IONIC

**INTRODUCTION**

**Ionic** is open source framework used for developing mobile applications. It provides tools and services for building Mobile UI with native look and feel. Ionic framework needs native wrapper to be able to run on mobile devices. This is an introductory tutorial, which covers the basics of the Ionic Open Source Framework and explains how to deal with its various components and sub-components.

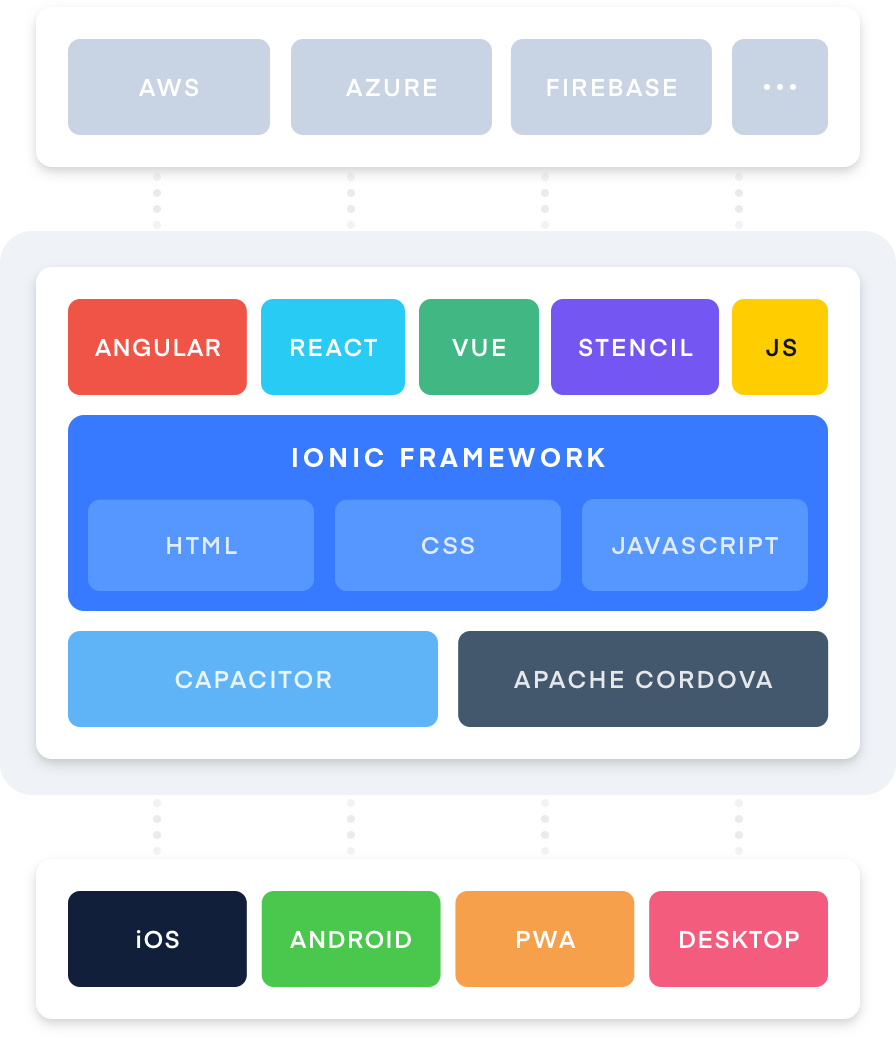
Ionic makes it easy to build high-performance mobile and Progressive Web Apps (or PWAs) that look and feel beautiful on any platform or device.

Ionic’s open source Framework and developer-friendly tools and services power apps for some of the world’s best-known brands - from highly successful consumer apps like Sworkit, Untappd and Dow Jones MarketWatch, to mission-critical apps supporting Nationwide, Amtrak, and NASA.

**A complete app development kit.**

The open source [Ionic Framework](https://ionicframework.com/framework) features a rich library of front-end building blocks and UI components that make it easy to design beautiful, high-performance mobile and Progressive Web Apps (or PWAs) using web technologies like HTML, CSS, and JavaScript.

Our universal web components pair with any JavaScript framework, including Angular, React, Vue, or no framework at all (just add a script tag!). Ionic apps are backend agnostic, with connections to AWS, Azure, and Firebase.



**IONIC OVERVIEW**

**Ionic** is a front-end HTML framework built on top of **AngularJS** and **Cordova**. As per their official document, the definition of this Ionic Open Source Framework is as follows −

Ionic is an **HTML5 Mobile App Development Framework** targeted at building hybrid mobile apps. Think of Ionic as the front-end UI framework that handles all the look and feel and UI interactions your app needs to be compelling. Kind of like "Bootstrap for Native", but with the support for a broad range of common native mobile components, slick animations and a beautiful design.

**IONIC FRAMEWORK FEATURES**

* **AngularJS** − Ionic is using AngularJS MVC architecture for building rich single page applications optimized for mobile devices.
* **CSS components** − With the native look and feel, these components offer almost all elements that a mobile application need. The components’ default styling can be easily overridden to accommodate your own designs.
* **JavaScript components** − These components are extending CSS components with JavaScript functionalities to cover all mobile elements that cannot be done only with HTML and CSS.
* **Cordova Plugins** − Apache Cordova plugins offer API needed for using native device functions with JavaScript code.
* **Ionic CLI** − This is NodeJS utility powered with commands for starting, building, running and emulating Ionic applications.
* **Ionic View** − Very useful platform for uploading, sharing and testing your application on native devices.
* **Licence** − Ionic is released under MIT license.

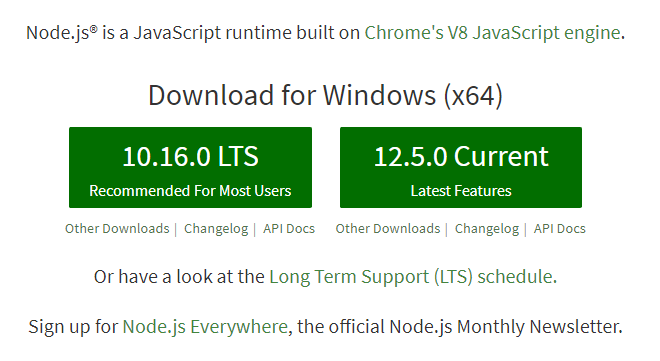
**IONIC FRAMEWORK ADVANTAGES**

* Ionic is used for Hybrid App Development. This means that you can package your applications for IOS, Android, Windows Phone and Firefox OS, which can save you a lot of working time.
* Starting your app is very easy since Ionic provides useful pre-generated app setup with simple layouts.
* The apps are built in a very clean and modular way, so it is very maintainable and easy to update.
* Ionic Developers Team have a very good relationship with the Google Developers Team and they are working together to improve the framework. The updates are coming out regularly and Ionic support group is always willing to help when needed.

**IONIC FRAMEWORK LIMITATIONS**

* Testing can be tricky since the browser does not always give you the right information about the phone environment. There are so many different devices as well as platforms and you usually need to cover most of them.
* It can be hard to combine different native functionalities. There will be many instances where you would run into plugin compatibility issues, which leads to build errors that are hard to debug.
* Hybrid apps tend to be slower than the native ones. However, since the mobile technologies are improving fast this will not be an issue in the future.

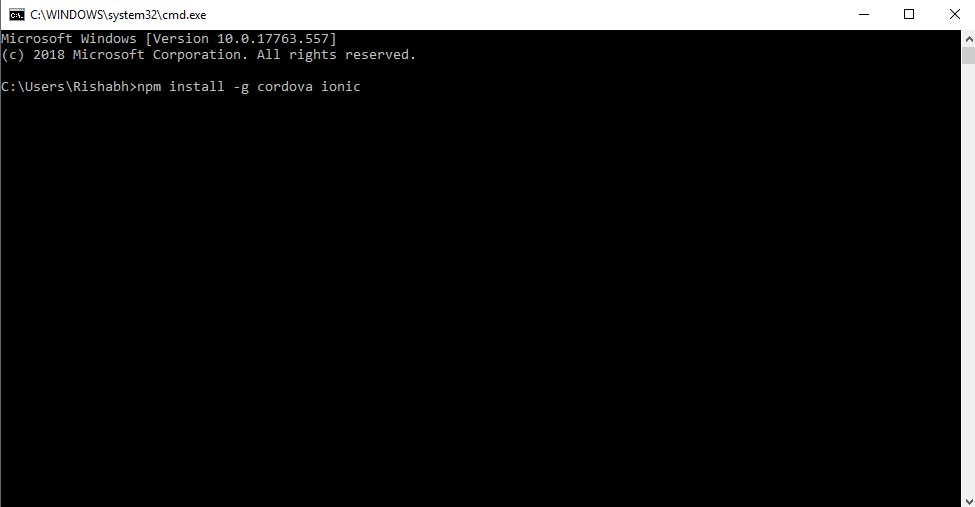
**IONIC- Environment Setup**

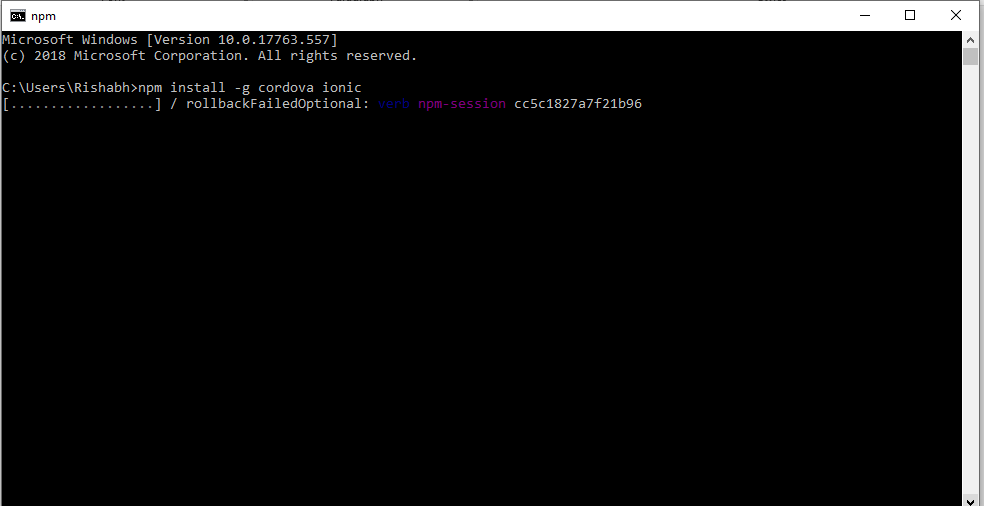
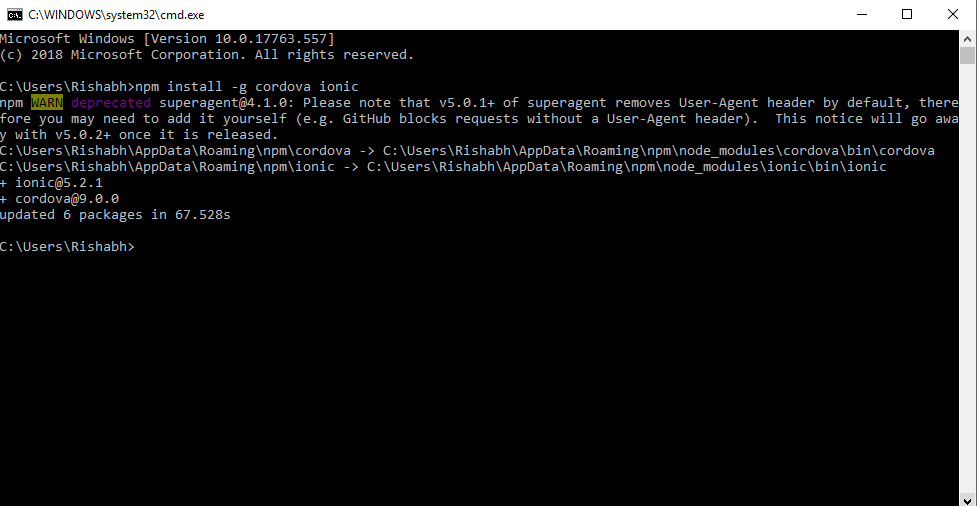
1. **Installing Node.js:-**

* Install any version LTS for stable and Current will be the latest version of node.js
* After installing Node.js ,you will easily use the command prompt to download cordova ionic.
* This is the base platform needed to create Mobile Apps using Ionic.
* It will install npm on your device.

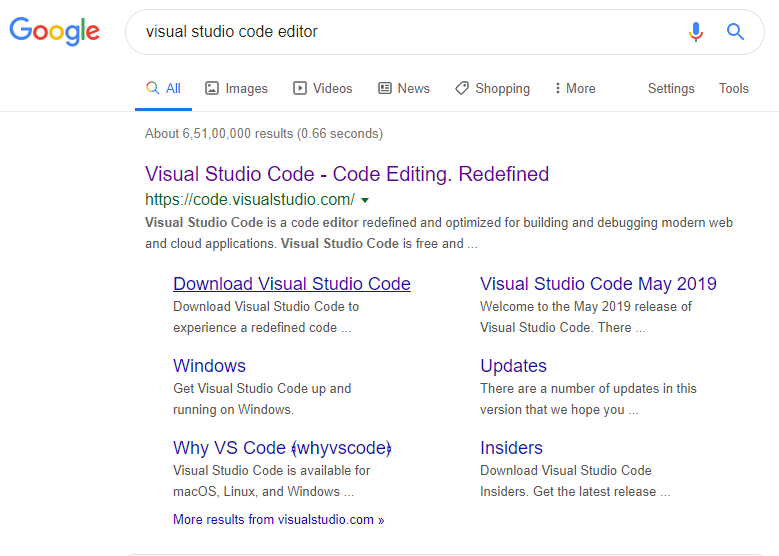
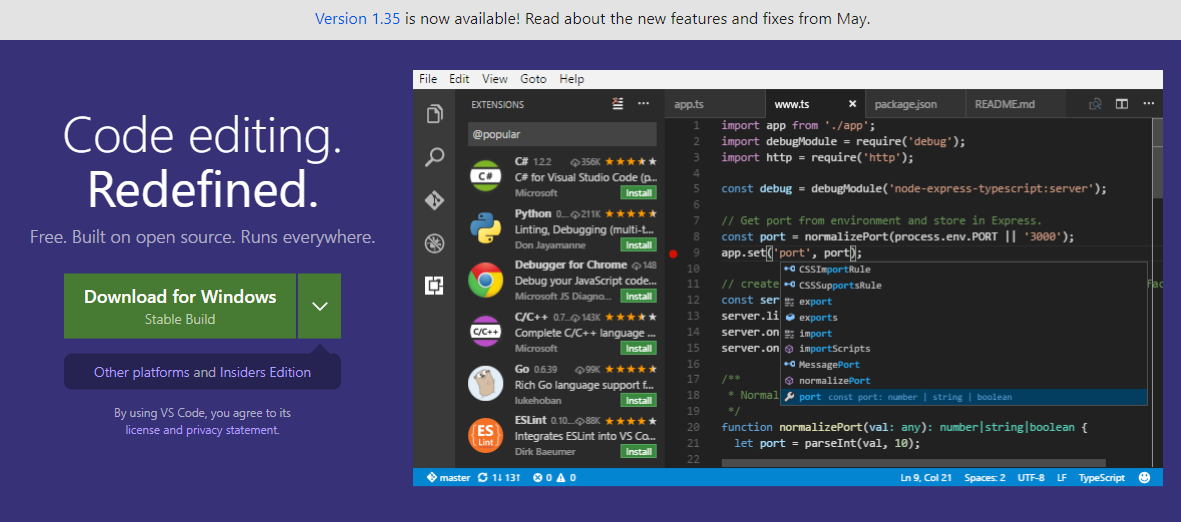
1. **Installing Cordova Ionic: -**

* After installing Node.js it will enable npm on your device.
* Open Command prompt and type

npm install -g cordova ionic

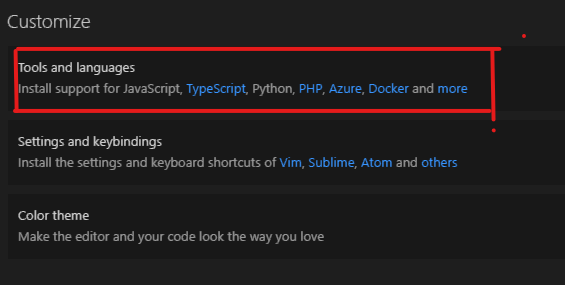
* After type press enter to execute the command.
* It will take 5-10min to install Cordova Ionic on your device
* It will also take a good internet connection to install
* After installing cordova ionic you will need a good programming editor to work on ionic.
* I recommended you use Visual Studio Code Editor for work on Ionic.

1. **Installing Visual Studio Code Editor: -**

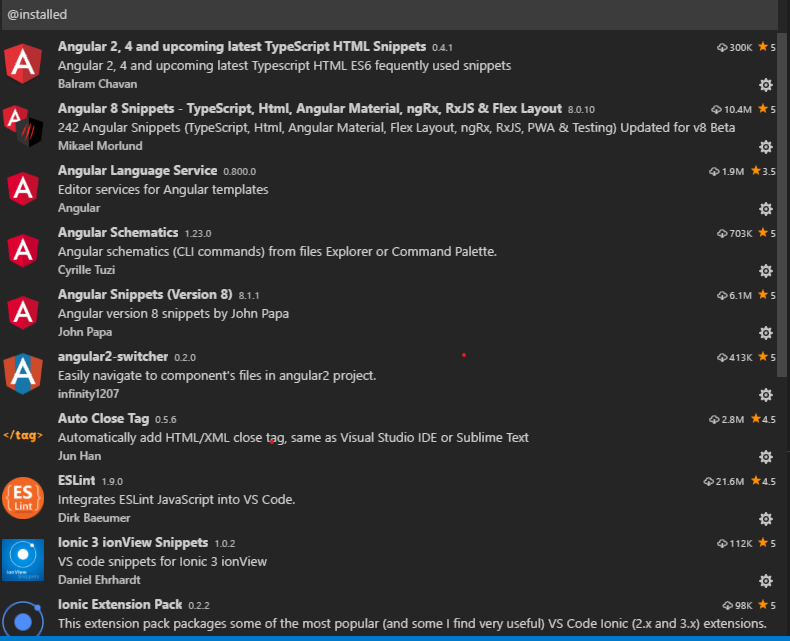
* Search for Visual Studio Code Editor on Google Search
* After open the site you will see the Download button for downloading Visual Studio Code Editor.

1. **Installing Visual Studio Code Editor: -**

* After installing the Editor on the device, installed the external extension for Angular, CSS, HTML, Autotag etc



* After open the editor Click on Tools and Languages for installing different extension for cordova ionic.

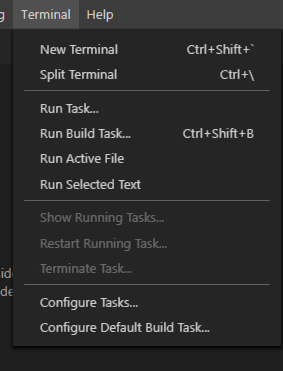


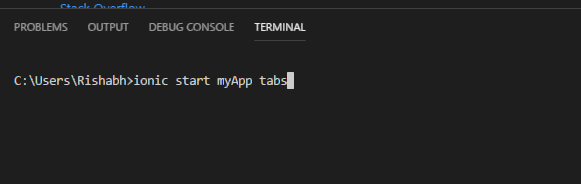
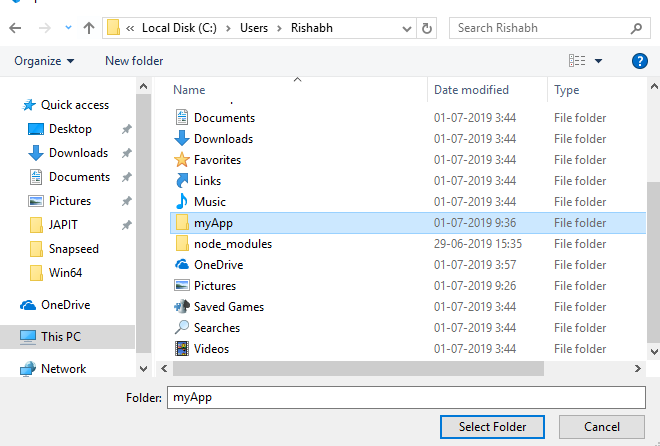
**How to Create Default Apps on Ionic Cordova**

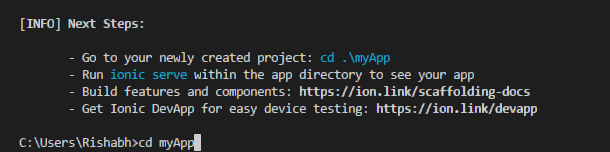
While creating apps in Ionic, you can choose from the following three options to start with −

* Tabs App
* Blank App
* Side menu app

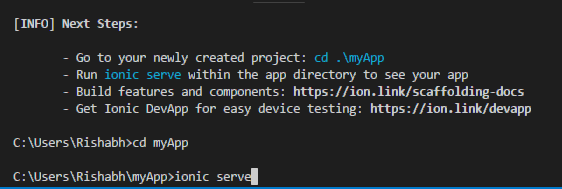
**1.Tabs App**

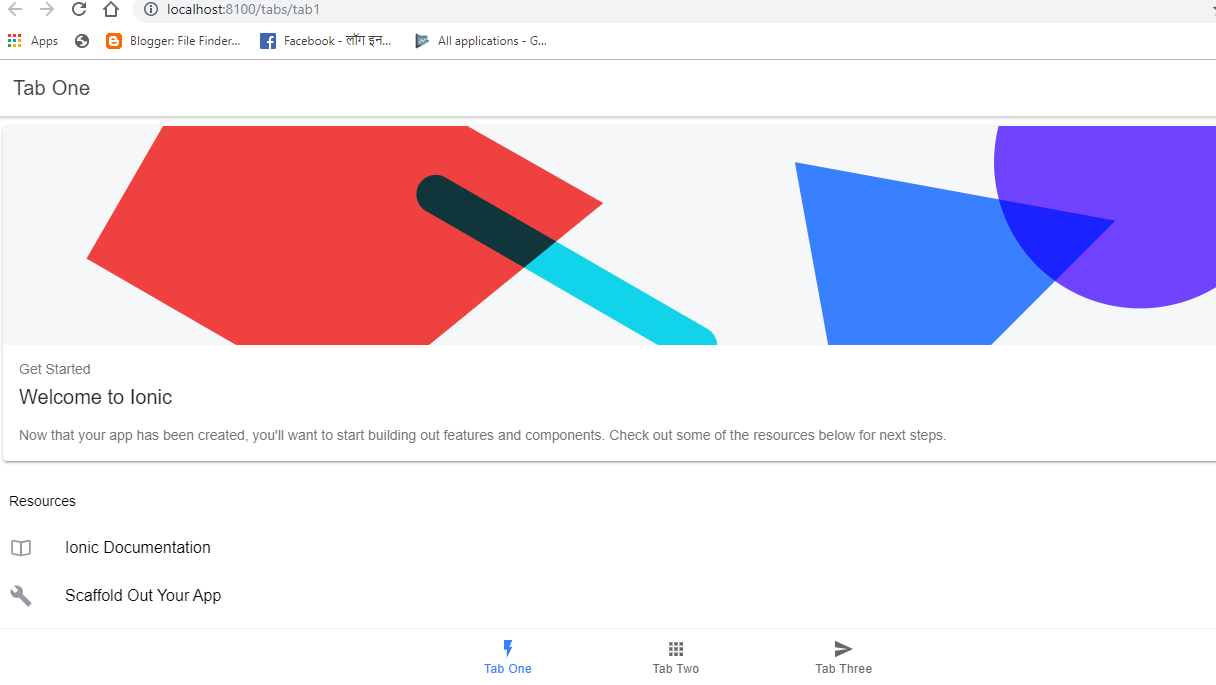
If you want to use the Ionic tabs template, the app will be built with the tab menu, header and a couple of useful screens and functionalities. This is the default Ionic template. Open your command window and choose where you want to create your app.

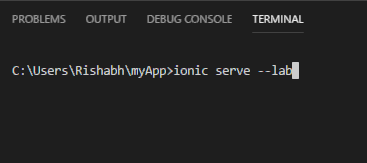
* **Open Editor and type on a terminal**
* **Type on terminal**
* **After type on terminal press enter for execute after executing it will created default tab app.**
* **After completion of the project, you want to browser the folder on the help of the editor and open it.**
* **After browser the folder then select the folder.**
* **Then change the directory to myApp just type: cd myApp on the terminal**

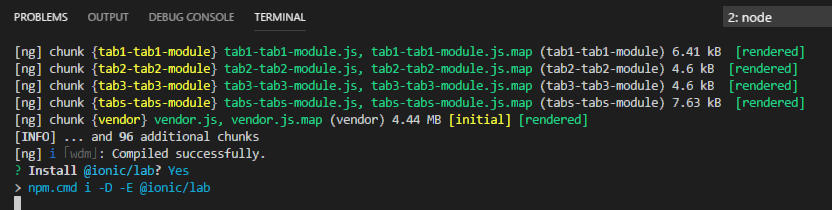


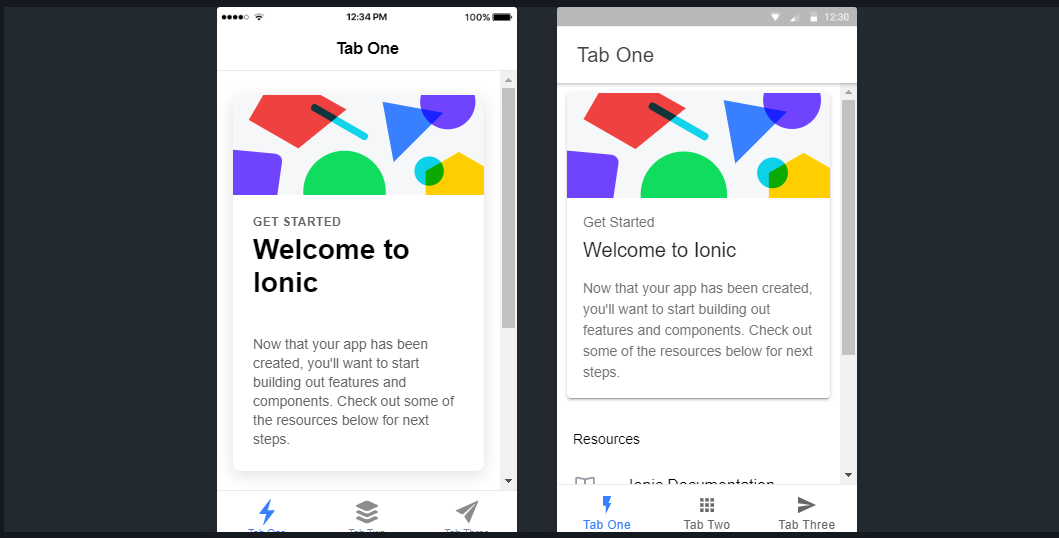
* **Then type: ionic serve command to execute the default tabs app.**



* **After ionic serve command finished it will open a lost host to your default browser.**
* **You will also use Ionic Lab for preview the project.**
* **Just type: ionic serve –lab on terminal**



* **After execution it will take some time to installed the ionic lab on your device.**
* **Ionic Lab provide a platform where you will preview your app on different platform like Android, iOS, Windows.**



**DIFFERENT FILES ON IONIC**

It will create five types of file:

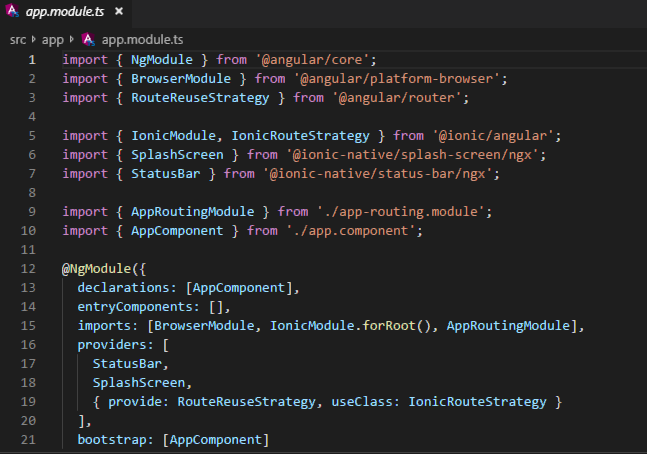
* module.ts
* page.html
* page.scss
* page.spec.ts
* page.ts

1. **module.ts :**

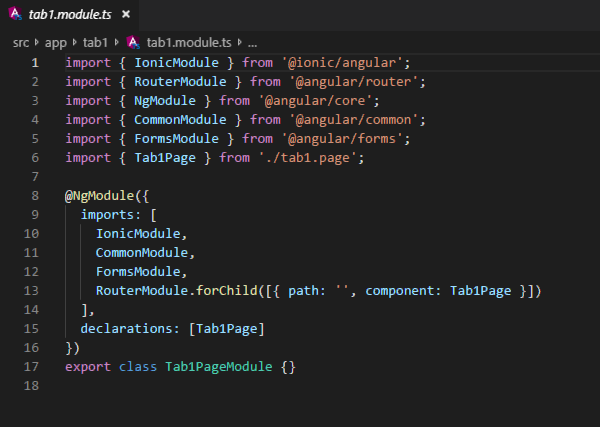
Angular apps are modular and Angular has its own modularity system called NgModules. NgModules are containers for a cohesive block of code dedicated to an application domain, a workflow, or a closely related set of capabilities. They can contain components, service providers, and other code files whose scope is defined by the containing NgModule. They can import functionality that is exported from other NgModules, and export selected functionality for use by other NgModules.

Every Angular app has at least one NgModule class, [the root module](https://angular.io/guide/bootstrapping), which is conventionally named AppModule and resides in a file named app.module.ts. You launch your app by bootstrapping the root NgModule.

While a small application might have only one NgModule, most apps have many more feature modules. The rootNgModule for an app is so named because it can include child NgModules in a hierarchy of any depth.



**(App.module.ts)**

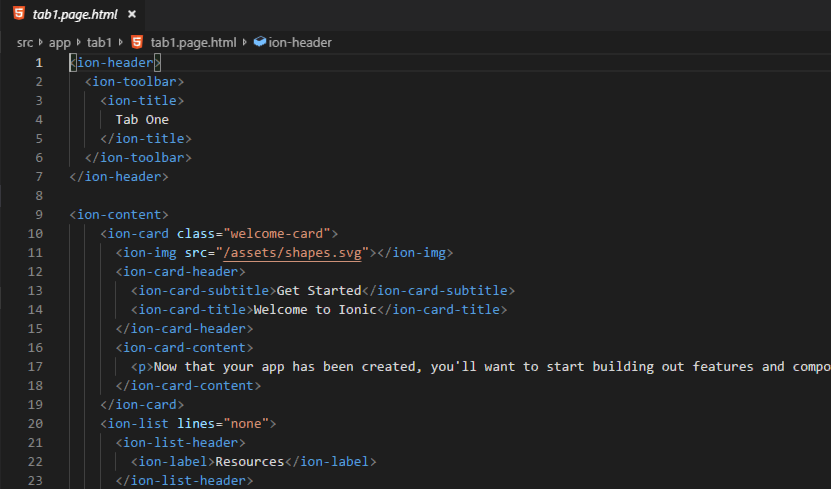


**(Tab1.module.ts)**

1. **page.html :**

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

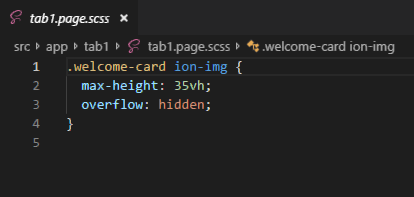
**Hyper Text:** HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

1. **page.scss:**

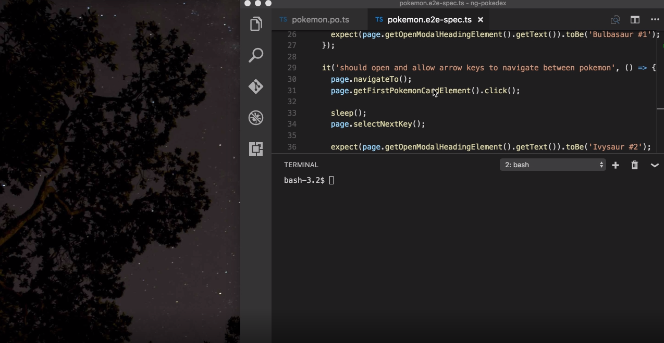
Ionic components are built with [CSS Variables](https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables) for easy customization of an application. CSS variables allow a value to be stored in one place, then referenced in multiple other places. They also make it possible to change CSS dynamically at runtime (which previously required a CSS preprocessor). CSS variables make it easier than ever to override Ionic components to match a brand or theme.

All of the Ionic components are themed using [CSS properties (variables)](https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables). CSS variables add dynamic values to an otherwise static language. This is something that has traditionally required a CSS preprocessor like Sass. The look of an application can easily be changed by changing the value of any of the [Ionic Variables](https://ionicframework.com/docs/theming/css-variables#ionic-variables).



1. **page.spec.ts:**

End to end testing (E2E) or also known as integration testing is a great way to make sure at a high-level overview that our applications function correctly. Commonly we use E2E tests to help ensure our components are working correctly together to create an entirely functioning feature cohesively. E2E tests are not Unit tests. Unit tests should test a small isolated piece of code while the E2E test is a high-level test of a feature or several interactions. E2E tests are also ideal for making sure critical line of business features such as checkouts and sign-ups are well tested. I won’t dig into all the details of Unit tests vs. E2E tests but focus on how integration and E2E tests work in Angular.



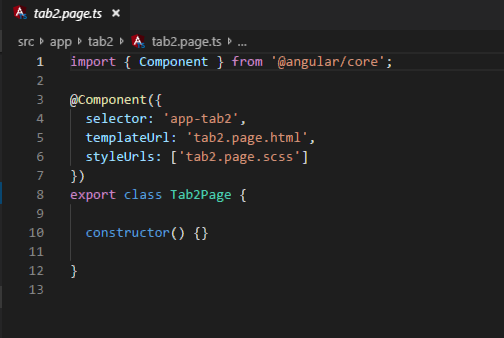
1. **page.ts:**

Components are the basic constructs of Angular.

A component is an independent construct which encapsulates the controller code, the view and styles,and has inputs and outputs.

An Ionic 4/Angular app is a bunch of component (an app component + page components) and providers organized together using Angular modules.

An Ionic 4/Angular app contains a root or app component and optionally 1 or more-page components.

Here is an example of a main Ionic app component which lives in the src/app/app.component.ts file: