Mobile App Installation Guide

GTNexus 2014

To be able to deploy the web based code located in the Mobile-App folder, there are a couple of steps that must be followed to be able to build the app to deploy it on the operating system of your choice. Ultimately, free open source software PhoneGap will do the majority of the heavily lifting of converting our web based source code into platform specific code.

Step 0

Before PhoneGap can be installed, freeware Node.js must be installed. Node.js is an environment that allows Javascript to be run outside of the browser. In this case, on the Phonegap cloud. It can be found at <http://nodejs.org/> . Downloading and installing it is generally straight forward.

Step 1

Once nodejs is installed, run the following command from your command prompt:

On Windows: npm install -g phonegap

On Mac or Linux: sudo npm install -g phonegap

This might take a few moments, while PhoneGap downloads and installs.

Step 2

Once phonegap is installed, follow directions here -> <http://docs.phonegap.com/en/3.5.0/guide_cli_index.md.html#The%20Command-Line%20Interface> on using the phone gap capabilities.

Step 3

Once you have followed the steps in the above URL to install Cordova, create a subdirectory, add the specific platform you wish to deploy the app on, and build, the app is ready to be deployed. Instead of trying to run it from the Cordova command line, I would suggest going into your SDK and opening up the workspace from there. It is important to note that PhoneGap will not let you add the app to a platform if the required SDK is not downloaded.

For Android, follow download from-> <http://developer.android.com/sdk/index.html>

For IOS, follow download from -> <https://developer.apple.com/xcode/downloads/>

For Blackberry, follow download -> <http://developer.blackberry.com/>

If you get errors in the platform add step, make sure to have the correct path’s added to your environment variables. For android, this requires you have apache ant added to your path as well as your Java Runtime Environment. Also, the SDK of your choice must also be attached to the path.

Step 4

The app now should be ready for deployment. Run it with an emulator or on a connected device. To edit the app or rebuild it, go to where your app has been built and replace the www folder with the updated www folder (www folder holds all of the source code for a web based app). Now, all you have to do is run -> **cordova build** -> from the command prompt and the app will be rebuilt with the new changes.

Plugins

Phone gap allows the use of plugins to access native phone features. Adding plugins is pretty straight forward and necessary if you want to access phone features such as the camera, storage, geolocation, etc. To add a plugin, enter the following command inside an already built phone gap application (in the folder containing the www/plugins/platform/etc folders) ->

**cordova plugin add <repository-of-plugin>**

Run this command this check if the plugin added correctly -> **cordova plugin ls**

More information on the plugins can be found here -> <http://docs.phonegap.com/en/3.3.0/guide_hybrid_plugins_index.md.html#Plugin%20Development%20Guide>

Here is a list of some of the most popular and useful available plugins. This information can be found in the Phone gap documentation under API reference, along with examples on how to use each of the plugins.

Name Repository Location

Accelerometer org.apache.cordova.device-motion

Camera org.apache.cordova.camera

Capture org.apache.cordova.media-capture

Compass org.apache.cordova.device-orientation

Connection org.apache.network-information

Contacts org.apache.cordova.contacts

Device org.apache.cordova.device

Battery org.apache.cordova.battery-status

File org.apache.cordova.file

Geolocation org.apache.cordova.geolocation

Globalization org.apache.cordova.globalization

InAppBrowser org.apache.cordova.inappbrowser

Media org.apache.cordova.media

Notification org.apache.cordova.dialogs + org.apache.cordova.vibration

Splashscreen org.apache.cordova.splashscreen

For an example of an android built app, check out the Android-App branch on the repository to view the build of the To do List App.