Quiz Time

Table of Contents

[Overview: 1](#_Toc536729131)

[Ionic 5 Overview: 1](#_Toc536729132)

[Installing Ionic and Environment 2](#_Toc536729133)

[Node.js 2](#_Toc536729134)

[Ionic and Cordova 2](#_Toc536729135)

[Code Editor 2](#_Toc536729136)

[Starting Your Ionic App 3](#_Toc536729137)

[Ionic Project Structure 4](#_Toc536729138)

[Working with “Src” Folder 5](#_Toc536729139)

[Typescript 5](#_Toc536729140)

[Preparing iOS & Android Environment 6](#_Toc536729141)

[Build for iOS 6](#_Toc536729142)

[Build for Android 7](#_Toc536729143)

[Installation Steps in Short for QuizTime App 8](#_Toc536729144)

# Overview:

Create a beautiful app in Android, IOS and Windows phone with single code template providing a great native user experience in all the mobile OS using Ionic 5, a great powerful UI framework. Read all the contents to get an overview and proceed to start with app or go to section “Installation Steps in Short for QuizTime App” to start up with your quiz app.

# Ionic 5 Overview:

Ionic is a powerful HTML5 SDK that helps you build native-feeling mobile apps using web technologies like HTML, CSS, and Typescript (Angular 5).

The app on a mobile device is running inside a webview, but of course **you don’t see a navigation bar** inside that app later! It looks and feels like a regular app, and can be installed from the app store of course.

Through [Cordova](https://cordova.apache.org/) our “webview app” get’s packaged into a native iOS or Android project, and from that point on you can deploy it like you could any other native app.With Ionic your app is built from one single code base, and the apps are often referred to as Cross Platform Apps or also Hybrid Apps (although the latter has a bitter taste for some people).

Ionic is focused mainly on the look and feel, and UI interaction of your app. Ionic is that missing piece when creating native apps with web standards. Just drop in some CSS and Typescript, and you’ll quickly get up to speed developing native applications with HTML5. And when you are ready to push to the app stores, compile your app with [Phone Gap](http://phonegap.com/) or [Cordova](http://cordova.apache.org/).

To get more details on the usage, read <https://ionicframework.com/docs>.

# Installing Ionic and Environment

To work with Ionic we need to install a few packages on our machine. Make sure your environment is set up before beginning.

## Node.js

Whether you are using Windows, Linux or Mac the first step is to make sure you have [Node.js](https://nodejs.org/en/) installed on your machine. If you haven’t, simply [go to the official Node.js page](https://nodejs.org/en/), pick the right package for your machine, download and finally install it.

## Ionic and Cordova

Ionic and cordova can be installed as a Node.js package through the **Node Package Manager** (npm) directly from our command line. On a Mac, you can open the Terminal, on other machines use the according Shell and run this command:

npm install –g ionic cordova

Here we are installing ionic and cordova package together. Cordova package helps to build our apps into native projects for iOS or Android.

“**-g**” option means to install the package globally on the machine and not just in your current folder.

Make sure if ionic is installed perfectly without any issues and it works on the command line prompt (CLI) by checking the below command.

ionic info

This command should retrieve the ionic versions and other package details. In case of error, make sure to uninstall the entire package and re install again.

## Code Editor

You can use code editors like Sublime Text Editor, Atom, Visual Studio Code, Webstorm or whatever you prefer. My preference is to use webstorm or Visual Studio Code which supports HTML, CSS and Typescript and it have lot of plugin support for compilation.

# Starting Your Ionic App

We are now ready to start the first project. Ionic provides many starter templates for easy creation of projects. To view the existing list of starter templates, execute the below command from your command line interface (CLI) and choose your template for app creation.

ionic start –list

To create projects simply navigate to a folder where you want to **start your app** and run the below command from your command line:

ionic start <Your App Name> blank

Eg: ionic start MyFirstApp blank

This might take a bit of time depending on your network connection to complete normally as it will install quite a few packages required for your project along with the “blank” ionic app.Once this bootstrapping process is done you can navigate into your folder (cd MyFirstApp) and directly **serve your app** using below command from CLI.

ionic serve

This will **compile** the project and run it on a local browser.Alternatively you can install ionic lab and serve your app from CLI to ionic lab with below command.

ionic lab

This will **compile** the project and run it on a local server inside ionic lab.

In both the cases, you don’t have to take care of anything here, just sit and wait. And finally your browser should open with a view of your project. That is a very good sign, because you have just created and started your first Ionic app.

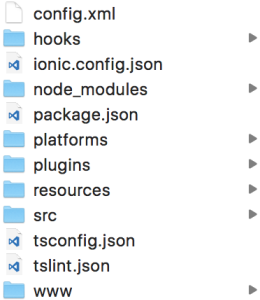
The screen displayed is a **preview function** which you will use 95% of the time to develop your apps – and its unbelievable fast. If you are a former native developer, this will feel so much faster, and if you are coming from web development you feel right at home.

You can pick which **platform** you would like to see the preview for – different platforms will automatically have a **different styling.**

# Ionic Project Structure

The files inside your folder might look quite very big, but most of the time you will be working only in the **src** folder and can forget about the rest.

Your app folder will look very likely like the one in the image below.



The **config.xml** is used once the native project is built from this Ionic project. If you need special permissions in the native app or change other stuff, it will be set inside this file.

The **hooks** directory contains actions that will be automatically executed at given lifecycle events of the app. Most of the time you don’t need to touch any file in there.

The **ionic.config.json** contains some basic information about your project and is used if you upload your app to the [Ionic.io platform](https://apps.ionic.io/).

The **node\_modules** folder is automatically generated once you install the npm dependencies with “npm install” (Ionic already did this for you in the beginning). This command will scan the **package.json** for all the packages that need to be installed and is a classic Node.js file.

The **platforms** folder contains your native projects (Android, iOS, Windows). You need to add them first and they will be generated to this folder.

The **plugins** folder contains all the Cordova plugins that you installed. As you might know, these are the wrapper around native functions.

The **resources** folder is an Ionic folder that can contain app icon and splash screen.

The **src** folder is the **most important folder**, and 99% of your work will happen in that folder. It’s the folder that contains your actual Angular code.

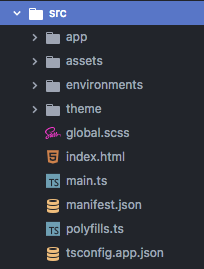
The next two files (**tsconfig.json**, **tslint.json**) are related to Typescript and how it gets compiled. We don’t really have to care for them, just keep them like they are and you are fine.

The last folder is the **www** folder which is auto generated on production build of your app and it contains your current build when you run the ionic preview inside your browser. Don’t change code here. This is very important, especially if you come from Ionic 1.

With the knowledge about all of these folders we can dive into what is actually our app and change a few things!

## Working with “Src” Folder

As said before, our entire working are inside the “src” folder of your project, therefore we inspect what we currently have there. The folder structure inside src looks like in the image below.



The **app** folder is more or less the **entry point** of your app. Everything inside that folder is used when your app is bootstrapped. This folder works in combination with the **index.html**, which also imports some stuff and has this part somewhere inside the body of your HTML:

<!-- Ionic's root component and where the app will load -->

<ion-app></ion-app>

This is the place where your app will be loaded into.

You rarely have to touch that file, but it’s good to know how everything works.

Our App is made up of different **pages**, and after starting our app we already have on page inside that folder called **home**. Every page will have its own folder with a HTML, CSS and TS file. The CSS file defines styling for your view, the HTML page represents your view and the TypeScript file contains the class that is associated with your view. The one thing new here might be the file type TS which stands for TypeScript.

## Typescript

TypeScript is a superset of JavaScript which most important adds Types to our variables like **void**, **string** or **Boolean**. We will always code in TypeScript and the build system will **transpile** this code into plain JavaScript which every browser can understand and all it happens automatically.

# Preparing iOS & Android Environment

If you want to build your app into a native package, you need few more things to be done apart from app coding.

## Build for iOS

**Note: If you are not on a Mac you can’t build your app for iOS! This is simply not possibly yet due to how iOS works.**

Open the **App Store** on your Mac and search for “**Xcode**“. This is the native development environment for iOS developers and we can use it to try out our app on iOS devices.After installing Xcode (which is free) you can run directly inside your Ionic project the command to build everything into a native project:

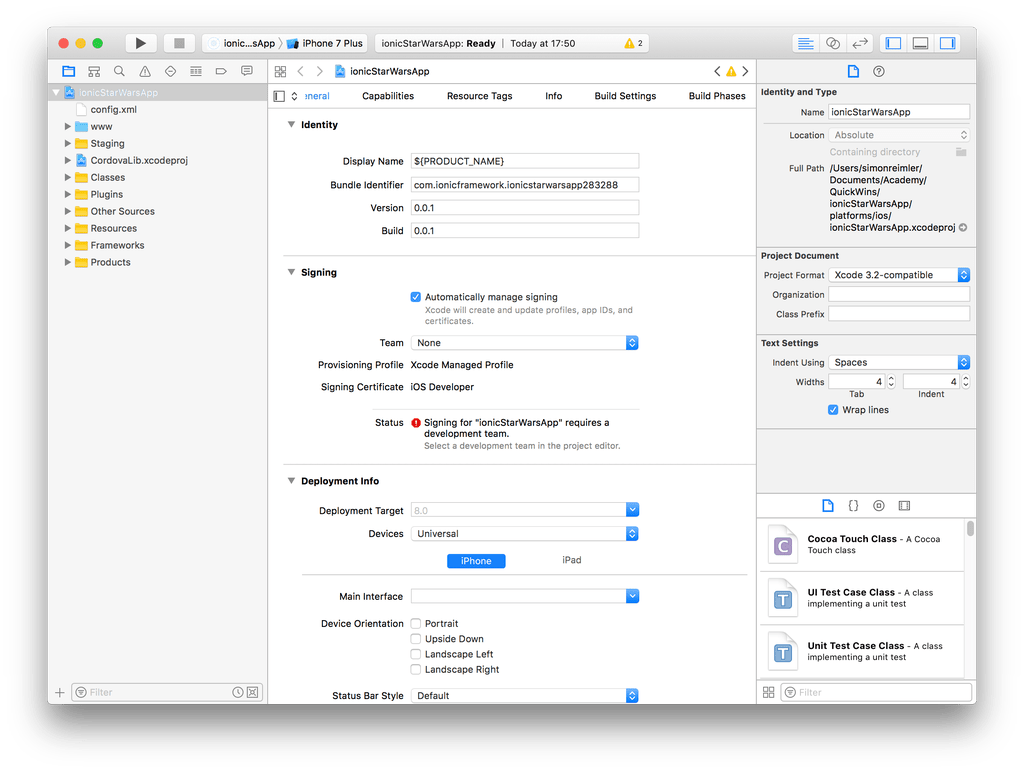
ionic cordova platform add ios

The above command needs to be **called only once for a project**, and it sets up the iOS platform for the project. This might take a few moments and once done, you will find the iOS platform added **inside the platforms folder.**

Run the below command to build your code to an iOS application file (.ipa). Make sure to run the below command every time you make changes in your application file.

ionic cordova build ios

To open the project in Xcode double click the **MyFirstApp.xcodeproj**, which will bring up Xcode. If you click on the project on the left side the general view opens which should look like this:

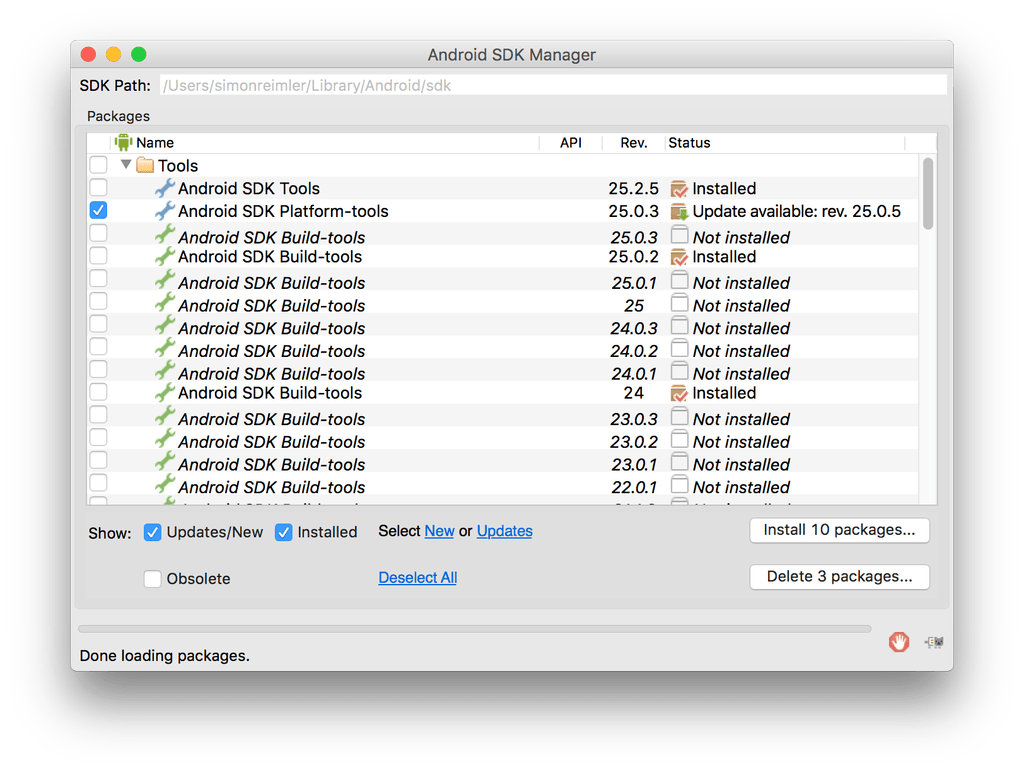


At the top left next to the run button you now see the name of your project and the currently selected simulator. You can now **hit the run button** to deploy your app to an iOS simulator and get the feeling of a native app. If you want to **deploy your app to your iPhone**, you can connect it through USB and it will appear at the top of that list above the simulators.

## Build for Android

For Android, [download the official environment Android Studio](https://developer.android.com/studio/index.html) which will help you to install the Android SDK and AVD managers.

After installing Android Studio open it and inside the Wizard click **Configure** and pick **SDK Manager**. This will bring you to a view where you can install an Android SDK – simply pick the newest one and install what you need!

  
Now we can add the Android platform to our project, so go to your command line and run:

ionic cordova platform add android

ionic cordova build android

|  |  |
| --- | --- |
|  |  |

This results in a new Android project inside the **platforms** folder. You can now either go ahead and import this project into Android Studio or directly install it to your Android device if it’s connected with USB.

Ionic already told us the **path to the generated APK** after the build command.

If you have followed the course so far, this should be:

**/platforms/android/build/outputs/apk/android-debug.apk**

You can now use the [Android Debug Bridge](https://developer.android.com/studio/command-line/adb.html) (ADB) to install it by running:

Adb install platform/android/build/outputs/apk/android-debug.apk

|  |  |
| --- | --- |
|  |  |

This will install the App on your device.Also for testing purpose, you can directly install your code android device connected with USB using the below command.

ionic cordova run android

# Installation Steps in Short for QuizTime App

1. Download and install NodeJs from <https://nodejs.org/en/>
2. Open command prompt and install Cordova and Ionic globally

npm install –g cordova

npm install –g [ionic](mailto:ionic@3.9.2)

1. Unzip the downloaded app zip file
2. Create a new ionic project from the command prompt

ionic start <YourAppName> blank

1. Copy the **folders** inside **quizTime/src** to your projects **src** folder
2. Include the npm chart library by opening the current folder in the terminal/ command prompt and execute the below lines of code

npm install --save chart.js && npm install --save-dev @types/chart.js

1. To execute the app, open command prompt and move to project directory. Run “**ionic serve**” to execute the project in browser.