

**First Install node.js**, find it here: <https://nodejs.org/en/download/>

### **Install Cordova.**

Open a terminal window and execute the following commands:

NOTE: IF you are installing for iOS, you will have to provide your password for the 'sudo' command, you probably won't have to provide it again. Sudo is the 'super user', or administrator level of commands. You may also not need to issue 'sudo' before the next npm command either. The 'npm' command stands for 'node package manager'. It is used here with cordova to install packages needed to run your apps.

Install the IOS simulator:

```
sudo npm install -g ios-sim
```

For windows:

```
npm install -g
```

### **To create a Cordova app:**

```
cordova create myFirstApp com.projects.myFirstApp ISTE-app01
```

The first argument specifies the 'ex directory to be generated for your project. Its www subdirectory houses your application's home page, along with various resources under css, js, and img, which follow common web development file-naming conventions.

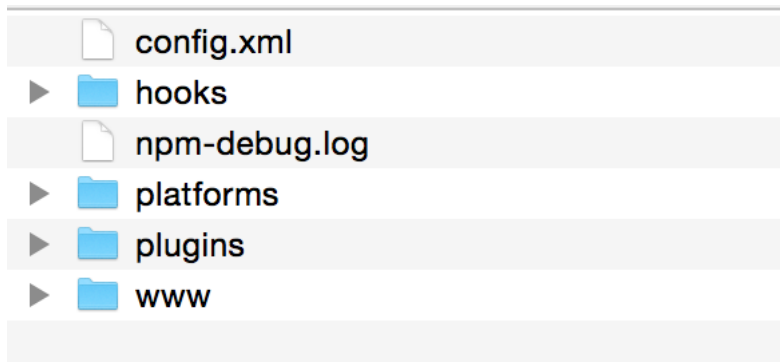
A config.xml is created that contains important metadata needed to generate and distribute the application.

The other two arguments are optional: the **com.projects.myFirstApp** argument provides your project with a reverse domain-style identifier, and the **ISTE-app01** provides the application's display text. You can edit both of these values later in the config.xml file if you did not include them in the create command.

All subsequent commands need to be run within the project's directory, or any subdirectories within its scope. So change to the myFirstApp directory:

```
cd myFirstApp
```

You now have a director structure as pictured below:



1. **config.xml** contains the app's configuration and behavior. API features, plug-ins and platform specific settings can be configured here. The file comes with a lot of pre-configured details and is in a ready to go state. The access element should be configured properly to allow network traffic to make API calls as well as to receive data. Further details like author, application display name, contact details and version numbers can also be mentioned in this file.
2. **npm-debug log**: warning, errors, and good messages from the create command.
3. **hooks** are pieces of code that are executed through the application build lifecycle. Code is arranged in folders that are named according to the point they belong to in the build lifecycle.
4. **platforms** all the code will be build in their respective platforms and the installers will be located in this folder locally. YOU SHOULD TREAT THESE DIRECTORIES AS READ ONLY!!! Each platform in the platforms folder will have its own local www folder, which is used to create platform specific builds. The local www folder should not be changed!!!
5. **plugins** All the plug-ins that are added to the project are kept in this folder.
6. **www** is the root folder that contains all the resource files with the naming convention of a web application.

Before you can build the project, you need to specify a set of target platforms. Your ability to run these commands depends on whether your machine supports each SDK, and whether you have already installed each SDK.

For mac users: be sure you have XCode installed on your MAC.

For Android Users be sure to install the Android SDK

## **Add Platforms:**

The following commands can be run on a MAC:

```
cordova platforms add ios
cordova platforms add android
cordova platform add browser
```

For Windows users you can try the following commands:

```
cordova platform add windows
cordova platform add browser
```

Run either of the following synonymous commands to remove a platform:

```
cordova platform remove android
cordova platform rm android
```

## **Build and Run the app.**

You can run the '**prepare**' without specifying the platform and Cordova will generate code for all platforms that have been installed (added), however, running the prepare command with specific platforms seems to make the iOS platform more successful.

```
cordova prepare ios    or    cordova prepare android
```

Run the Emulator:

```
cordova emulate ios    or    cordova emulate android
```

An emulator should open with your app running.

Emulate on the browser:

```
Cordova emulate browser    or    cordova prepare browser
```

For Mac users wanting to run their apps in XCode: Once created, you can open the project from within Xcode. Browse to the www folder in your ios platform and double-click the file named myFirstApp.xcodeproj file.

---

To run the app on your device:

```
cordova run ios -device    OR    cordova run android -device
```