



ionic

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

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.

Audience

This tutorial is created for JavaScript developers that are new to mobile development. It provides simple, easy to understand explanations with useful working examples. We will go through most of the Ionic Framework so you can also use this as a reference for your projects.

This tutorial is intended to make you comfortable in getting started with the Ionic Open Source Framework and its various functions.

Prerequisites

Since Ionic is built on top of AngularJS and Apache Cordova, you will need to have basic knowledge about these technologies. You also need to be familiar with HTML, CSS and JavaScript, if you want to understand all the information provided.

Copyright and Disclaimer

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright and Disclaimer	i
Table of Contents	ii
IONIC – BASICS.....	1
1. Ionic – Overview	2
Ionic Framework Features.....	2
Ionic Framework Advantages	2
Ionic Framework Limitations	3
2. Ionic – Environment Setup	4
Installing Cordova and Ionic	4
Creating Apps	4
Tabs App	5
Blank App.....	7
Side Menu App	9
Test App in Browser	11
Project Folder Structure	12
IONIC – CSS COMPONENTS	14
3. Ionic – Colors	15
Ionic Color Classes	15
Ionic Color Usage.....	15
Customizing Colors with CSS	16
Customizing Colors using SASS.....	17
4. Ionic – Content.....	19
5. Ionic – Header	20
Adding Header	20
Header Colors.....	21
Header Elements	22
Sub Header	23
6. Ionic – Footer	26
Adding Footer	26
Footer Colors	27
Footer Elements	28
7. Ionic – Buttons.....	30
Block Buttons.....	30
Button Size.....	31
Button Colors.....	31
Button Outline	33
Adding Icons	34
Button Bar	35

8. Ionic – Lists	36
Creating Ionic List	36
Inset List.....	37
Item Dividers	37
Adding Icons	39
Adding Avatars and Thumbnails.....	40
9. Ionic – Cards	42
Adding Cards.....	42
Card Header and Footer	43
Complete Card.....	44
10. Ionic – Forms.....	46
Using Input Form	46
Ionic Labels	47
Stacked Label.....	48
Floating Label	49
Inset Inputs	50
11. Ionic – Toggle	52
Using Toggle	52
Multiple Toggles	53
Styling Toggle.....	55
12. Ionic – Checkbox	59
Adding Checkbox.....	59
Multiple Checkboxes	60
Styling Checkbox.....	61
13. Ionic – Radio Button.....	64
Adding Radio Buttons.....	64
Multiple Radio Button Groups.....	66
14. Ionic – Range	68
Using Range	68
Adding Icons	69
Styling Range	70
15. Ionic – Select	72
Using Select	72
Styling Select	73
16. Ionic – Tabs	78
Simple Tabs.....	78
Adding Icons	79
Striped Tabs.....	82
17. Ionic – Grid.....	83
Simple Grid	83
Column Sizes.....	84
Horizontal and Vertical Positioning	86
Responsive Grid	88

18. Ionic – Icons	91
How to use Icons?	91
19. Ionic – Padding	93
IONIC – JAVASCRIPT COMPONENTS	95
20. Ionic – JavaScript Action Sheet	96
Using Action Sheet	96
21. Ionic – JavaScript Backdrop	100
Using Backdrop.....	100
22. Ionic – JavaScript Content	102
23. Ionic – JavaScript Forms	103
Using ion-checkbox.....	103
Using ion-radio	106
Using ion-toggle.....	107
24. Ionic – JavaScript Events	110
Using Events	111
25. Ionic – JavaScript Header	112
Using JavaScript Header	112
Styling Header	112
Adding Elements.....	113
Adding Sub Header	114
26. Ionic – JavaScript Footer	116
Using Footer	116
Adding Elements.....	117
Adding Sub Footer	117
27. Ionic – JavaScript Keyboard	119
Using Keyboard.....	119
28. Ionic – JavaScript List	123
Using List	123
Delete Button	124
Reorder Button.....	125
Option Button.....	127
Other Functions	129
29. Ionic – JavaScript Loading	130
Using Loading	130
Loading Config	132
30. Ionic – JavaScript Modal	134
Using Modal.....	134
31. Ionic – JavaScript Navigation	140
Using Navigation.....	140

Creating Navigation Menu.....	141
Adding Navigation Elements	143
Other Navigation Attributes	144
Caching	144
Controlling the Navigation Bar	145
Tracking History	145
32. Ionic – JavaScript Popover	148
Using Popover	148
33. Ionic – JavaScript Popup	153
Using Show Popup.....	153
Using Confirm Popup.....	155
Using Alert Popup.....	157
Using Prompt Popup.....	159
34. Ionic – JavaScript Scroll	162
Using Scroll	162
Infinite Scroll.....	166
Scroll Delegate.....	167
35. Ionic – JavaScript Side Menu.....	172
Using Side Menu.....	172
Side Menu Delegate	174
36. Ionic – JavaScript Slide Box	176
Using Slide Box	176
Slide Box Delegate.....	179
37. Ionic – JavaScript Tabs	181
Using Tabs	181
IONIC – ADVANCED CONCEPTS	184
38. Ionic – Cordova Integration.....	185
Installing ngCordova.....	185
39. Ionic – Cordova AdMob.....	186
Using AdMob	186
40. Ionic – Cordova Camera	190
Using Camera.....	190
41. Ionic – Cordova Facebook	197
Installing Facebook Plugin	197
Angular Service	198
42. Ionic – Cordova InAppBrowser	202
Using Browser.....	202
43. Ionic – Cordova Native Audio	205
Using Native Audio	205

44. Ionic – Cordova Geolocation	207
Using Geolocation	207
45. Ionic – Cordova Media	209
Using Media.....	209
46. Ionic – Cordova Icon & Splash Screen.....	212
Adding Splash Screen and Icon.....	212

Ionic – Basics

1. 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

Following are the most important features of Ionic:

- **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 needs. 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

Following are some of the most commonly known 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

Following are some of the most important 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.

In the next chapter, we will understand the environment setup of the Ionic Open Source Framework.

2. Ionic – Environment Setup

This chapter will show you how to start with Ionic Framework. The following table contains the list of components needed to start with Ionic.

S No.	Software	Description
1	NodeJS	This is the base platform needed to create Mobile Apps using Ionic. You can find detail on the NodeJS installation in our NodeJS Environment Setup . Make sure you also install npm while installing NodeJS.
2	Android SDK	If you are going to work on a Windows platform and are developing your apps for the Android platform, then you should have Android SDK setup on your machine. The following link has detailed information on the Android Environment Setup .
3	XCode	If you are going to work on the Mac platform and are developing your apps for the iOS platform, then you should have XCode setup on your machine. The following link has detailed information on the iOS Environment Setup .
4	Cordova and Ionic	These are the main SDKs which is needed to start working with Ionic. This chapter explains how to setup Ionic in simple step assuming you already have the required setup as explained in the table above.

Installing Cordova and Ionic

We will use the Windows command prompt for this tutorial. The same steps can be applied to the OSX terminal. Open your command window to install Cordova and Ionic:

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

Creating Apps

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

- Tabs App
- Blank App
- Side menu app

In your command window, open the folder where you want to create the app and try one of the options mentioned below.

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.

```
C:\Users\Username> cd Desktop
```

This command will change the working directory. The app will be created on the Desktop.

```
C:\Users\Username\Desktop> ionic start myApp tabs
```

Ionic **Start** command will create a folder named **myApp** and setup Ionic files and folders.

```
C:\Users\Username\Desktop> cd myApp
```

Now, we want to access the **myApp** folder that we just created. This is our root folder.

Let us now add the Cordova project for the Android Platform and install the basic Cordova plugins as well. The following code allows us to run the app on the Android emulator or a device.

```
C:\Users\Username\Desktop\myApp> ionic platform add android
```

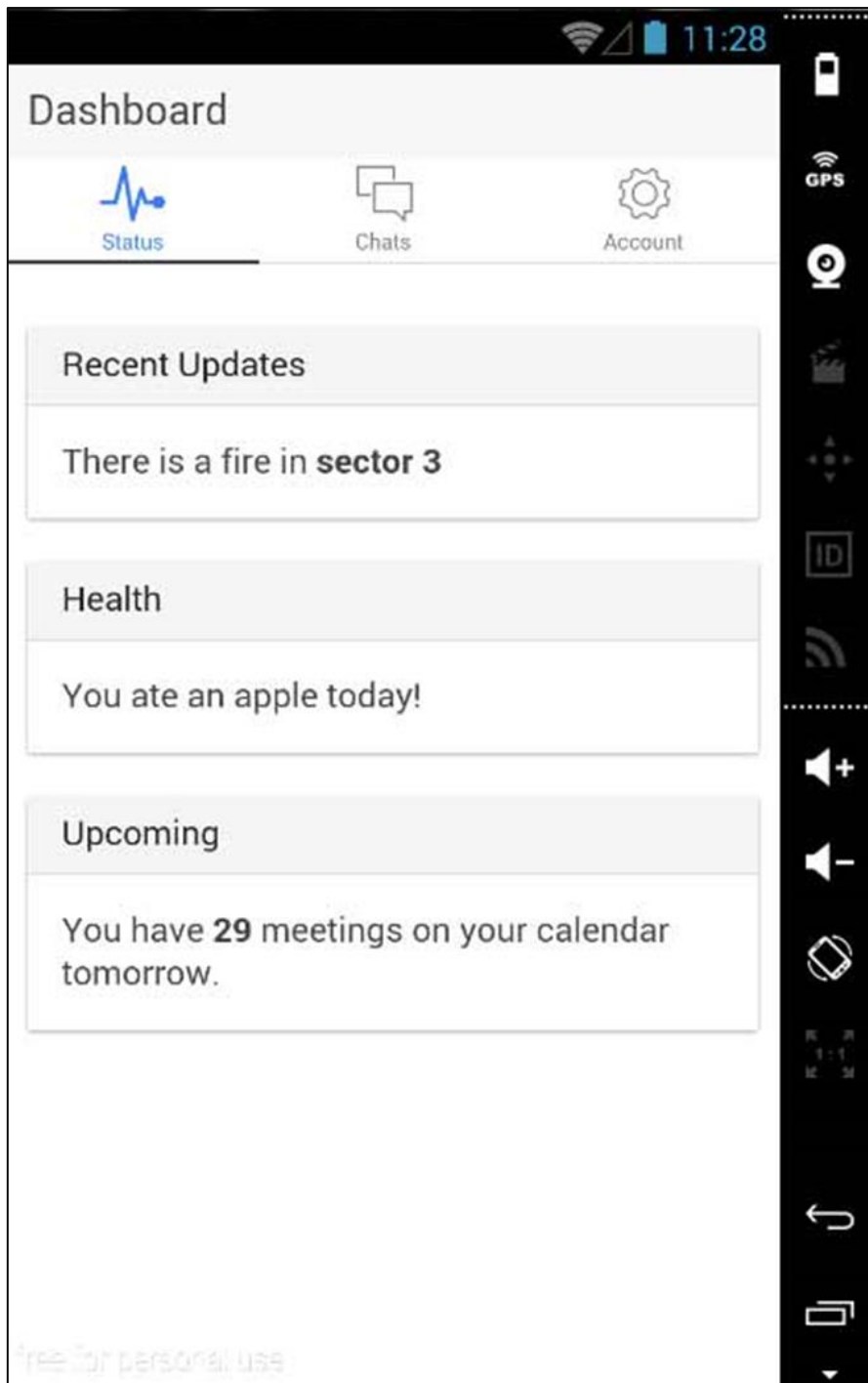
The next step is to build the app. If you have building errors after running the following command, you probably did not install the Android SDK and its dependencies.

```
C:\Users\Username\Desktop\myApp> ionic build android
```

The last step of the installation process is to run your app, which will start the mobile device, if connected, or the default emulator, if there is no device connected. Android Default Emulator is slow, so I suggest you to install [Genymotion](#) or some other popular Android Emulator.

```
C:\Users\Username\Desktop\myApp> ionic run android
```

This will produce below result, which is an Ionic Tabs App.



Blank App

If you want to start from the scratch, you can install the Ionic blank template. We will use the same steps that have been explained above with the addition of **ionic start myApp blank** instead of **ionic start myApp tabs** as follows.

```
C:\Users\Username\Desktop> ionic start myApp blank
```

The Ionic **Start** command will create a folder named **myApp** and setup the Ionic files and folders.

```
C:\Users\Username\Desktop> cd myApp
```

Let us add the Cordova project for the Android Platform and install the basic Cordova plugins as explained above.

```
C:\Users\Username\Desktop\myApp>ionic platform add android
```

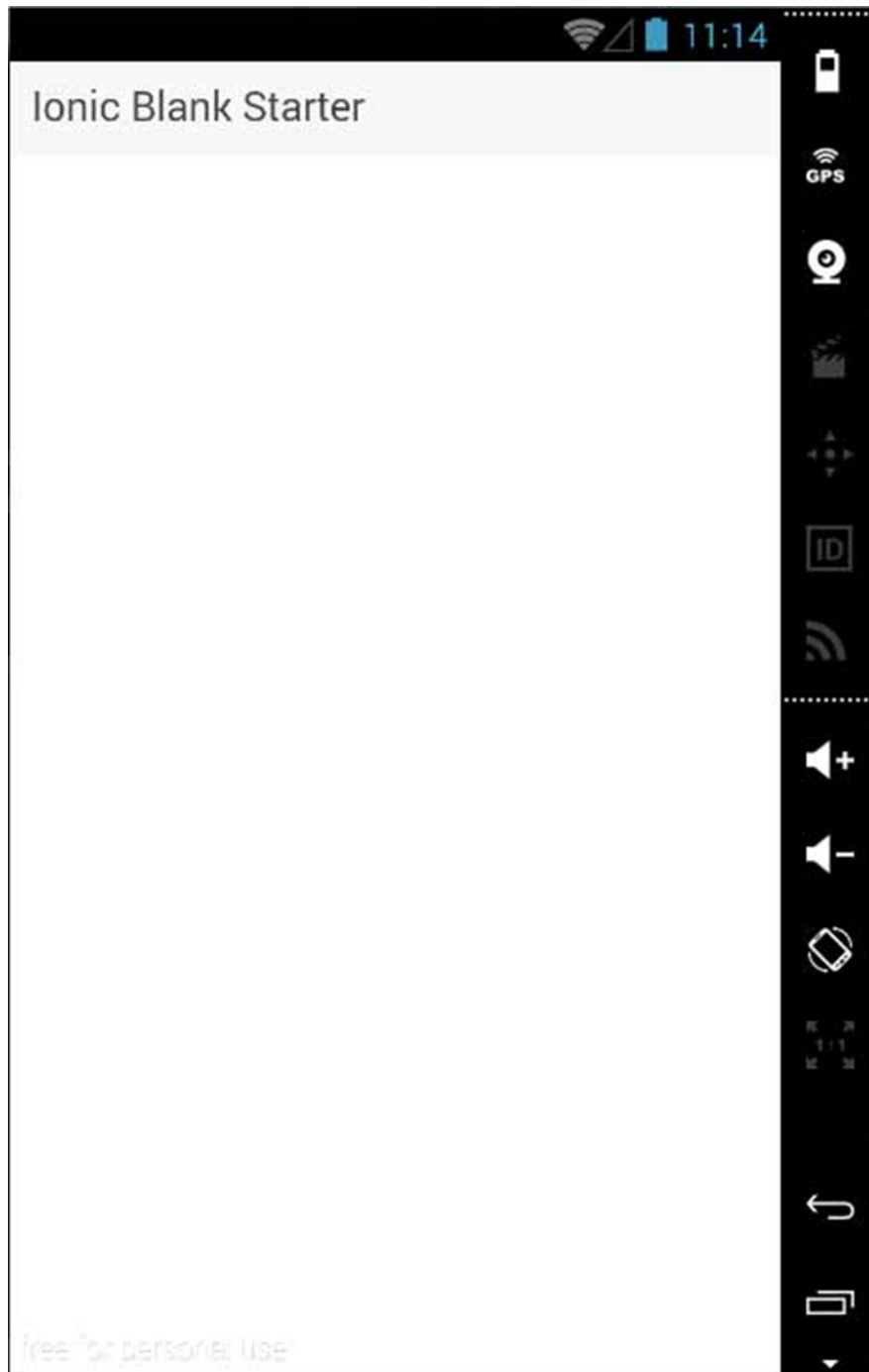
The next step is to build our app:

```
C:\Users\Username\Desktop\myApp> ionic build android
```

Finally, we will start the App as with the following code:

```
C:\Users\Username\Desktop\myApp> ionic run android
```

This will produce the following result, which is a Ionic Blank App.



Side Menu App

The third template that you can use is the Side Menu Template. The steps are the same as the previous two templates; we will just add **sidemenu** when starting our app as shown in the code below.

```
C:\Users\Username\Desktop> ionic start myApp sidemenu
```

The Ionic **Start** command will create a folder named **myApp** and setup the Ionic files and folders.

```
C:\Users\Username\Desktop> cd myApp
```

Let us add the Cordova project for the Android Platform and install the basic Cordova plugins with the code given below.

```
C:\Users\Username\Desktop\myApp> ionic platform add android
```

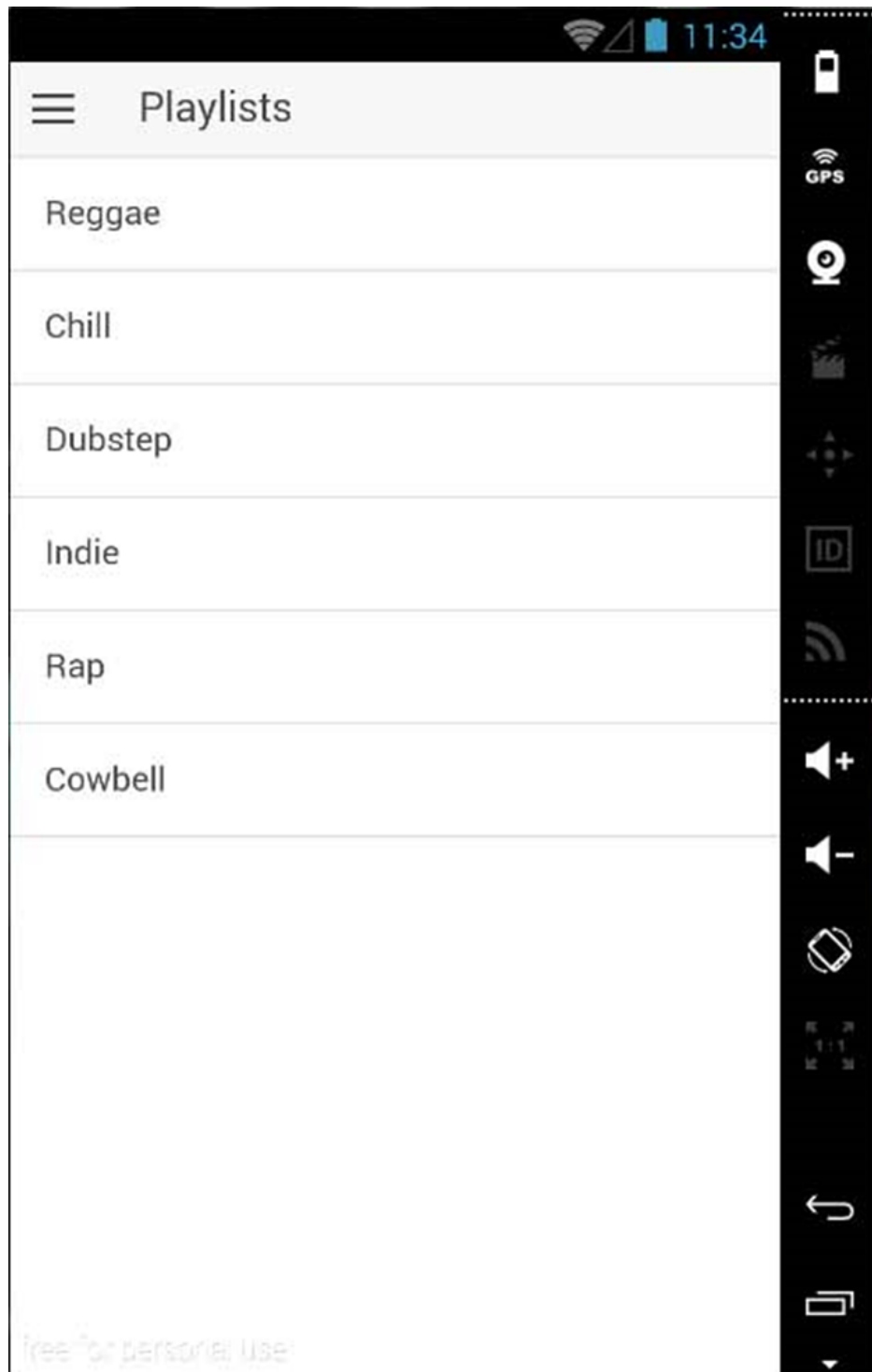
The next step is to build our app with the following code.

```
C:\Users\Username\Desktop\myApp> ionic build android
```

Finally, we will start the App with the code given below.

```
C:\Users\Username\Desktop\myApp> ionic run android
```


This will produce the following result, which is an Ionic Side Menu App.

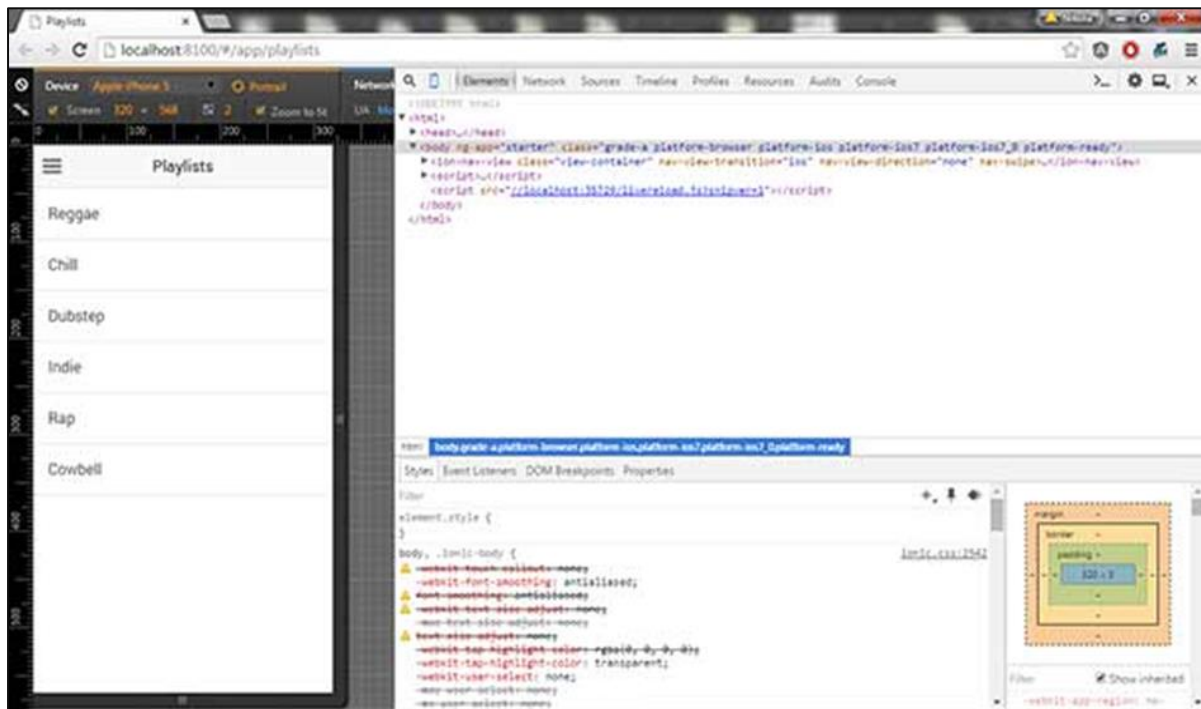


Test App in Browser

Since we are working with the JavaScript, you can serve your app on any web browser. This will speed up your building process, but you should always test your app on native emulators and devices. Type the following code to serve your app on the web browser.

```
C:\Users\Username\Desktop\myApp> ionic serve
```

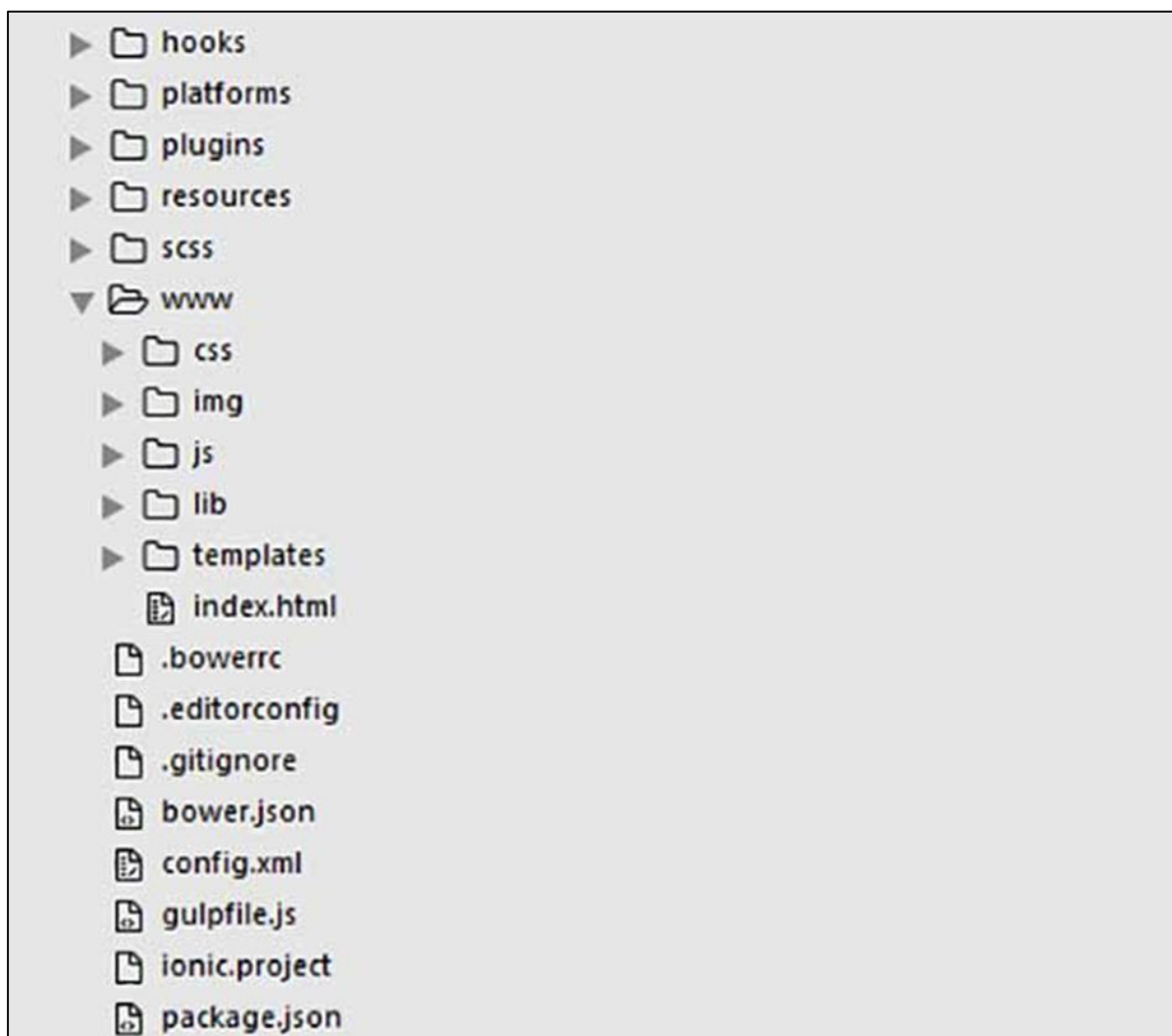
The above command will open your app in the web browser. Google Chrome provides the device mode functionality for mobile development testing. Press **F12** to access the developer console.



The top left corner of the console window click has the "Toggle Device Mode" icon. The next step will be to click "Dock to Right" icon in the top right corner. Once the page is refreshed, you should be ready for testing on the web browser.

Project Folder Structure

Ionic creates the following directory structure for all type of apps. This is important for any Ionic developer to understand the purpose of every directory and the files mentioned below:



Let us have a quick understanding of all the folders and files available in the project folder structure shown in the image above.

- **Hooks:** Hooks are scripts that can be triggered during the build process. They are usually used for the Cordova commands customization and for building automated processes. We will not use this folder during this tutorial.
- **Platforms:** This is the folder where Android and IOS projects are created. You might encounter some platform specific problems during development that will require these files, but you should leave them intact most of the time.
- **Plugins:** This folder contains Cordova plugins. When you initially create an Ionic app, some of the plugins will be installed. We will show you how to install Cordova plugins in our subsequent chapters.

- **Resources:** This folder is used for adding resources like icon and splash screen to your project.
- **Scss:** Since Ionic core is built with Sass, this is the folder where your Sass file is located. For simplifying the process, we will not use Sass for this tutorial. Our styling will be done using CSS.
- **www:** www is the main working folder for Ionic developers. They spend most of their time here. Ionic gives us their default folder structure inside 'www', but the developers can always change it to accommodate their own needs. When this folder is opened, you will find the following sub-folders:
 - The **css** folder, where you will write your CSS styling.
 - The **img** folder for storing images.
 - The **js** folder that contains the apps main configuration file (app.js) and AngularJS components (controllers, services, directives). All your JavaScript code will be inside these folders.
 - The **libs** folder, where your libraries will be placed.
 - The **templates** folder for your HTML files.
 - **Index.html** as a starting point to your app.
- **Other Files:** Since this is a beginner's tutorial, we will just mention some of the other important files and their purposes as well.
 - **.bowerrc** is the bower configuration file.
 - **.editorconfig** is the editor configuration file.
 - **.gitignore** is used to instruct which part of the app should be ignored when you want to push your app to the Git repository.
 - **bower.json** will contain the bower components and dependencies, if you choose to use the bower package manager.
 - **gulpfile.js** is used for creating automated tasks using the gulp task manager.
 - **config.xml** is the Cordova configuration file.
 - **package.json** contains the information about the apps, their dependencies and plugins that are installed using the NPM package manager.

In the next chapter, we will discuss the different colors available in Ionic open source framework.

Ionic – CSS Components

3. Ionic – Colors

Before we start with actual elements available in the Ionic framework, let us have a little understanding on how Ionic makes use of colors for different elements.

Ionic Color Classes

Ionic framework gives us a set of **nine predefined color classes**. You can use these colors or you can override it with your own styling.

The following table shows the default set of nine colors provided by Ionic. We will use these colors for styling different Ionic elements in this tutorial. For now, you can check all the colors as shown below:

Class	Description	Result
light	To be used for white color	
stable	To be used for light grey color	
positive	To be used for blue color	
calm	To be used for light blue color	
balanced	To be used for green color	
energized	To be used for yellow color	
assertive	To be used for red color	
royal	To be used for violet color	
dark	To be used for black color	

Ionic Color Usage

Ionic makes use of different classes for each element. For example, a header element will have **bar** class and a button will have a **button** class. To simplify the usage, we use different colors by prefixing element class in a color name.

For example, to create a blue color header, we will use a **bar-calm** as follows:

```
<div class="bar bar-header bar-calm">
  ...
</div>
```

Similarly, to create a grey color button, we will use **button-stable** class as follows.

```
<div class="button button-stable">
  ...
</div>
```

You can also use Ionic color class like any other CSS class. We will now style two paragraphs with a balanced (green) and an energized (yellow) color.

```
<p class="balanced">Paragraph 1...</p>
<p class="energized">Paragraph 2...</p>
```

The above code will produce the following screen:



We will discuss in detail in the subsequent chapters, when we create different elements using different classes.

Customizing Colors with CSS

When you want to change some of the Ionic default colors using CSS, you can do it by editing the **lib/css/ionic.css** file. In some cases, this approach is not very productive because every element (header, button, footer...) uses its own classes for styling.

Therefore, if you want to change the color of the "light" class to orange, you would need to search through all the elements that use this class and change it. This is useful when

you want to change the color of a single element, but not very practical for changing color of all elements because it would use too much time.

Customizing Colors using SASS

SASS (which is the short form of – **Software as a Service**) provides an easier way to change the color for all the elements at once. If you want to use SASS, open your project in the command window and type:

```
C:\Users\Username\Desktop\tutorialApp> ionic setup sass
```

This will set up SASS for your project. Now you can change default colors by opening the **scss/ionic.app.scss** file and then typing in the following code before this line – **@import "www/lib/ionic/scss/ionic";**

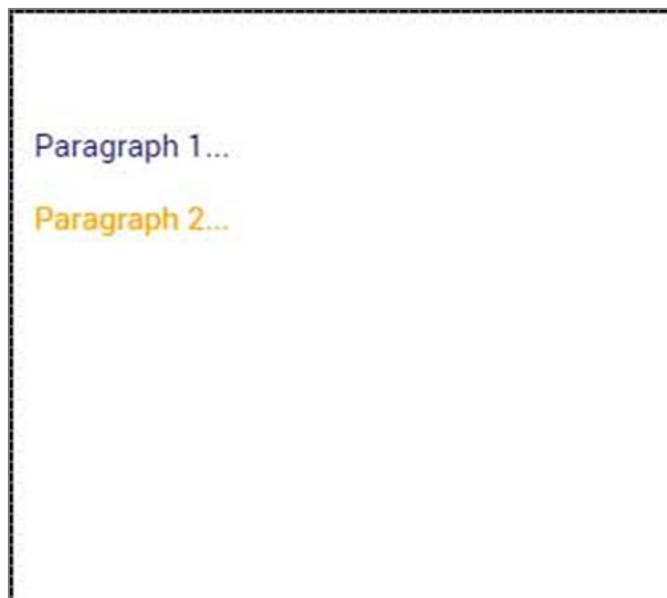
We will change the balanced color to dark blue and the energized color to orange. The two paragraphs that we used above are now dark blue and orange.

```
$balanced: #000066 !default;  
$energized: #FFA500 !default;
```

Now, if you use the following example:

```
<p class="balanced">Paragraph 1...</p>  
<p class="energized">Paragraph 2...</p>
```

The above code will produce the following screen:



All the Ionic elements that are using these classes will change to dark blue and orange. Take into consideration that you do not need to use Ionic default color classes. You can always style elements the way you want.

Important Note

The **www/css/style.css** file will be removed from the header of the index.html after you install SASS. You will need to link it manually if you still want to use it. Open index.html and then add the following code inside the header.

```
<link href="css/style.css" rel="stylesheet">
```

4. Ionic – Content

Almost every mobile app contains some fundamental elements. Usually those elements include a header and a footer that will cover the top and the bottom part of the screen. All the other elements will be placed between these two. Ionic provides **ion-content** element that serves as a container that will wrap all the other elements that we want to create.

This container has some unique characteristics, but since this is a JavaScript based component which we will cover in the later part of this [tutorial](#).

```
<div class="bar bar-header">
  <h1 class="title">Header</h1>
</div>

<div class="list">
  <label class="item item-input">
    <input type="text" placeholder="Placeholder 1">
  </label>

  <label class="item item-input">
    <input type="text" placeholder="Placeholder 2">
  </label>
</div>

<div class="bar bar-footer">
  <h1 class="title">Footer</h1>
</div>
```

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>