Ionic/Cordova Setup

Download the latest stable Node.js version (Includes NPM):

Version x.x was used for this tutorial

Download latest stable Java SDK:

Version x.x was used for this tutorial

Download latest stable Android SDK:  
<https://developer.android.com/studio/index.html#win-bundle>

Include Android 8.0 (Oreo), API Level 26 for the current version of Cordova

Version x.x was used for this tutorial

<https://stackoverflow.com/questions/46557775/how-to-fix-partially-installed-sdk-android-studio>

Open Node.js command prompt as Administrator

Setup NPM Proxy connections:  
(This must be completed each time you open the command line if you need to download or install packages)  
>npm config set proxy "http://##ADID##:##PASSWORD##@proxy.njc.ups.com:8080/"  
>npm config set https-proxy http://##ADID##:##PASSWORD##@proxy.njc.ups.com:8080/

Install Ionic & Cordova

>npm install -g cordova ionic

Add Cordova and Ionic to your env variables by including this in the PATH

>C:\Users\YOUR\_ADID\AppData\Roaming\npm

\*\*this is assuming this is where they have their –g (global) packages from npm installing to (which is the default)

Create a new Cordova Project:  
(The highlighted portions should be relevant to your project. The first portion will be the C: drive location)  
>cordova create grid com.example.grid FirstGrid

Update config.xml as needed (Don’t break it!)

Configure target (Android and iOS) deployments:  
cd into project directory (i.e.: cd grid <enter>)  
>cordova platform add android  
>cordova platform add ios

Add packages to access device features/sensors:  
(Yellow warnings are normal, and will display when compatibility packages are skipped for older versions)  
  
[Device Information](https://www.npmjs.com/package/cordova-plugin-device)  
 >cordova plugin add cordova-plugin-device  
 Allows for the polling of device and platform information  
  
[Flashlight](https://www.npmjs.com/package/cordova-plugin-flashlight)  
 >cordova plugin add cordova-plugin-flashlight  
 Allows the application to toggle on/off the phone’s camera flash as a flashlight

[Camera](https://www.npmjs.com/package/cordova-plugin-camera-with-exif)   
 >cordova plugin add cordova-plugin-camera  
 Allows the application to open the device’s camera to take a picture that also returns the geo   
 location data (if available)

[Barcode Scanner](https://www.npmjs.com/package/cordova-plugin-barcodescanner)  
 >cordova plugin add phonegap-plugin-barcodescanner   
 Allows application to open the camera, scan a barcode, and return the value as a string

[Geo Location](https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/)  
 >cordova plugin add cordova-plugin-geolocation  
 Allows access to the device’s latitude and longitude  
[Notifications](https://github.com/phonegap/phonegap-plugin-local-notification/blob/master/docs/api.md)  
 Allows the application to access the notification bar from the background

> cordova plugin add phonegap-plugin-local-notification

[Dialogs](https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-dialogs/index.html)  
 Allows the user of System Dialogs (Pop-up notifications/prompts from the OS)

> cordova plugin add cordova-plugin-dialogs

[File System](http://cordova.apache.org/docs/en/latest/reference/cordova-plugin-file/index.html)

Allows the application to access the OS file system. Note that the directory structure varies by device.

> cordova plugin add cordova-plugin-file

File System

Long Press

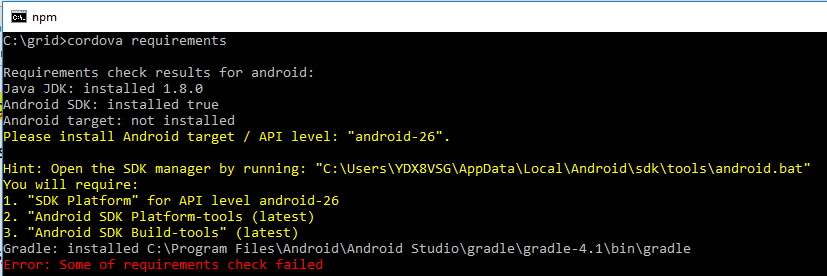
<https://www.npmjs.com/package/ionic-long-press>  
npm install --save ionic-long-press

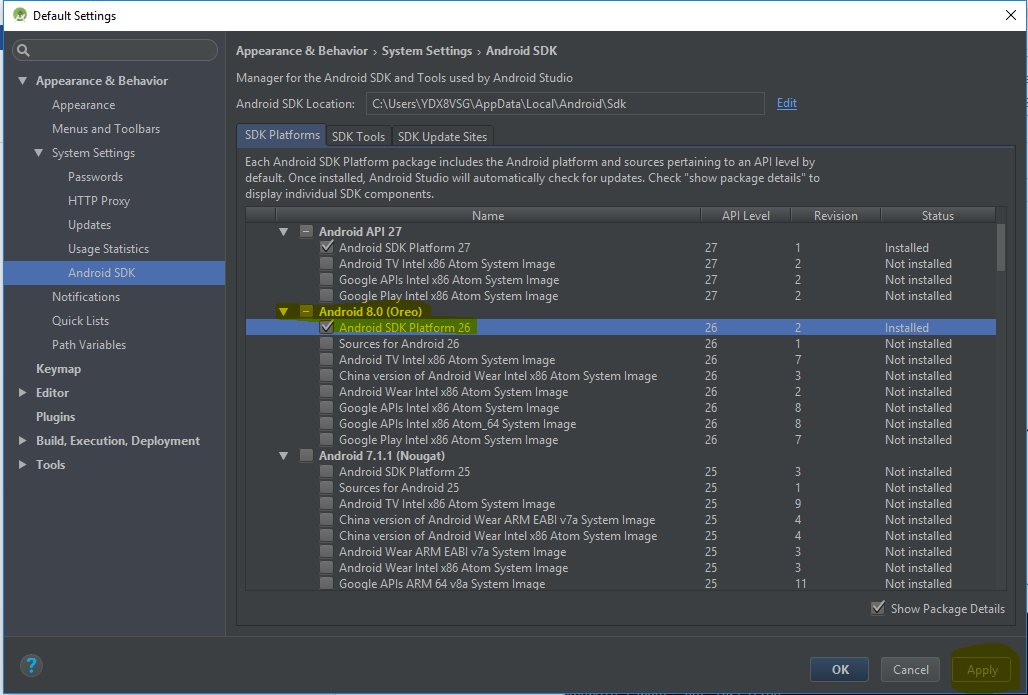
Stop watch / Clock timer

Check Prerequisites and requirements for target platforms:  
>cordova requirements

At this step, if there are any warnings (such as android version), they will have to be addressed in the relevant SDK.

For example:





It’s also possible to run into an “Android Target: Not Installed” error. In this case, Cordova is not able to see the proper Android SDK version. First, confirm that the correct SDK version is installed. If that is not the issue, attempt to reboot and confirm your Path directories (ANDROID\_HOME and PATH) in Environment Variables. PATH should include

Environment Variables

* JAVA\_HOME -> Java SDK
  + i.e. C:\Program Files\Java\jdk1.8.0\_131
* ANDROID\_HOME -> Android SDK
  + i.e. C:\Users\{{ADID}}\AppData\Local\Android\Sdk
* PATH -> Key runtimes
  + JAVA SDK
    - i.e. C:\Program Files\Java\jdk1.8.0\_131\bin
  + npm & gradle (may be automatically added during installation)
    - i.e. C:\Users\{{ADID}}\AppData\Roaming\npm
    - i.e. C:\gradle\bin
  + Android Tools/Platform Tools
    - i.e. C:\Users\YDX8VSG\AppData\Local\Android\Sdk\tools
    - i.e. C:\Users\YDX8VSG\AppData\Local\Android\Sdk\platform-tools

Assemble your application

Test via emulator:  
 > cordova emulate android  
  
Deploy to device:  
 > cordova run android  
Note: USB should be connected before running this command. If this is the first time building to that device, you will have to accept the connection prompt  
(If you have issues building a new version to the device, try uninstalling the application first)  
  
If you encounter an [INSTALL\_FAILED\_UPDATE\_INCOMPATIBLE] failure:  
 run “>adb uninstall com.example.grid”

Build APK:  
 > cordova build android

Change Application’s Display Name:

In Package.json, change

"displayName": "[Kitchen Sink]"

Debugging

One useful feature of developing using Cordova is the ability to use remote debugging over USB in Chrome’s Developer Tools.

Related Article:  
<https://gonzalo123.com/2014/08/04/debugging-android-cordovaphonegap-apps-with-chrome/>

1. Connect the phone via USB
2. Launch the application on the phone
3. Open Chrome and navigate to chrome://inspect
4. Select “Inspect” on the connected device

You will now be able to see the device screen mirrored in the Chrome window, complete with access to the inspector tools. You can control the application from either the device or the browser.

Git Integration  
  
The Framework is available via Git. You will need access to the following Git project:  
P08AGit\_Mobile\_Framework

First, you will need to install the Git CLI from here

Once installed, create a project directory and run the following command:

> git clone <https://tfs.ups.com/tfs/UpsProd/_git/P08AGit_Mobile_Framework>

(You may be prompted for your ADID & password)

Git Issues

If you change your password and you get the fatal: authentication failed issue, fix it by using the git command found here:

<https://stackoverflow.com/questions/37182847/how-to-disable-git-credential-manager-for-windows>

And to resetup your username and password store:

<https://stackoverflow.com/questions/5343068/is-there-a-way-to-skip-password-typing-when-using-https-on-github>

Note: You should exclude your [www] folder from your project’s git repository, to avoid older builds from causing confusion on another workstation. A clean build should always be created after the code is retrieved from source control.

Next you need to run npm install which will download all of the node\_modules necessary for the project. \* Make sure your proxy settings are configured for this step!

> npm install

Ionic Application Setup

> npm install -g ionic  
> ionic cordova build  
 > cordova [run/build] [android/browser/ios]

Angular Application Setup

Angular is a populate Single Page Application framework that is approved for use at UPS. Single Page Applications have a performance advantage when compiled to run from the device using a wrapper such as Cordova. While data fetching will still come over the wire, the core layout is loaded when the application is first opened. Additionally, Angular monitors for data changes using {{handlebars}} which allows real-time visual updates for any values defined in your application’s JavaScript layer.

Note that it is difficult to convert existing applications to use Angular templates. The Kitchen Sink is being developed with Angular to show its advantages. If you already have an application that is tightly coupled between the business layer and front-end (such as .NET), you would first have to separate all web service calls and validations before replacing the GUI.

Follow [THIS GUIDE](https://www.becompany.ch/en/blog/2016/10/19/creating-apache-cordova-app-with-angular2) to merge Angular with your Cordova application.

Todo: Move the necessary steps into this guide

Important Note: If your application will not start on android, try this fix:

In index.html remove the line:  
<base href="/">

Build Notes:

When building a new version of the Angular app, you must first build the project from the angular directory, building into the www directory

We need to install Angular CLI globally on your machine

> npm install –g @angular/cli@1.4.2

Once is this complete, you will be able to run the following commands to compile and run/deploy:

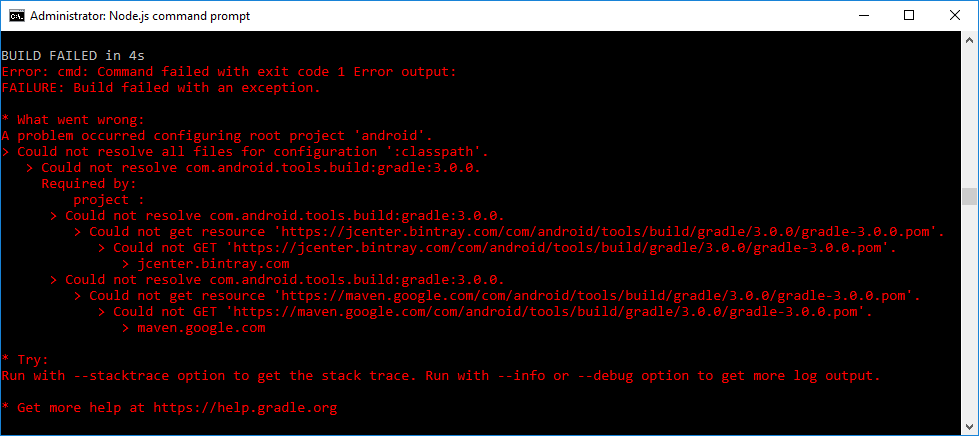
> ng build  
 > cordova [run/build] [android/browser/ios]

Proxy Issues:

At this stage in development, there is one step that requires the UPS Laptop be taken off the proxy. The mobility team is working on a fix for this. The information below are the related notes.

Requires Gradle?  
(<https://services.gradle.org/distributions/gradle-3.3-all.zip>)

Requires Gradle Proxy?  
(<https://medium.com/@petehouston/execute-gradle-behind-a-corporate-proxy-network-509260961813>)



**Both HTTP and HTTPS Proxy configuration**

gradle -Dhttp.proxyHost=proxy.njc.ups.com -Dhttp.proxyPort=8080 Dhttp.proxyUser=us\ydx8vsg -Dhttp.proxyPassword=##pass## -Dhttps.proxyHost=proxy.njc.ups.com -Dhttps.proxyPort=8080 -Dhttps.proxyUser=us\ydx8vsg -Dhttps.proxyPassword=##pass##

C:\Users\{{USER\_HOME}}\.gradle  
gradle.properties

systemProp.http.proxyHost=proxy.njc.ups.com

systemProp.http.proxyPort=8080

systemProp.http.nonProxyHosts=localhost|127.0.0.1|10.10.1.\*

systemProp.http.proxyUser=US/{{user}}

systemProp.http.proxyPassword={{pass}}

systemProp.https.proxyHost=proxy.njc.ups.com

systemProp.https.proxyPort=8080

systemProp.https.nonProxyHosts=localhost|127.0.0.1|10.10.1.\*

systemProp.https.proxyUser=US/{{user}}

systemProp.https.proxyPassword={{pass}}

(DETERMINE IN-LINE COMMAND)

Requires Java Proxy?

(<https://www.java.com/en/download/help/proxy_setup.xml>)

In Line Commands:

java -DproxySet=true -DproxyHost=proxy.njc.ups.com -DproxyPort=8080 -DproxyUser=ydx8vsg -DproxyPassword=XXXXXXX