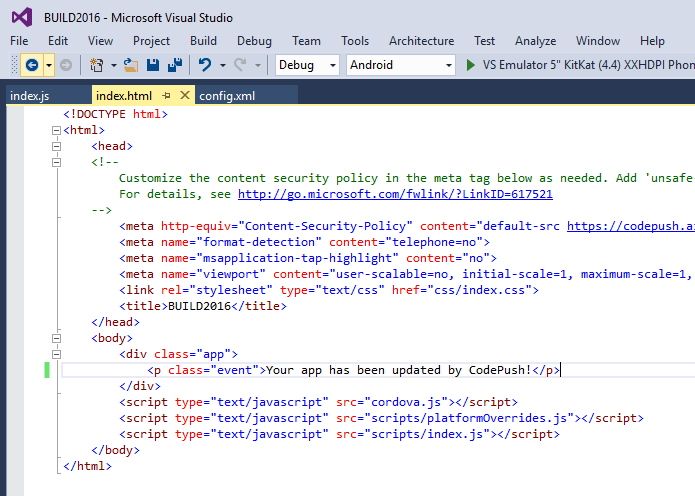
</div>

To:

<div class="app">

<p class="event">Your app has been updated by CodePush!</p>  
</div>

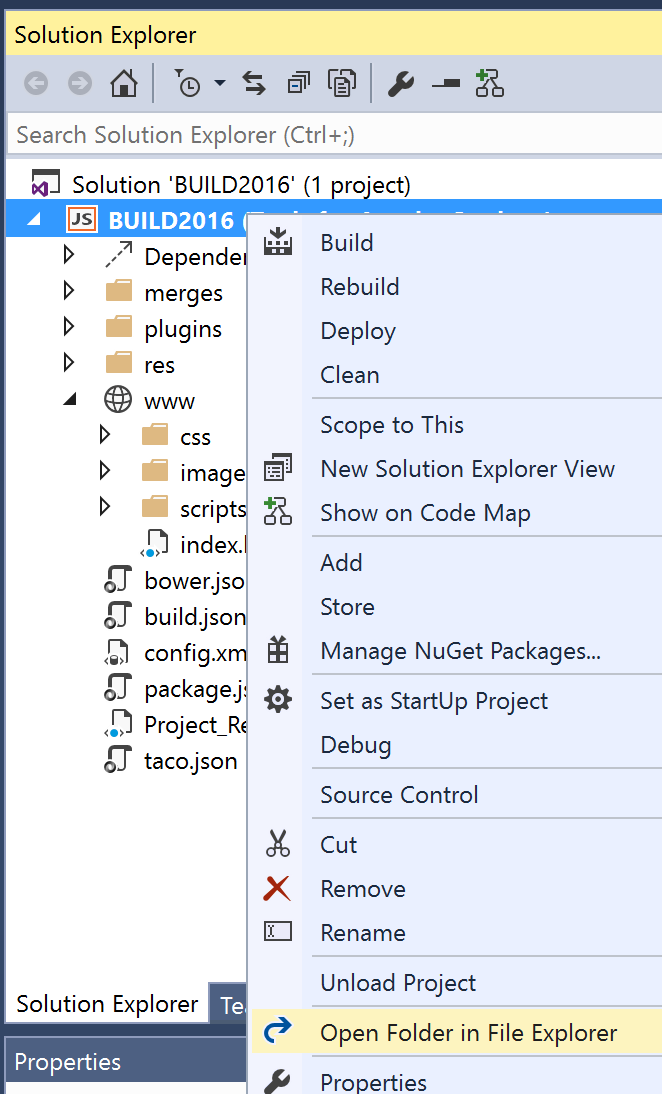
The resulting file should look as follows:



You are now ready to release your update! Open the NodeJS CLI, navigate to the root of your project and use the following CodePush CLI command to update your app:

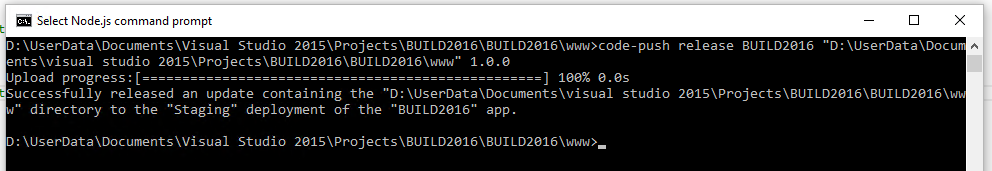
$ code-push release-cordova <appName> <platform>

**Note**: You can see the project root path by selecting the BUILD2016 node inside the Solution Explorer pane, opening its context menu and selecting “Open Folder in File Explorer”.



Since you will be targeting Android on this challenge, modify the release command as follows:

$ code-push release-cordova BUILD2016 android



Once the update is successfully released, the app is ready to receive the update!

To test the CodePush update, open *index.html* page and undo the previous changes done to the HTML markup. In short, ensure the contents of the div.app are changed from:

<div class="app">

<p class="event">Your app has been updated by CodePush!</p>  
</div>

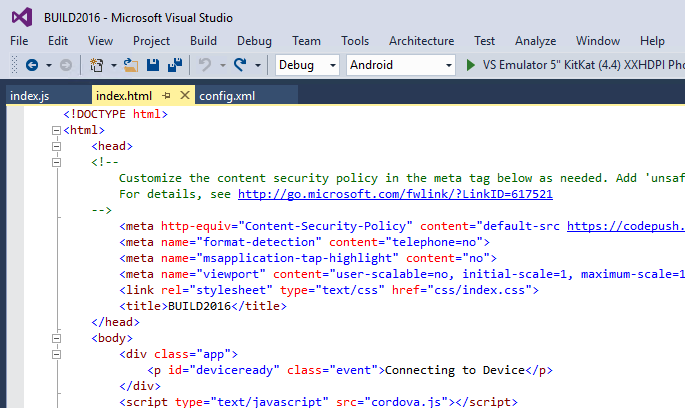
To:

<div class="app">

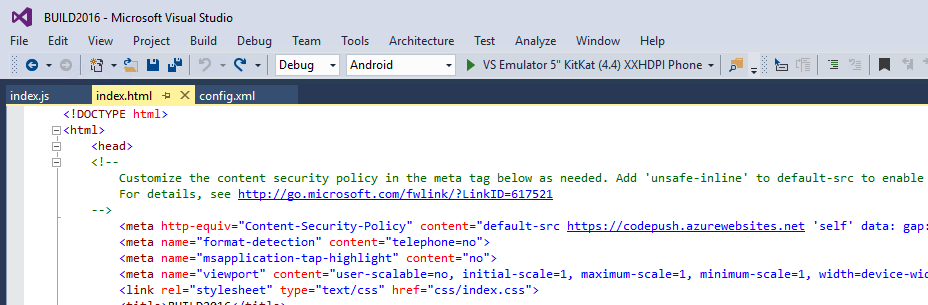
<p id="deviceready" class="event">Connecting to Device</p>

</div>

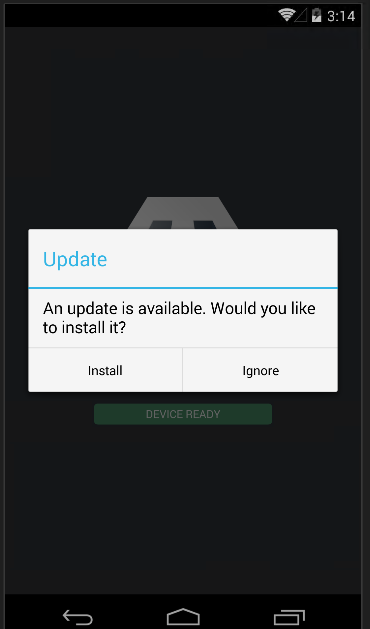
The resulting file should now look as follows:



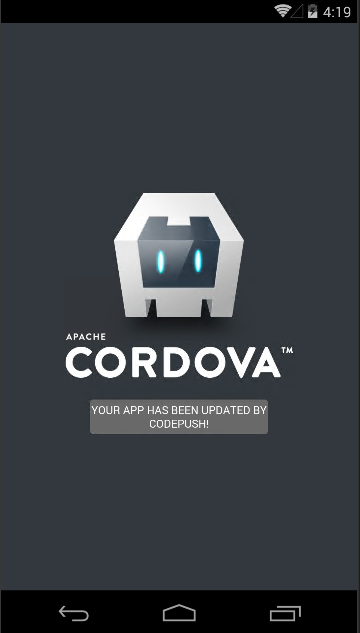
Once that’s done, re-launch the app by either pressing F5 or the “Play” button.



The app will launch and after a few seconds, a prompt will ask you to install the update.



Press Install and notice the code change.



You have successfully “CodePushed” your app! ☺

# Summary

Congratulations on completing this Quick Start Challenge! In this lab, you’ve learned how you can use CodePush to instantly update your beta and/or production Cordova apps! See the additional resources section for more information on CodePush and both the Cordova and React Native SDKs.

# Additional Resources

Refer to the following resources if you are interested in learning more about CodePush:

**CodePush website**: <http://microsoft.github.io/code-push/>

**CodePush repo:** <https://github.com/microsoft/code-push>

**CodePush Cordova plugin**: <https://github.com/Microsoft/cordova-plugin-code-push>

**CodePush React Native plugin:** <https://github.com/Microsoft/react-native-code-push>