Cordova - Guide - Install & Setup

- Dr Nick Hayward

A brief overview of install and setup for Cordova v.6.x onwards.

Platform specific guides,

- Android install and setup - see `android-platform-guide` document
- iOS install and setup - see `ios-platform-guide` document

Contents

- Install
  - Update versions & plugman
- Check Cordova requirements
- Create basic app
- Add platforms
- Build app
- Emulate app

Install Cordova

- `Node.js` is required for install and general usage.
- `Git` is also useful for general development and maintenance.

Then install Cordova using Node's **NPM**,

```
npm install -g cordova
```

n.b. on OS X and Linux it may be necessary to prepend `sudo` to the above command. This is often due to permissions on the `/usr/local/share` directory.

Update versions & plugman

Once we've installed and setup Cordova, we can update the core using the following command,

```
npm install -g cordova@latest
```

Then, we can install and update the Plugin manager for Cordova, **Plugman**, using the following command,

```
npm install -g plugman@latest
```

Check Cordova requirements

These checks and requirements are per platform, for example for development with Android, iOS &c.

Initially, we can check required and installed dependencies for Cordova with the following command in a directory containing a Cordova app,

```
cordova requirements
```

This command will check build requirements for each install platform, e.g. Android and iOS.

Create basic app

To test the new Cordova install, we can now create a simple initial app

```
cordova create test com.example.test TestApp
```

This command will create a new app with its own basic, shell template for the app's underlying structure. The home page for this app will be stored in `www/index.html` .

Add platforms

For a chosen project, we can then add required platforms for our application. These are the mobile OSs the app needs to run on, and Cordova will support. We can add our chosen platforms with the following commands from within the root directory of our Cordova app,

```
cordova platform add android --save
cordova platform add ios --save
```

We can then check all current install platforms with the following command,

```
cordova platform ls
```

Each platform added to a project is stored in the app's `/platforms/` directory. These files are maintained and updated as part of the build and update process run by Cordova for the current app.

Build app

After creating our basic test app, we can now build it from the directory created for the app.

```
cordova build
```

This command will build the current app for the installed platforms. However, we may also wish to limit the scope of a given build to a specific platform, e.g.

```
cordova build android
```

Emulate app

We can also build our app, and then open it for testing with an installed emulator

```
cordova emulate android
```

n.b. this assumes that you have correctly installed and setup Android or iOS, and required emulators, on your local machine. If not, consult

- Android, see `android-platform-guide` document
- iOS, see `ios-platform-guide` document

The emulate command will choose the default installed emulator for the chosen platform. However, we may also wish to add a flag for a specific emulated device,

```
cordova emulate android --target=Nexus_6P_API_23
```

This will load the app in the specific emulator for the Nexus 6P.