

Ionic Tutorial

For Cross-Platform Mobile Software Development

About

This Tutorial is for setting up a basic hybrid mobile application using the Ionic framework. The setup will be shown for both Mac and PC computers using a virtual Android device.

Step 1: NodeJS

The first step requires us to download and install NodeJS. These steps differ on a Mac and PC. Here is how we install NodeJS on both machine types:

Mac requirements:

Download and install Xcode (<https://itunes.apple.com/us/app/xcode/id497799835?mt=12>). Xcode is used for testing, running, and deploying IOS apps. Xcode can only be used on a Mac.

Next, we need to download and install Homebrew. Open a terminal and type:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

This will automatically install Homebrew. Homebrew will help us install NodeJS, which helps us install Ionic's dependencies.

Next, we need to install NodeJS. Type

```
brew install node
```

at the command line and hit enter.

PC requirements:

Go to NodeJS's website (<https://nodejs.org/en/>) and download the version of NodeJS that best fits your machine (32 or 64 bit).



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

Important [security releases](#), please update now!

Download for Windows (x64)

8.9.1 LTS

Recommended For Most Users

9.2.0 Current

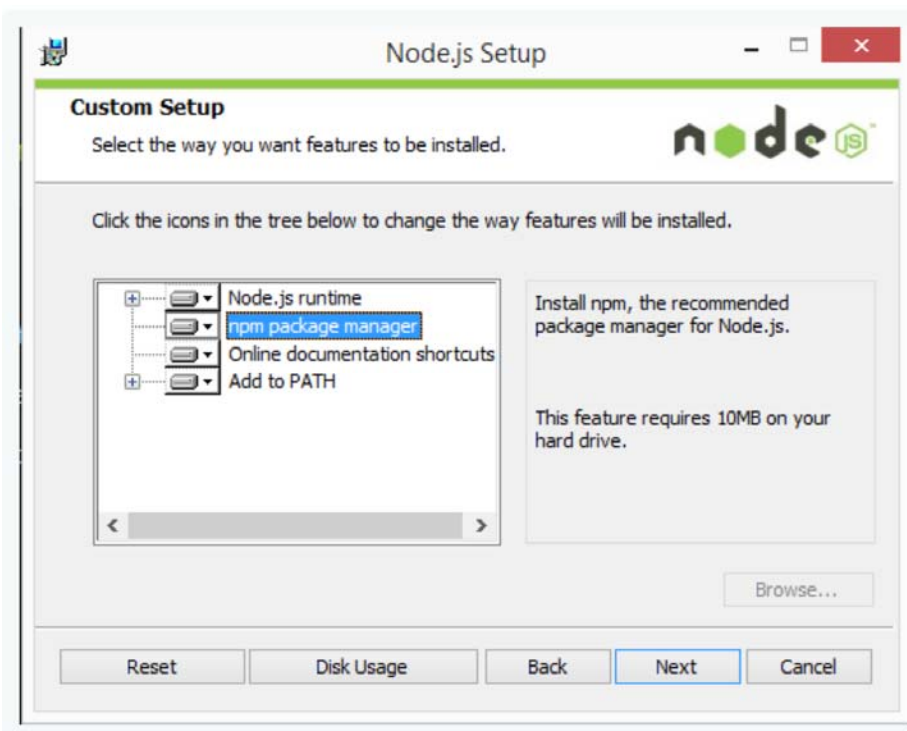
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) | [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [LTS schedule](#).

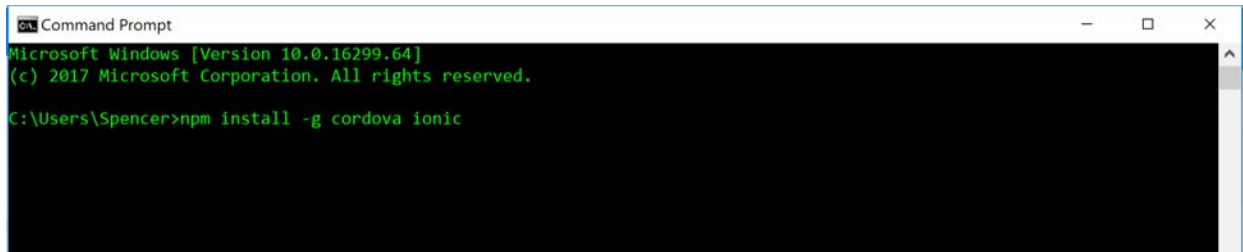
Sign up for [Node.js Everywhere](#), the official Node.js Weekly Newsletter.

This will download an installer. Open the installer and use the default settings



Step 2: Ionic-Cordova

Open up a terminal. Type `npm install -g cordova ionic` and hit enter.



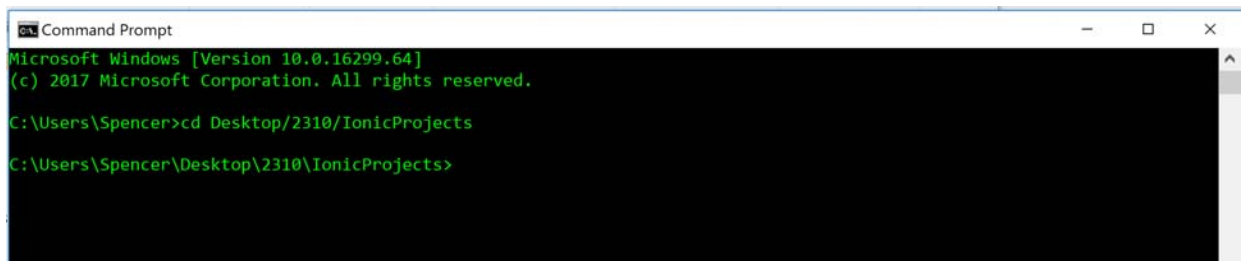
```
Command Prompt
Microsoft Windows [Version 10.0.16299.64]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Spencer>npm install -g cordova ionic
```

This uses NodeJS's package manager to install all the dependencies that Ionic needs to run. It also installs all of Cordova's dependencies. Cordova builds the application from your JavaScript source code into IOS, Android, or Windows phone applications.

Step 3: Create an App from a template

Create a new folder somewhere on your computer called IonicProjects. This directory is just to hold any and all of your IonicProjects as you develop. Open a terminal in this new directory.

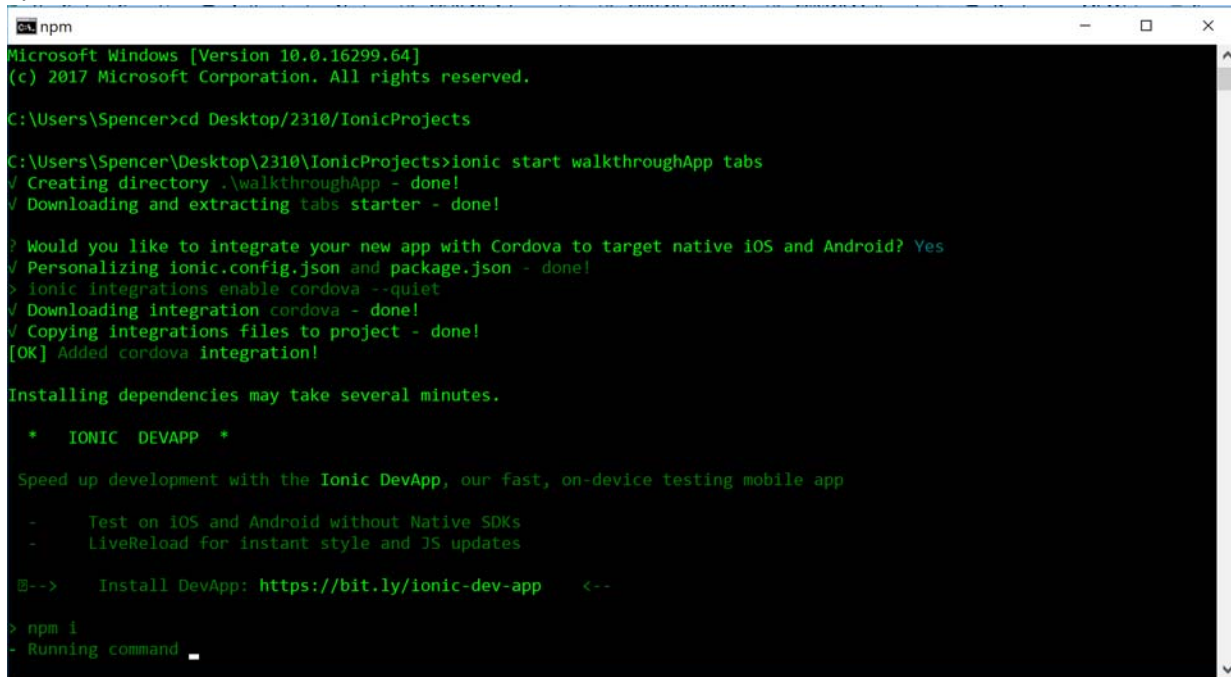


```
Command Prompt
Microsoft Windows [Version 10.0.16299.64]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Spencer>cd Desktop/2310/IonicProjects
C:\Users\Spencer\Desktop\2310\IonicProjects>
```

We now have 3 options. We can create a blank app from no template, create an app with 3 tabs at the bottom for different pages, or an app with a side menu with 3 sample pages. For our example, we will use the tabs template. At your terminal, type `ionic start walkthroughApp tabs`. When prompted, type

“y” so we can build for iOS and Android with Cordova:



```
npm
Microsoft Windows [Version 10.0.16299.64]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Spencer>cd Desktop/2310/IonicProjects

C:\Users\Spencer\Desktop\2310\IonicProjects>ionic start walkthroughApp tabs
√ Creating directory .\walkthroughApp - done!
√ Downloading and extracting tabs starter - done!

? Would you like to integrate your new app with Cordova to target native iOS and Android? Yes
√ Personalizing ionic.config.json and package.json - done!
> ionic integrations enable cordova --quiet
√ Downloading integration cordova - done!
√ Copying integrations files to project - done!
[OK] Added cordova integration!

Installing dependencies may take several minutes.

*   IONIC DEVAPP   *

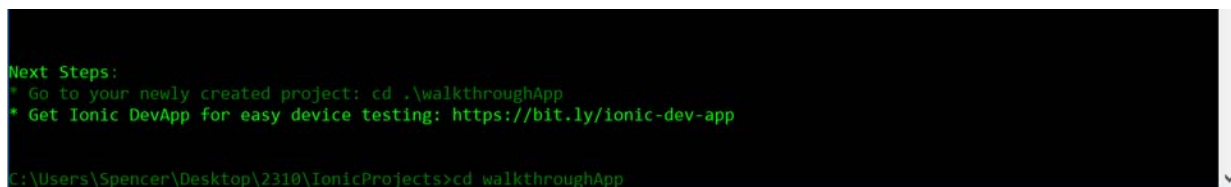
Speed up development with the Ionic DevApp, our fast, on-device testing mobile app

-   Test on iOS and Android without Native SDKs
-   LiveReload for instant style and JS updates

@-->   Install DevApp: https://bit.ly/ionic-dev-app   <--

> npm i
- Running command _
```

It will take some time for this command to run. When it is done, it will have generated a new folder called “walkthroughApp”, which you should change directory into:



```
Next Steps:
* Go to your newly created project: cd .\walkthroughApp
* Get Ionic DevApp for easy device testing: https://bit.ly/ionic-dev-app

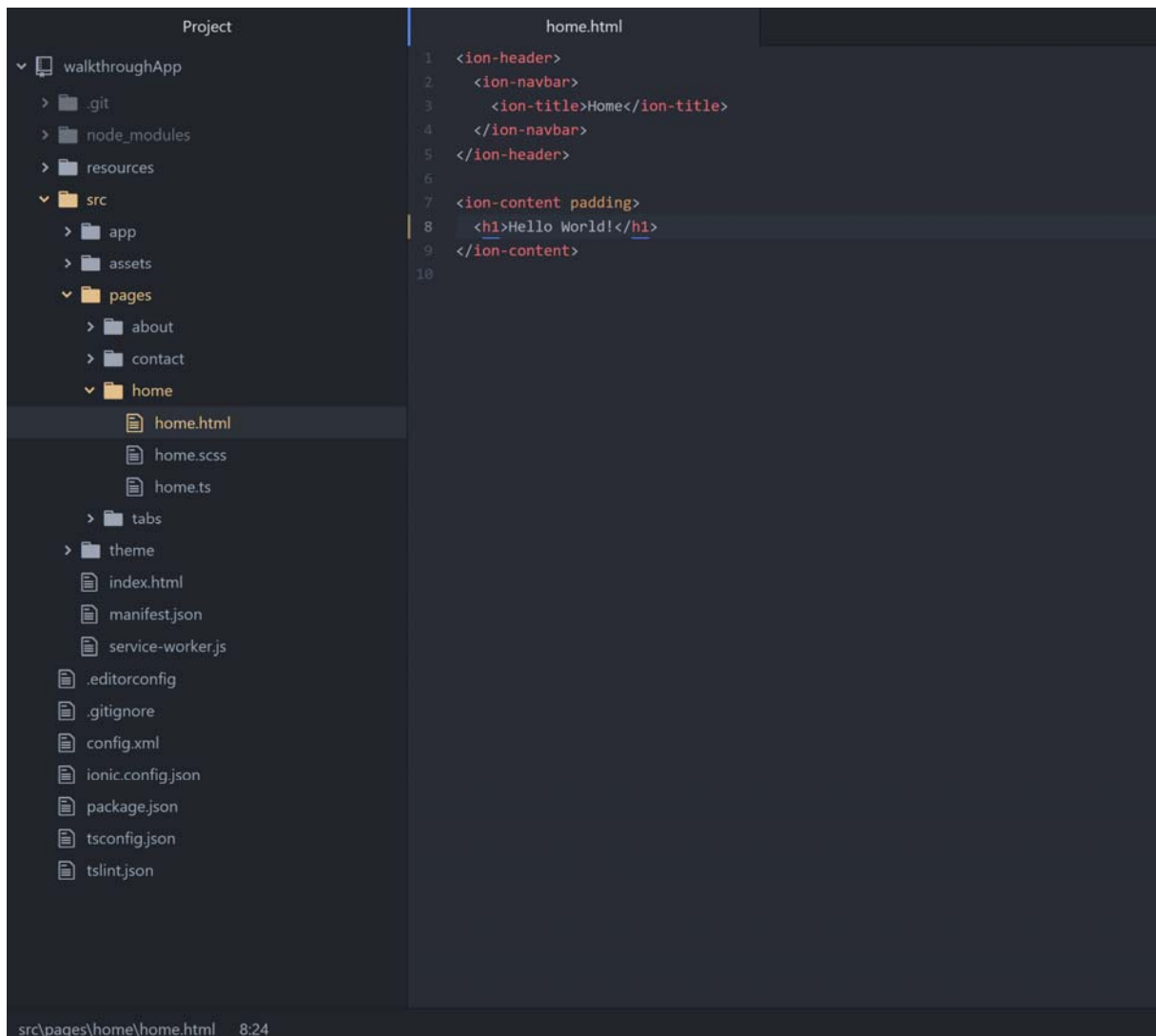
C:\Users\Spencer\Desktop\2310\IonicProjects>cd walkthroughApp
```

Pick your favorite text editor or IDE to open the project folder “walkthroughApp”. For this tutorial, I will be using Atom, which can be downloaded at <https://atom.io/>. Whichever text editor you use, open the file src/pages/home/home.html:

The image shows a code editor interface with two main panels. The left panel, titled "Project", displays a file tree for a project named "walkthroughApp". The tree includes folders for ".git", "node_modules", "resources", "src", "pages", "theme", and "tabs". The "home" folder under "pages" is expanded, showing files "home.html", "home.scss", and "home.ts". The right panel, titled "home.html", shows the HTML code for the home page. The code is as follows:

```
1 <ion-header>
2   <ion-navbar>
3     <ion-title>Home</ion-title>
4   </ion-navbar>
5 </ion-header>
6
7 <ion-content padding>
8   <h2>Welcome to Ionic!</h2>
9   <p>
10    This starter project comes with simple tabs-based layout for apps
11    that are going to primarily use a Tabbed UI.
12  </p>
13  <p>
14    Take a look at the <code>src/pages/</code> directory to add or change tabs,
15    update any existing page or create new pages.
16  </p>
17 </ion-content>
18
```

Change the contents of the home page to simply say "Hello World!"



The screenshot shows an IDE interface with a project explorer on the left and a code editor on the right. The project explorer shows a tree view of the project structure, including folders like .git, node_modules, resources, src, app, assets, pages, about, contact, home, tabs, and theme. The code editor displays the content of the home.html file, which is a simple HTML structure using Ionic framework components.

```
Project                               home.html
├── walkthroughApp
│   ├── .git
│   ├── node_modules
│   ├── resources
│   └── src
│       ├── app
│       ├── assets
│       └── pages
│           ├── about
│           ├── contact
│           └── home
│               ├── home.html
│               ├── home.scss
│               └── home.ts
│       ├── tabs
│       └── theme
│           ├── index.html
│           ├── manifest.json
│           ├── service-worker.js
│           ├── .editorconfig
│           ├── .gitignore
│           ├── config.xml
│           ├── ionic.config.json
│           ├── package.json
│           ├── tsconfig.json
│           └── tslint.json
└── src\pages\home\home.html 8:24
```

```
1 <ion-header>
2   <ion-navbar>
3     <ion-title>Home</ion-title>
4   </ion-navbar>
5 </ion-header>
6
7 <ion-content padding>
8   <h1>Hello World!</h1>
9 </ion-content>
10
```

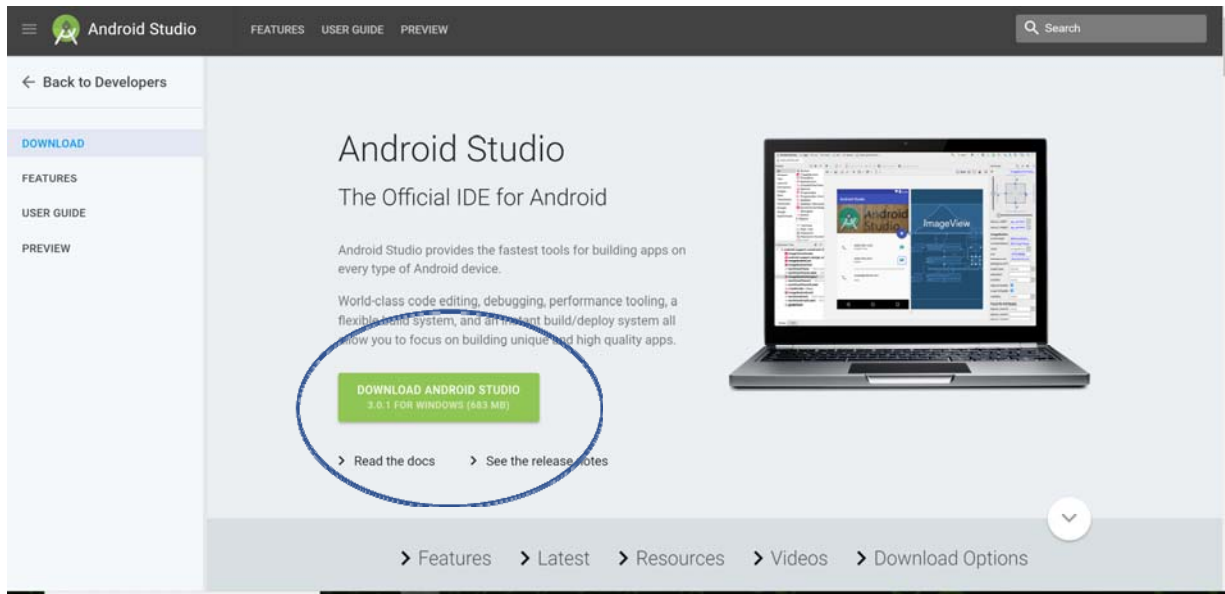
Next, I will show you how to run the app we created.

Step 4: Set up Android Studios and Virtual Device

In order to run your app, we need a device. If you are using a Mac and have an Iphone, you can deploy it on your actual phone using Xcode. If you have an Android device, you can deploy it on your actual phone with Android Studios on both Mac and PC. For this tutorial, we will use a virtual Android device on a PC. Details on the other deployment methods can be found here:

<https://ionicframework.com/docs/intro/deploying/>.

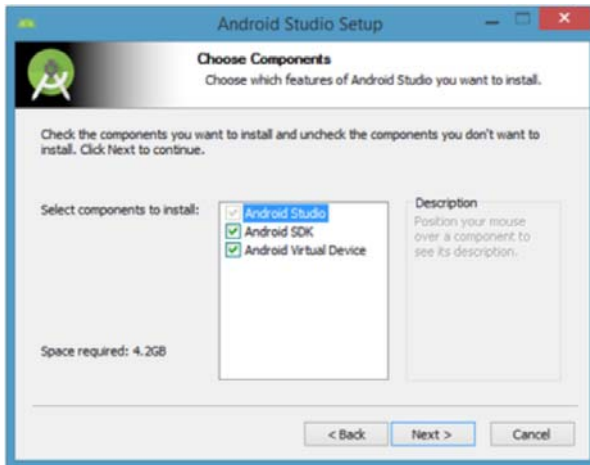
To start, let's download Android Studios (<https://developer.android.com/studio/index.html>):



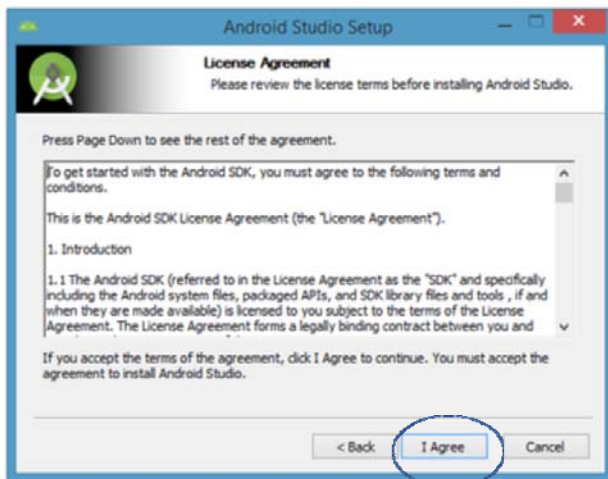
The screen for Mac will look slightly different, but the process is similar. Download the file and then run the Installer.



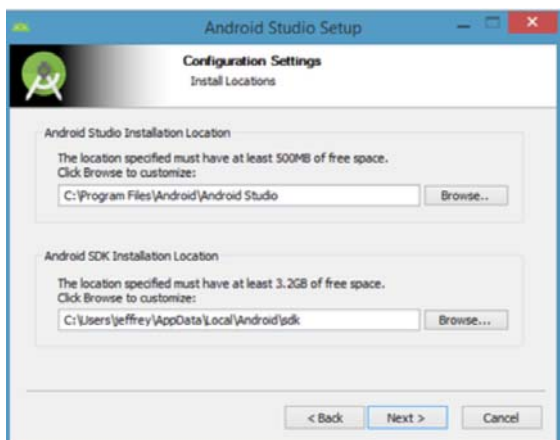
Click through the Installer



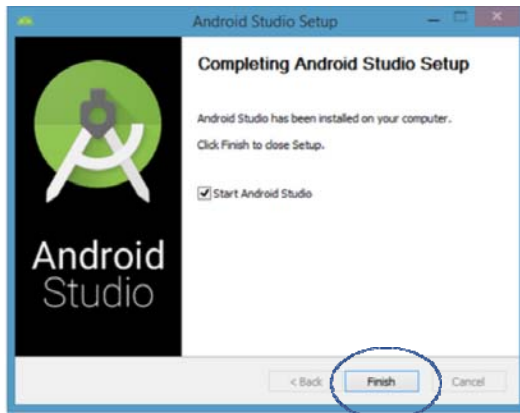
Use the default settings. Note: Android Studio will take up a lot of space on your machine.



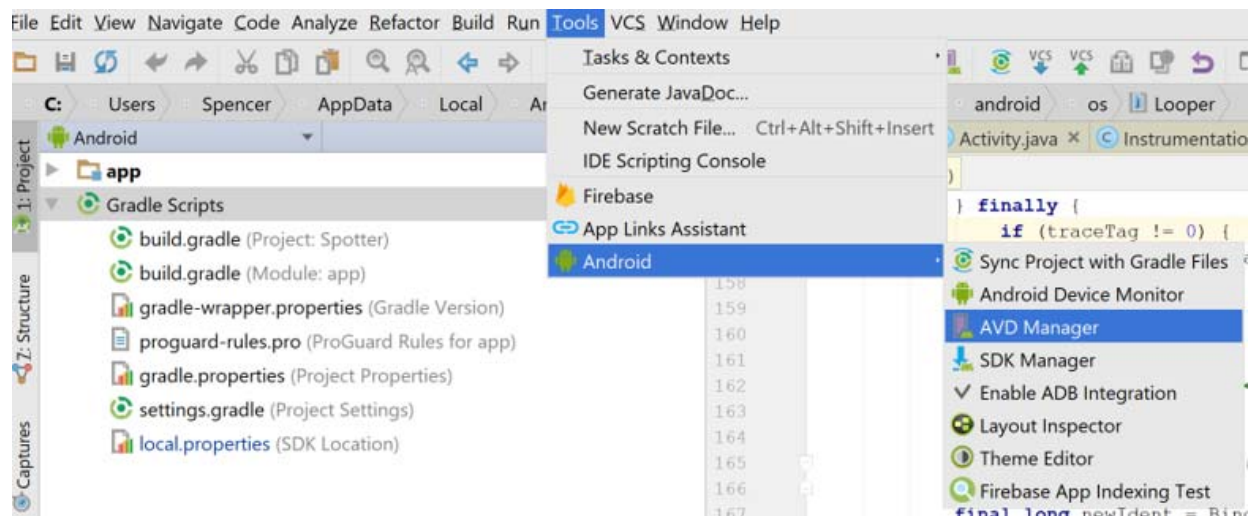
Agree to the terms.



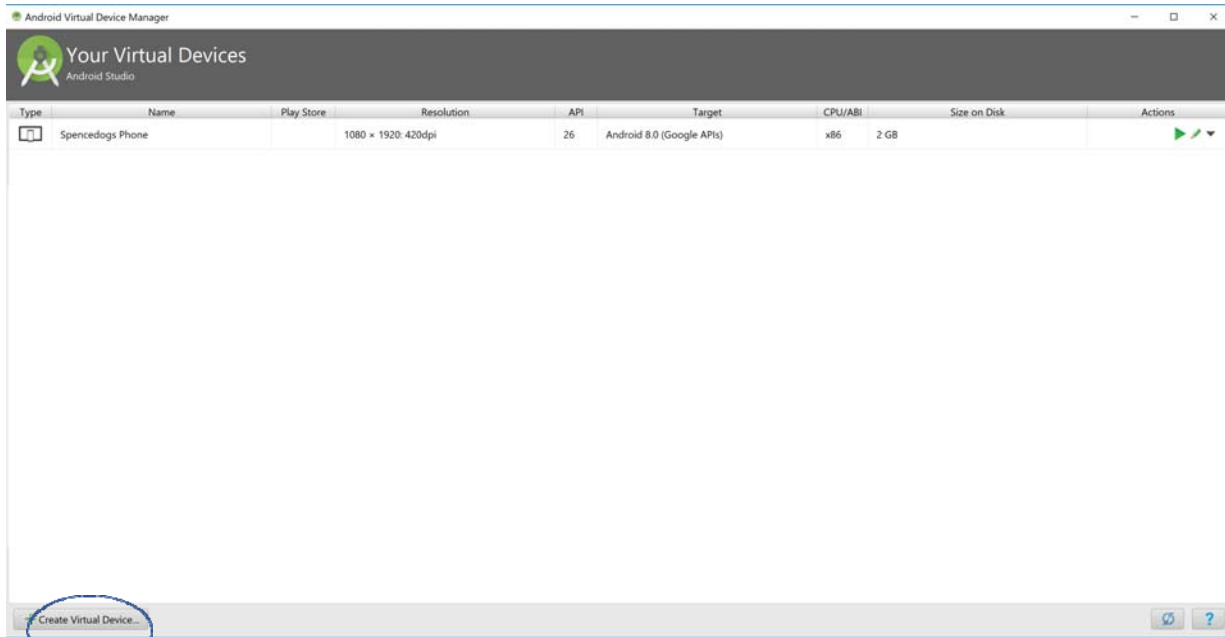
Use the default locations for the program files.



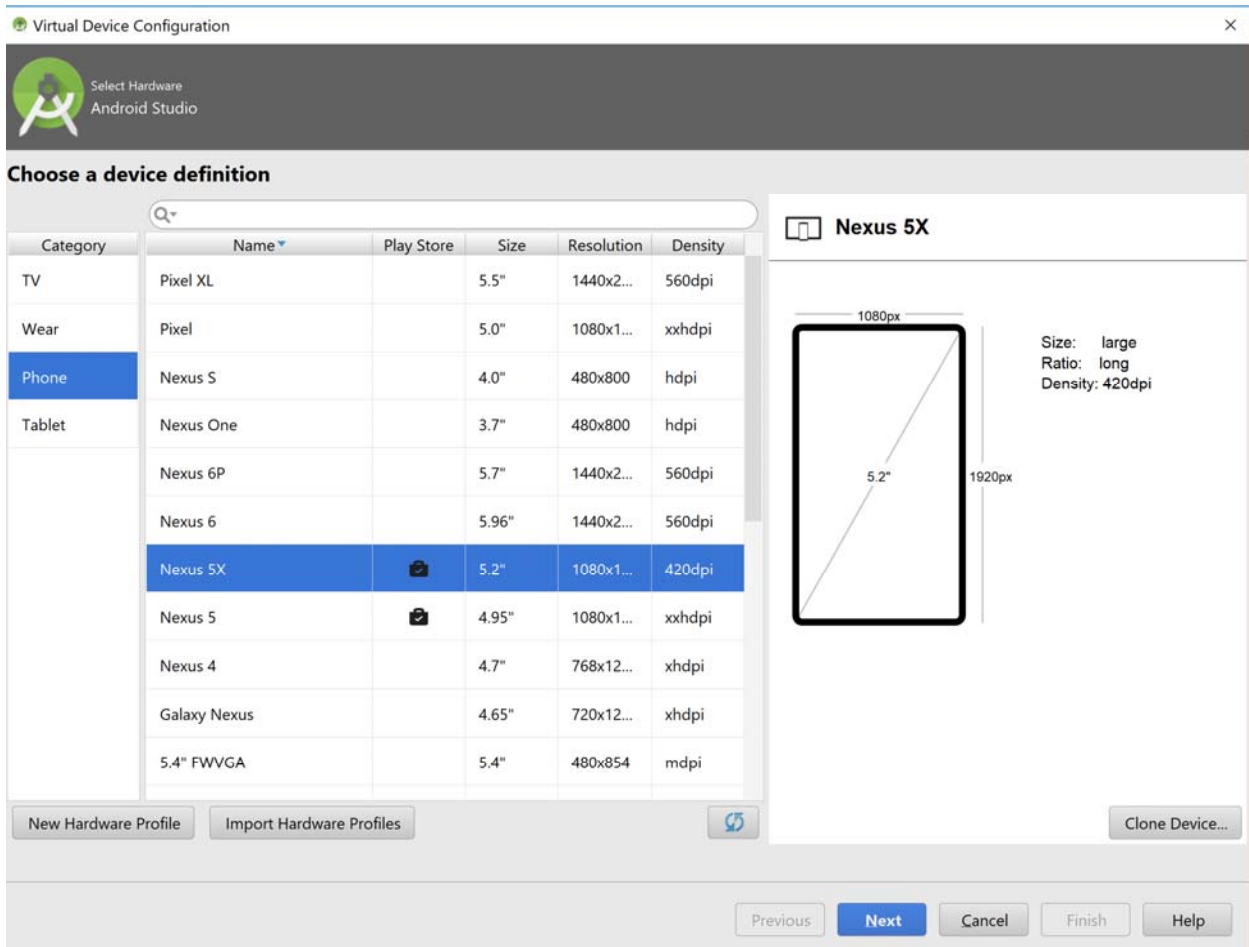
Next, we need to create a virtual device. Open Android Studio and on the top bar select Tools -> Android->AVD Manager



Next, select the Create Virtual Device Button



Next, select the Nexus 5X



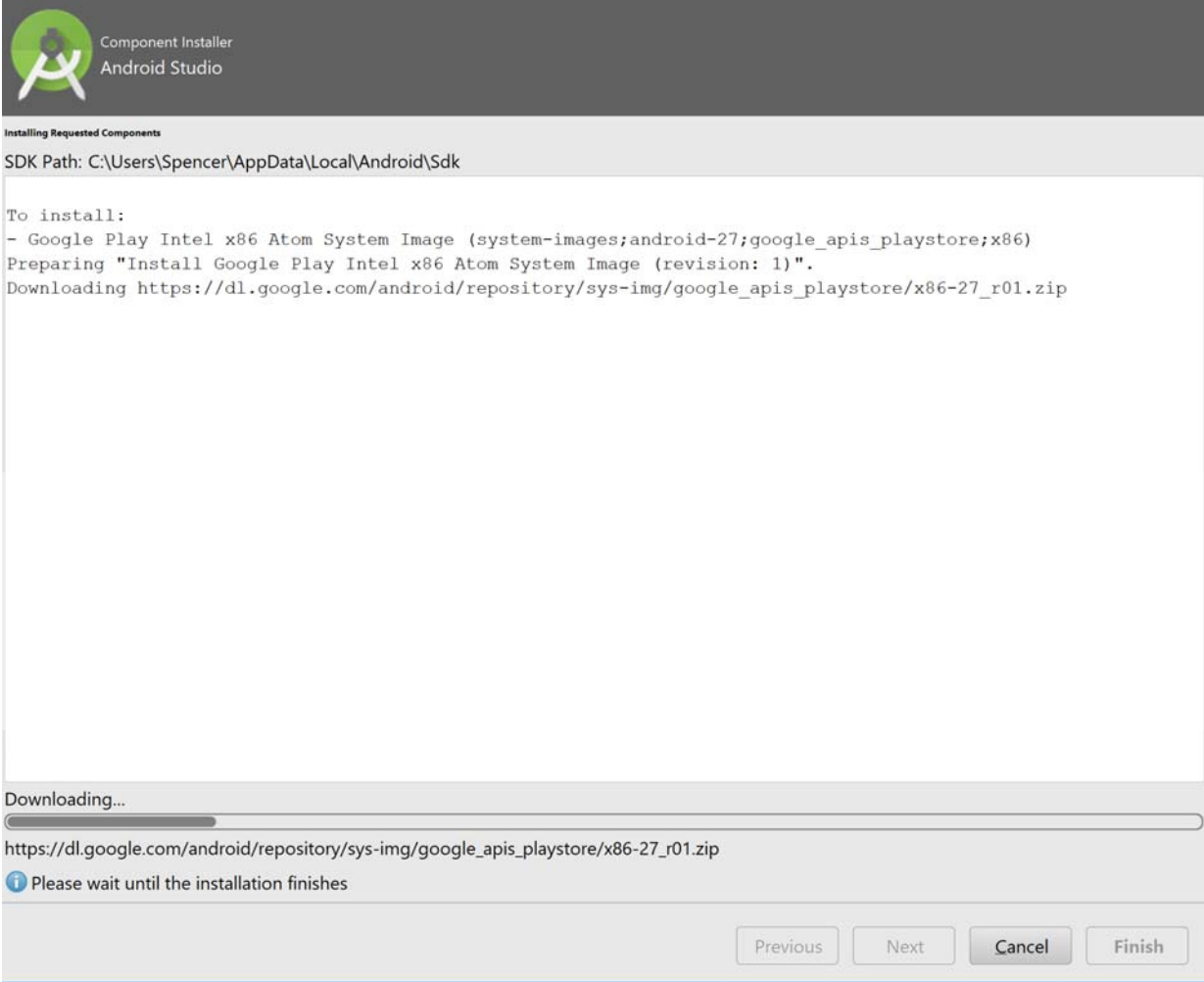
Select the newest version and download it

Select a system image

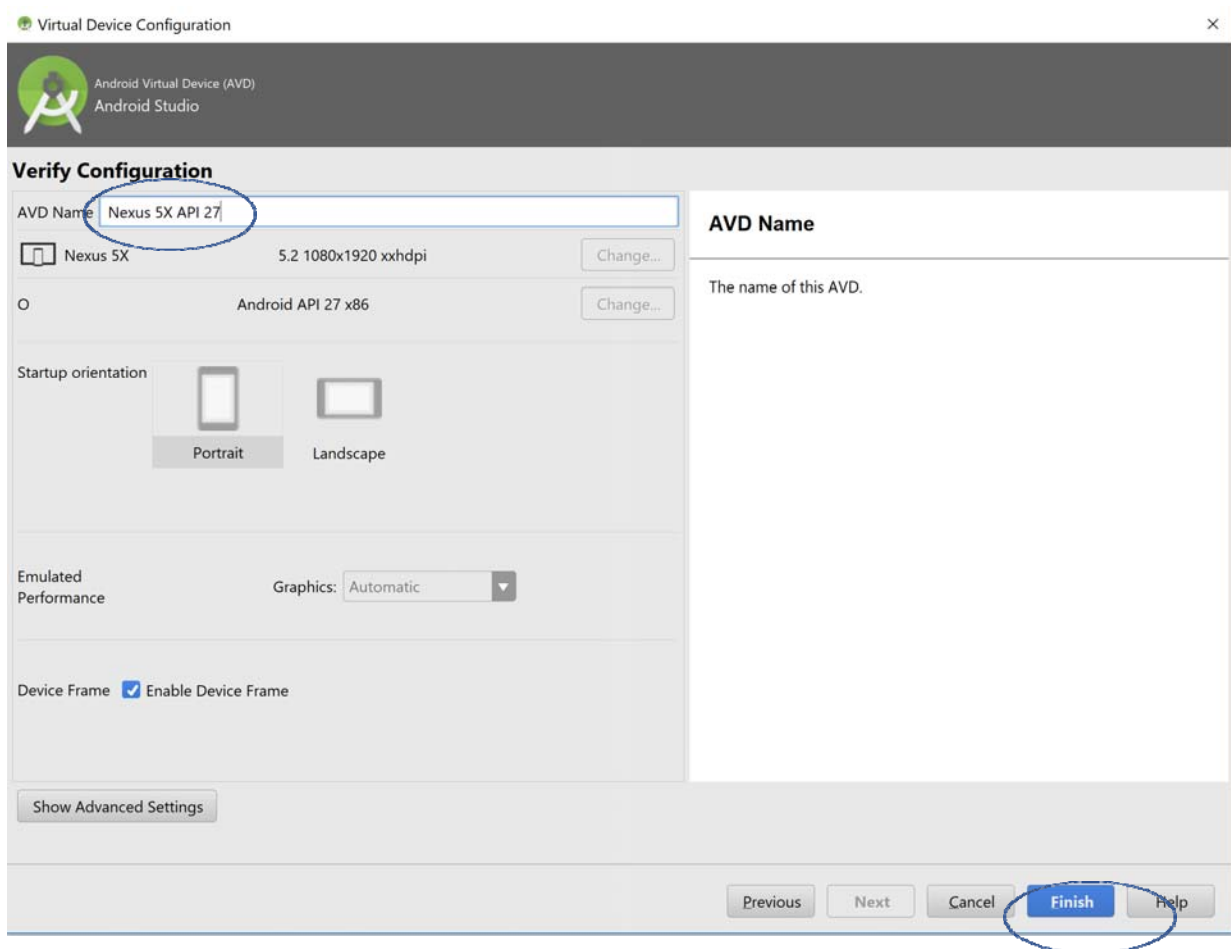
Recommended x86 Images Other Images

Release Name	API Level ▼	ABI	Target
<i>null</i> Download	27	x86	<i>Android API 27 (Google Play)</i>
O Download	26	x86	<i>Android 8.0 (Google Play)</i>
Nougat Download	25	x86	<i>Android 7.1.1 (Google Play)</i>
Nougat Download	24	x86	<i>Android 7.0 (Google Play)</i>

It will take a minute or two to download



When the download is complete, click Finish. Click on the Image you just downloaded and use the default settings. Don't forget to give your Device a name before clicking finish again.



Step 6: Run the App

Finally! We are able to build and run the app. Open a terminal in your `IonicProjects/walkThroughApp/` directory. At the terminal, type `ionic cordova emulate android -c -l -s`. This command builds your app and deploys it to the virtual device that you just created. The options are as follows

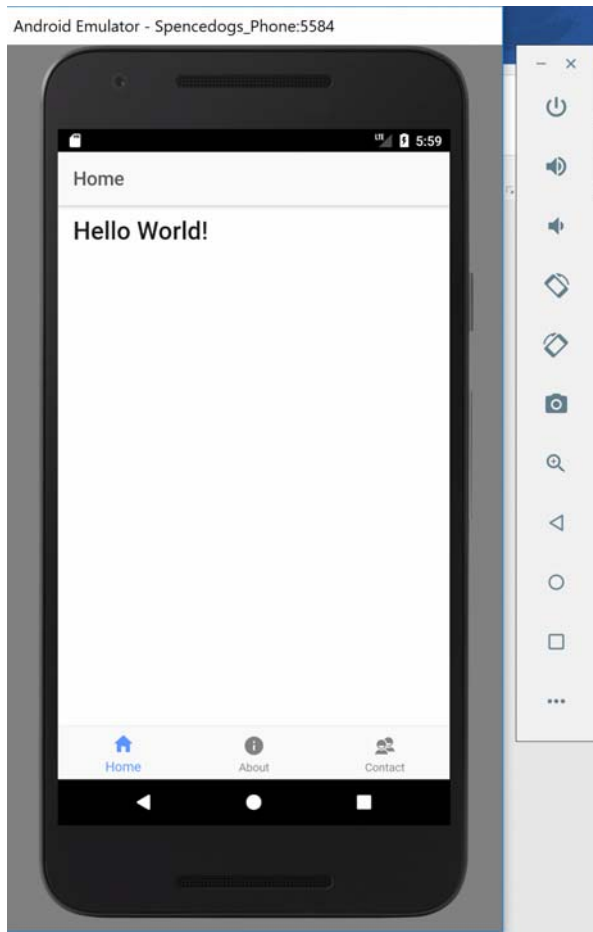
-c = console logs. Anything you print in your app will show up on the screen

-l = enables live reload. Anytime you save changes to your code, it will reload your app with the changes

-s = will print stack traces for your errors to your terminal

```
npm
C:\Users\Spencer\Desktop\2310\IonicProjects\walkthroughApp>ionic cordova emulate android -c -l -s
> cordova platform add android --save
Using cordova-fetch for cordova-android@6.3.0
```

This command will take some time. After it executes, your emulator should start and our Hello World homepage should be visible:



And that's it! That's all it takes to get a sample app up and running on a virtual device. To start coding with Ionic, you need to start learning Angular2 and NodeJS.

For NodeJS, here is a link to a tutorial: <https://www.tutorialspoint.com/nodejs/>. NodeJS is used as the backend for your apps. You can create a database or a RESTful API very quickly after mastering the basics of NodeJS

For Angular 2, here is a link to a tutorial: <https://www.tutorialspoint.com/angular2/>. Angular 2 is used for the frontend for your apps. It is used to respond to events in your HTML such as button clicks and page changes.

(Prepared by Spencer Gray)