Ionic Framework By Example shows you how to get started with the Ionic framework immediately. But it doesn’t just give you instructions and then expect you to follow them. Instead, it demonstrates what Ionic is capable of through three practical projects you can follow and build yourself.

From a basic to-do list app and a London Tourist App, to a complete social media app, all three projects have been designed to help you learn Ionic at its very best. From setting up your project to developing on both the server side and the frontend, and best practices for testing and debugging your projects, you’ll quickly become a better mobile developer, delivering high performance mobile apps that look awesome.

What you will learn from this book

- Learn Ionic by creating three complete mobile applications
- Get to know the Ionic CLI
- Run and test your Ionic app on the browser and on real device
- Add basic and advanced features to an Ionic application
- Connect an Ionic app with a Firebase backend
- Customize look and feel of an Ionic application with SCSS

Who this book is written for

This book is for anyone who wants to see Ionic in action and find out how it could transform the way they build mobile apps. If you’re a JavaScript web developer, you’ll be building great projects in no time.


Sani Yusuf
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'First Look at Ionic'
- A synopsis of the book’s content
- More information on Ionic Framework By Example
Sani Yusuf is the founder of Haibrid, a company focused on creating innovative mobile solutions with hybrid technologies operating from London, England. Starting his career as a web developer, he moved on to native development of both desktop and mobile applications before developing a focus on hybrid mobile apps due to their philosophy of reusability and cross-platform operability.

When not managing his company, Sani spends his time public speaking, writing, and endlessly researching new educational and healthcare mobile solutions. Although a Nigerian by birth, Sani moved to the UK in 2010 to pursue a degree in computer science, having a great interest in healthcare and educational advancement with mobile technology.

A lover of the movie "Avatar" and a speaker of four languages, namely French, English, Arabic, and Hausa, Sani spends his free time travelling, watching movies, and watching Arsenal football club being great.

During his career, Sani has worked with companies like Anritsu, Microsoft, Huddlebuy, Envato, and more recently, Nexercise, with some of his works awarded by Microsoft, Samsung, and Aimia Ltd.

He has also written Windows Phone Beginner Series, an online tutorial series for Envato.
Ionic Framework By Example is a step-by-step guide that covers the very basics of Ionic aiming to equip the reader with all the necessary knowledge needed to understand and create Ionic apps. You will start off by learning a bit about the history of Ionic, and then slowly learn to get it set up and work with its great features. You will learn to work with Ionic and create four different Ionic apps, with each app teaching you different important features of Ionic. You will also learn to connect your app to a database using Firebase. This book will also provide you with links to some great resources to further your quest for more advanced Ionic knowledge.

What this book covers

Chapter 1, First Look at Ionic, covers a brief history of Ionic and aims to make the user understand exactly what Ionic is, briefly exposing some of its features to the user. This chapter will also teach you how to set up Ionic on your computer.

Chapter 2, To-Do List App, will help the reader to create their first Ionic application, a simple to-do list application. The user will also write their very first lines of Ionic code and get to understand what an Ionic project looks like.

Chapter 3, Running Ionic Apps, equips you with the necessary knowledge of how to see your Ionic apps in action in different ways. You will run and test your app on a browser, on a mobile device using the Ionic view app, and also on a real mobile device.

Chapter 4, Ionic Components, teaches some really cool components that are part of Ionic. You will create a more advanced to-do list application that will have some Ionic components that will enable you to create some complex list items in your application.

Chapter 5, The London Tourist App, creates a new type of Ionic application. You will create an application that will hold some very cool tourist destinations in the city of London.
You will also learn to query data from a JSON resource and consume this data in your application.

Chapter 6, Advanced Ionic Components, helps the reader learn to implement some more advanced Ionic components. You will learn to implement Ionic Popover and Modal windows in your application.

Chapter 7, Customizing the App, focuses on customizing an Ionic application. The Ionic SDK comes by default with some great tools that make it easy to customize your application to fit the design guides of your brand.

Chapter 8, Building a Simple Social App, focuses on learning how to create an Ionic application that has tabs using the Ionic tabs template. You will also have a look at some of the things that make up the tabs template and learn how to add features to it.

Chapter 9, Connecting to Firebase, focuses solely on learning how to use Firebase to integrate a backend to our Ionic application. You will also be using the tabs-app that we created in Chapter 8, Building a Simple Social App, to learn to integrate Firebase into our application.

Chapter 10, Roundup, gives an overview of the important things that we haven't covered yet about Ionic that you might find very useful. You will also learn some useful tips about Ionic and discover some great tips on how to make even better use of Ionic to develop great apps.
Chapter 1

First Look at Ionic

Before we begin this book, it is very important that we understand just exactly what we are dealing with. The best way to understand this is by having a short history on mobile development, in general, and understand how tools like Ionic help mobile developers create beautiful mobile apps.

The beginning

The year 2006 saw the beginning of the smartphone era with the launch of the iPhone by Apple. By 2008, Google had launched its answer to Apple’s iOS operating system. This new operating system was called Android, and by 2010, it was clear that smartphones running iOS and Android dominantly covered the mobile ecosystem. Fast forward to today, the dominance of iOS and Android is not so different even though Windows for mobile by Microsoft has made some gains on the mobile front. It is fair to say that Android, iOS, and Windows make up the majority of the ecosystem with the first two at the forefront by a large margin.

The launch of the smartphone era also gave birth to the concept of mobile applications. Mobile apps are the medium by which we deliver and obtain most of our content on mobile phones. They are great and everyone with a smartphone pretty much has a number of apps downloaded on their devices to perform specific actions or achieve specific goals. This was massive for developers, and the software vendors also provided tools that enabled developers to create their own third-party mobile apps for users. We refer to these applications, built using the tools provided by the software vendors, as native mobile applications.
The problem

As great as mobile apps are, there is a small problem with how they are developed. Firstly, for each mobile development platform, the software vendor provides its own unique set of tools to build applications for its platforms. We know these tools as SDKs. The following table shows how each platform differs in terms of tools and SDK options to create native mobile apps for their ecosystems:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>SDK</th>
<th>Programming language</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>iOS SDK</td>
<td>Objective-C/Swift</td>
</tr>
<tr>
<td>Android</td>
<td>Android SDK</td>
<td>JAVA</td>
</tr>
<tr>
<td>Windows for mobile</td>
<td>Windows SDK</td>
<td>.NET</td>
</tr>
</tbody>
</table>

To make a clear statement, we are not trying to downplay the use of native tools. As noted earlier, native tools are great but come with a great cost and time constraint. Firstly, you are unable to build the same app for different platforms with the same set of tools. For the Android version of your app, you will need a team of skilled android developers. For the iOS version of your app, you will need a team of Objective-C or Swift developers to create the iOS version of the same app. Also, there is no code sharing between these two teams, meaning that a feature developed on one platform will have to be completely developed on the other platform again. This is highly inefficient in terms of development and very time consuming.

Another problem is that because you are hiring two separate teams that are completely independent of one another even though they are both trying to create the same thing, you are left with a growing cost. For example, if you decided you wanted to create a Windows for mobile version of your mobile app, you will need to recruit another team of .NET developers and they will have to build everything present on the other existing platforms from scratch since they cannot reuse any of the already built tools.

For a company like Facebook, which makes revenue in the billions, it might make sense to go down the native path as cost and talent for native development would probably not be a part of their concern. However, for the most part, not everyone building or trying to build a mobile app is a company like Facebook. Most people want to get a simple, great, powerful app out there as quick as possible. Furthermore, some of these people want to use their preexisting skill set to build apps for multiple platforms without having to learn new programming languages.
Before mobile applications, web apps ruled the world for the most part. We had more people developing for the web technologies consisting mostly of HTML, CSS, and JavaScript. One great thing we got used to with the web was that it was platform independent. This meant that as long as you had a browser application on any device, you were able to interact with any web application without any problem.

So when mobile apps came, it was a big change for most web developers because with mobile apps, each platform was self-dependent, and apps made for one platform would not work for another platform.

Apache Cordova

Apache Cordova is a technology that lets any web application be packaged as a native mobile application while also providing access to device features. Thanks to Adobe and the open source community, this technology has seen great growth over the years and more and more apps are being built with Cordova. The apps built with Cordova are commonly referred to as hybrid applications. But what is a hybrid app?
A hybrid application in the context of Cordova is actually a web app that lives within the thin container of a native mobile application.

In reality, the only difference between a native mobile app and a web application in terms of what they can do is the fact that the native mobile app has access to the device hardware features.

In truth, a hybrid app is actually a native app that serves up a web application on the phone's web view. It behaves and acts like a normal application and has complete device access, thanks to Cordova.

However, the main advantage that Cordova has over native development techniques is that you only have to maintain one code base, and can use that same code base to build applications for multiple platforms. This was and still is the main selling point of Cordova to date, as with this technique you are provided with a big cost and time saving advantage.

**Early hybrid apps**

When the first few hybrid apps started rolling out, there were a couple of problems that seemed to persist. The first problem was that a lot of people complained that these hybrid applications did not have the same user interface and user experience as native mobile apps. The major reason for this was that when building a native app, all the building blocks are already provided for you. For example, we have things like pre-provided animations, swipe gestures, tabs, and so on. Hybrid apps failed to provide similar features like these because on the web, all we have is HTML, CSS, and JS. There is no pre-provided component for things like animations, gestures, and tabs.
This meant that pretty much anyone trying to build a hybrid app had to build these features from scratch. This was not very good and different people had different implementations of the same features. As a result, a lot of applications that were built by the hybrid way were not so visually appealing. What we needed was a framework that was centrally maintained that provided us with all the tools we needed to build features that native apps had with web technologies.

**What is Ionic?**

Ionic is a framework that lets you build hybrid mobile applications with web technologies like HTML5, CSS, and JavaScript. But that is not where it stops with Ionic. Ionic provides you with components that you can use to build native-like features for your mobile applications. Think of Ionic as the SDK for making your Hybrid mobile application. Most of the features you have on a native app such as modals, gestures, popups, and many more, are all provided to you by Ionic and can be easily extended for new features or customized to suit your needs.

Ionic itself does not grant you the ability to communicate with device features like GPS and camera; instead, it works side-by-side with Cordova to achieve this. Another great feature of Ionic is how loosely coupled all its components are. You can decide to use only some of Ionic on an already existing hybrid application if you wish to do so.

The Ionic framework is built with AngularJS, which is arguably the most well-tested and widely-used JavaScript framework out there. This feature is particularly powerful as it gives you all the goodness of Angular as part of any Ionic app you develop. In the past, architecting hybrid applications proved to be difficult, but with Angular, we can create our mobile applications using the Single Page Application (SPA) technique. Angular also makes it really easy to organize your application for the development and working across teams while providing you the possibility of easily adding custom features or libraries.

**Short history of Ionic**

Before we dive in, first let's revisit what we already know about hybrid applications and how they work. Remember that a hybrid mobile application is simply a web application that runs in a web view, within a thin native wrapper environment.
Also remember that native apps came with already built components that enabled you to create beautiful user interfaces for mobile applications. Since hybrid apps used web technologies, there was no SDK or components provided for creating mobile UIs. The Ionic team saw this problem and created a solution in the form of the Ionic framework. The Ionic framework provides UI components to build beautiful hybrid applications.

Features of Ionic

Ionic provides you with a lot of cool neat features and tricks that help you create beautiful and well functioning hybrid apps in no time. The features of Ionic come under three categories:

- CSS features
- JavaScript features
- Ionic CLI

CSS features

To start off, Ionic comes stock with a great CSS library that provides you with some boilerplate styles. These Ionic CSS styles are generated with SASS, a CSS preprocessor for more advanced CSS style manipulation.

Some of the cool CSS features that come built-in with Ionic include:

- Buttons
- Cards
- Header and footers
- Lists
- Forms elements
- Grid system

All these features and more, are already provided for you and are easily customizable. They also have the same look and feel that native equivalents have so you will not have to do any editing to make them look like native components.
JavaScript features

The JavaScript features are at the very heart of the Ionic framework and essential for building Ionic apps. They also consist of other features that let you do things from under the hood like customize your application or even provide you with helper functions you can use to make developing your app more pleasant. A lot of these JavaScript features actually exist as HTML custom elements that make it easy to declaratively use these features.

Some of these features include:

- Modal
- Slide box

- Action sheet
First Look at Ionic

- Side menu

- Tabs
• Complex lists

All the JavaScript features of Ionic are built with Angular, and most can be easily plugged in as Angular directives. Each of them also perform different actions that help you achieve specific functions and are all documented in the Ionic website.

The Ionic CLI

This is the final part that makes up the three major arms of the Ionic framework. The Ionic CLI is a very important tool that lets you use the Ionic commands via the command line/terminal. It is also with the Ionic CLI that we get access to some Ionic features that make our app development process more streamlined. It is arguably the most important part of Ionic and it is also the feature you will use to do most actions.

Ionic CLI features let you do the following:

• Create Ionic projects
• Issue Cordova commands
• Development and testing
• Ionic splash/Icon generator
The Ionic CLI is a very powerful tool and most of the time, it is the tool we will be using throughout this book to perform specific actions. This is why the first thing we are going to do is set up the Ionic CLI.

**Setting up Ionic**

The following steps will give a brief of how to setup Ionic:

1. **Install NodeJS**: To set up Ionic, the first thing you will need to do is to install NodeJS on your computer so you can have access to Node Package Manager (NPM). If you already have node installed on your computer, you can skip this step and go to step 2. To install NodeJS on your computer, perform the following steps:
   1. Go to [www.nodejs.org](http://www.nodejs.org) and click on the latest stable version for your computer. That should download the latest version of NodeJS on your computer. Don't worry if you are on Mac, PC, or Linux, the correct one for your operating system will be automatically downloaded.
   2. After the download is finished, install the downloaded software on your computer. You might need to restart your computer if you are running Windows.
   3. Open up the terminal if you are on Mac/Linux or the Windows command line if you are on a Windows machine. Type the command `node --v` and press Enter.

   You should see the version number of your current installation of NodeJS. If you do not see a version number, this might mean that you have not correctly installed NodeJS and should try running step 1 again.

2. **Install Ionic CLI**: The next step is to use NPM to install the Ionic CLI.
   1. Open a new terminal (OS X and Linux) or command-line (Windows) window and run the following command: `npm install ionic -g`. If you are on Linux/OS X, you might need to run `sudo npm install ionic -g`. This command will aim to install Ionic globally.
2. After this has finished running, run the command `ionic -v` on your terminal/command line and press Enter.

You should see a version number of your Ionic CLI. This means that you have Ionic installed correctly and are good to go. If you are on a Windows machine, you might need to restart your machine to see the version number appear.

If you did not see a version number, then you do not have Ionic installed correctly on your machine and should do step 2 again.

**Summary**

In this chapter, we started off by getting to know a bit of background about mobile applications in general. We learned how native mobile applications work, how they are built with native SDKs, and how each platform is built with a completely different set of tools without any resource sharing between them all. We then went ahead and discussed briefly about Apache Cordova and how it aimed to solve the problem of cross-platform development.

We then discussed exactly what Ionic means and what problems it aims to solve. We also got to discuss the CSS, JS, and Ionic CLI features of the Ionic framework lightly.

In the next chapter, we will be creating our very first Ionic application with the Ionic CLI, and we will create a nice to-do list style application with some great Ionic features.
Where to buy this book
You can buy Ionic Framework By Example from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.