

# Getting Started with Apache Cordova

---

## Setting up your environment

You'll need to start by following the general instructions for setting up your development environment for mobile found in our [Environment Guide](#). Since Cordova takes advantage of web technologies, you won't need any specific additional tools beyond what you would use for web development.

## Installing Cordova

The installation process for Cordova is a single command:

```
``sh npm install -g cordova // or yarn global add cordova``
```

## Writing your application

Cordova is a great way to build a mobile app when you already have an existing web application (designed to be mobile-friendly).

You can also use it for new projects - just build your web application as you normally would and add the Cordova project wrapper around it.

*To bring in Material Design components and Brightlayer UI themes to your application, follow our Getting Started Guides for [Angular](#) or [React](#).*

## Setting up a Cordova wrapper around your project

Start by creating a folder that will hold your entire project:

```
```sh mkdir myprojectfolder cd myprojectfolder ```
```

Then, create a new cordova wrapper project.

```
```sh cordova create cordova your.app.id AppName ```
```

This will create a /cordova folder that will contain all of the necessary code to create the mobile app. You can see the [Cordova Documentation](#) to learn more about the parameters for the create command.

From here, you will need to bring in the code for your web application. If you have an existing project, you can copy the root folder into the /myprojectfolder you created above (it should be at the same level as the /cordova folder). If you are starting a project from scratch, you can run one of the following commands in the /myprojectfolder directory, depending on your selected framework:

```
```sh create-react-app appname ng new appname ```
```

The resulting folder structure should look like this:

```
``` └── /myprojectfolder |── /cordova └── /appname ```
```

## Building your application

Cordova works by placing your application code (bundled and minified, as it would be for production) into the /cordova/www directory. We can automate this process to simplify the development process. Inside of your /appname directory, you should have a package.json file. In the "scripts" section, add an entry for the following:

For Angular projects:

```
```json "build-cordova": "ng build --prod --base-href . --output-path ../cordova/www/"
```
```

For React projects (Windows):

```
```json "build-cordova": "react-scripts build && (robocopy .\\build\\ ..\\cordova\\www\\ /e /purge ^& exit 0)" ```
```

For React projects (Linux / Mac):

```
```json "build-cordova": "react-scripts build && rsync -aq ./build/ ../cordova/www/ --delete"```
```

If you are building a React app and your developers are using a mix of operating systems, you may want to include both versions of this script and give them different names (e.g., ``build-cordova-windows`` and ``build-cordova-linux``).

Once you have added this script, you can build your app and have the files moved to the `/www` directory with a single command:

```
```sh cd appname npm run build-cordova // or yarn build-cordova```
```

You can then add your desired platforms in your cordova project, and test them with the following commands:

```
```sh cd cordova cordova platform add cordova run```
```

For more information about the various Cordova commands, check out the [Quick Start Guide](#) or the [API Reference](#).

When you are ready to create a final build, you'll use the build command:

```
```sh cordova build```
```

## Using plugins

There are many different plugins available for use in Cordova applications that will allow you to access various native / hardware features of the device that your application is running on. To add a plugin to your app, navigate to the `/cordova` directory and install it:

```
```sh cd cordova cordova plugin add```
```

To use plugins in your application, you will need to add the `cordova.js` file as a script in your `index.html` file (or equivalent):

```
```html```
```

This file is automatically created and added at the root level by the build process, so you do not need to modify the path in the `src` attribute.

Then, make sure that the script is available before loading your application:

- For Angular projects, add the DOM manipulation code to a `cordova.service.ts` file:

```
``` let onDeviceReady = () => ; document.addEventListener('deviceready',  
onDeviceReady, false); ```
```

- For React projects, update your index.js file (or equivalent root file) to look like the following:

```
``` const startApp = () => ;  
  
if(!window.cordova) else ```
```

The plugins will then be available from the cordova object (window.cordova.``.``). Refer to the specific plugin documentation for proper usage.

## License Information

[Cordova](#) is available under the Apache 2.0 License.