Hybrid Mobile Apps with Ionic and AngularJS

# Ionic IN ACTION

Jeremy Wilken

MEAP

MANNING

**MEAP Edition**
**Manning Early Access Program**
**Ionic in Action**
**Version 1**

Copyright 2014 Manning Publications

For more information on this and other Manning titles go to
[www.manning.com](http://www.manning.com)

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

# Welcome

Thanks for purchasing the MEAP of *Ionic in Action: Hybrid Mobile Apps with Ionic and AngularJS*. I hope you find it useful. I'm looking forward to hearing from you in the Author Online forum–let me know how to make this book better!

The Ionic Framework is a combination of tools and utilities that enable developers to quickly build hybrid mobile apps using the same technologies used to build websites and web applications. With the Ionic Framework and AngularJS, you'll begin building hybrid mobile apps that look and feel just like a native mobile app.

By the end of the book you should understand:

- How to setup your development environment for Ionic
- How to use AngularJS as the foundation for your app
- How to use Ionic to build stunning mobile interfaces
- How to employ Ionic Services to further manage the UI
- How to use Cordova to integrate with device features, like GPS
- How to test and deploy your app to stores

The book is divided into three parts.

In Part I, you'll install your working and testing environment for either Android or iOS, get familiar with it, and build your first app. I'll also teach you some foundations of AngularJS that we'll be using in this book. I'll explain things clearly for you and have created several graphics to give you an idea of how everything fits together.

In Part II, you'll learn how to use Ionic to create and manipulate a user interface optimized for both users and devices. Your users will think they are using an app developed using the native SDK. You'll be able to build complex interfaces using Ionic's components, and see how they work together.

In Part III, you will learn some advanced Ionic and Angular development skills and how to test and deploy your app. I'll show you how to use Cordova to access the device's features such as the camera. We'll work through different ways to test your app to get it deployment ready and I'll also show you how to configure and deploy your app for both the iOS App Store and the Android Play Store.

Tell me what you think of what I've written and what you'd like to see in the rest of the book. Your feedback through the AO forum will be very valuable as I write and improve *Ionic in Action: Hybrid Mobile Apps with Ionic and AngularJS*.

Thanks again for your interest and for purchasing the MEAP!

—Jeremy Wilken

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

# brief contents

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

# 1 Ionic and Hybrid Apps

***This chapter covers***

- Why to choose Ionic and how it benefits you.
- What Ionic is and how it uses AngularJS and Cordova.
- Why hybrid apps are an ideal choice for mobile development.
- Introduction and requirements for Android and iOS platforms

Congratulations, you've picked up the best resource to begin building hybrid mobile apps that look and feel just like a native mobile app using the Ionic Framework. The Ionic Framework is a combination of tools and utilities that enable developers to quickly build hybrid mobile apps using the same technologies used to build websites and web applications. Ionic works by embedding a web application inside of a native app by using Cordova. It is designed to work together with AngularJS to create a web application for the mobile environment, and includes support for mobile features like responding to touch inputs and user interface controls.

This book aims to give developers the skills necessary to build amazing hybrid apps using the same skills from building web applications. You might be a freelance developer who would like to offer clients the service of building custom mobile apps or a project manager in a Fortune 500 company trying to determine the viability of building a hybrid app for your company. There is a wide range of reasons why you might want to learn about building hybrid mobile apps. Before we get too far along, we should clarify what Ionic is and why it is a solid choice for building hybrid mobile apps.

## 1.1 What is Ionic?

The Ionic Framework is a combination of technologies and utilities designed to make building hybrid mobile apps fast, easy, and beautiful. Ionic doesn't run entirely by itself, it is built with AngularJS as the web application framework and is designed to work with Cordova for the

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

building of the native app. We'll dig into each in more detail later, but figure 1.1 shows an overview of the several technologies and how they stack. Let's take a moment to cover the basics of how the technology stack works on a device.

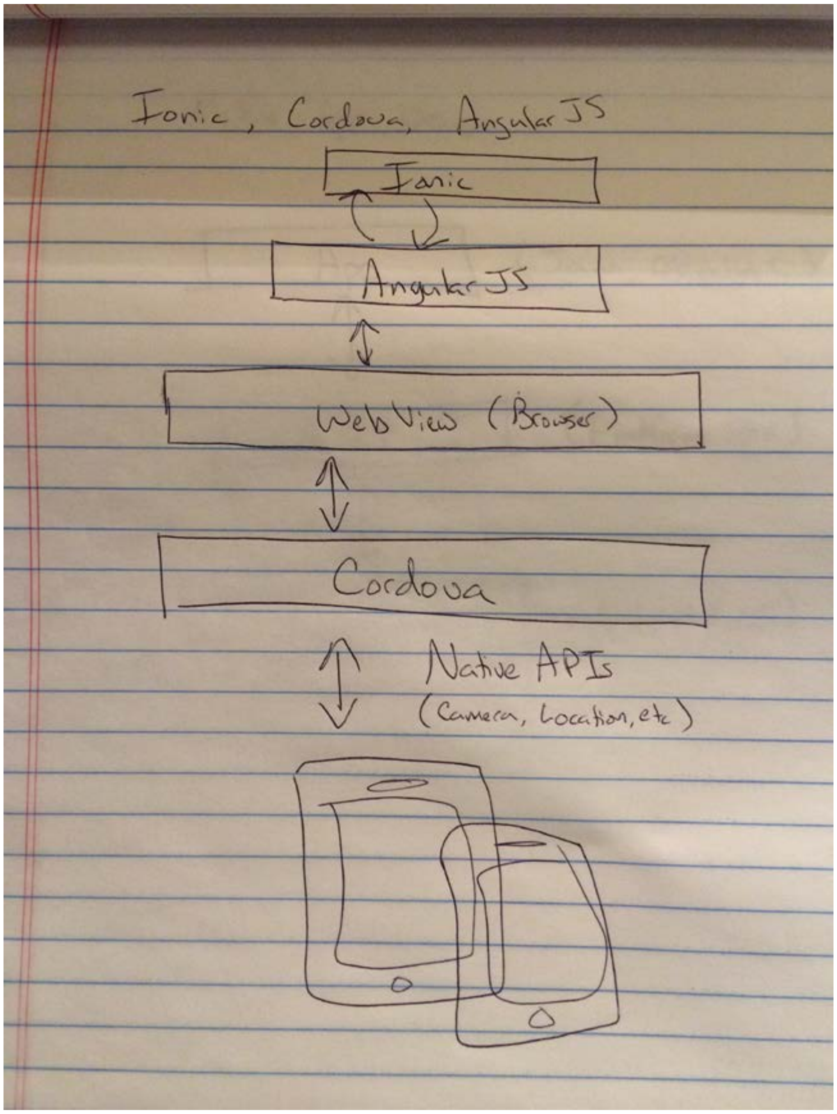


Figure 1.1 The stack of technologies used with the Ionic Framework, and how they fit together.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

In figure 1.1, the stack begins with the device. Imagine this is an iPhone running iOS or a Nexus 10 running Android. The device contains the operating system that manages the installation of apps, which are downloaded from the platform’s store. The operating system also provides a set of APIs for apps to use to access various features, like the GPS location, contacts list, or camera.

The next layer is Cordova, which is a utility for creating a bridge between the platform and our application. It creates a native mobile app that can be installed, and contains what is called a WebView (essentially an isolated browser window), which the web application will run inside. This generated native app is able to access both the web application and the native platform, since Cordova provides a JavaScript API for the web application to use to communicate with the native device. This is primarily handled behind the scenes, and Cordova ultimately generates the native app for you.

The last two technologies are AngularJS and Ionic. Inside of the WebView our AngularJS web application runs. AngularJS is primarily used to manage the web application’s logic and data. Ionic is built on top of AngularJS, and is primarily used to design the user interface and experience. This includes the visual elements such as tabs, buttons, and navigation headers.

The core of Ionic is the user interface controls described above. However Ionic also includes a number of additional utilities and features that help manage your app from creation to previewing to deployment.

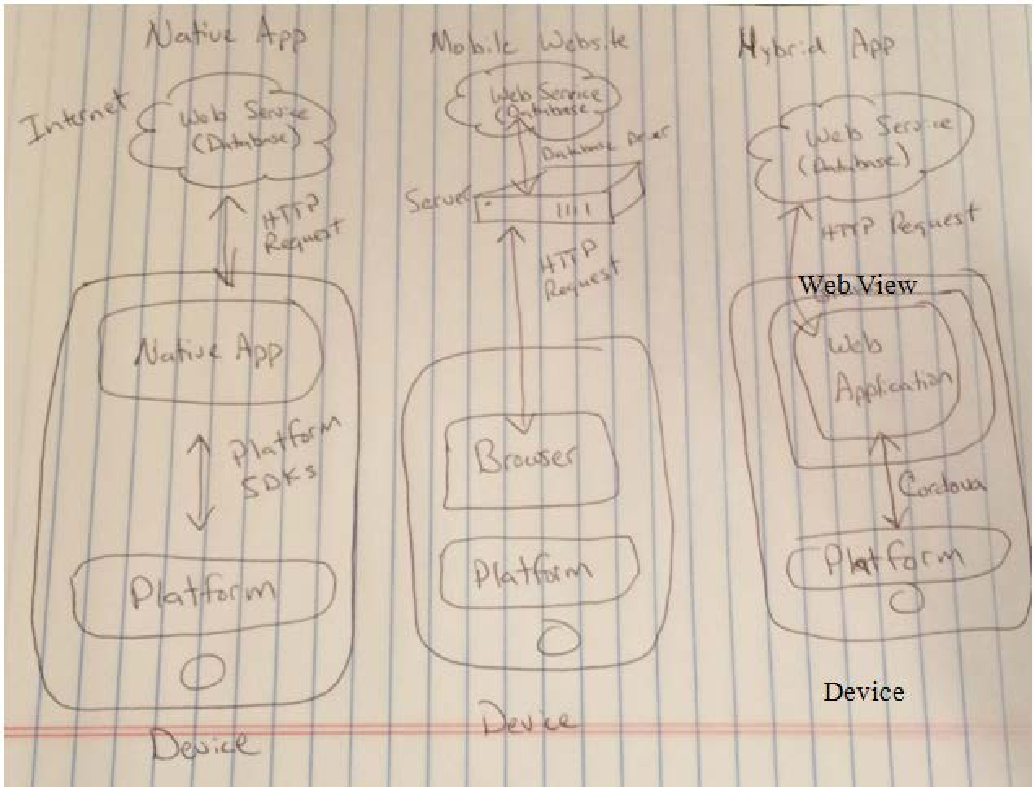
Now that you have a bird’s eye view of Ionic and the technology, let’s look a little closer at the types of mobile experiences and why Ionic’s approach is beneficial.

## 1.2 Types of mobile experiences

It is important to understand there are several ways to build applications for mobile devices, and each has strengths and weaknesses. There are three basic types, native apps, mobile websites, and hybrid apps. We'll look at each of these in detail to clarify the differences.

In the following figure, you can see how the three types compare in design and architecture. It shows also how the apps would access a database or web service API to load data.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931



Native apps, mobile websites, and hybrid app architectures compared side by side.

### 1.2.1 Native Mobile Apps

Native apps are written using the default language for the mobile platform, which is Objective C for iOS (soon to be Swift for iOS8) or Java for Android. Native apps are compiled and execute directly on the device. Using the platform SDK (API), the app can communicate with the platform to access device data or load data from an external website using http requests.

Both iOS and Android provide a set of tools to enable developers to leverage the platform features in a controlled manner through predefined APIs. There are tools, both official and unofficial, which can also aid in the development of native apps. It is also common for developers to use frameworks in their native app to make development easier.

#### Native App Advantages

The native app comes with a number of benefits over the other types. The benefits revolve around being tightly integrated with the device platform.

- **Native APIs.** They can use the native APIs directly in the app, making the tightest connection to the platform.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

- **Performance.** Native apps can experience highest levels of performance when they are designed well.
- **Same environment.** They use the same environment as the platform, which is helpful for developers familiar with the languages used.

#### Native App Disadvantages

The disadvantages of native apps are generally the level of difficulty in developing and maintaining them.

- **Language requirements.** Requires proficiency in the platform language (for example Java) and knowledge how to use APIs.
- **Not cross platform.** Native apps can only be developed for one platform at a time.
- **High level of effort.** Typically native apps are more work and overhead to build, increasing costs.

Native apps may be best suited for developers who have a command of Java and Objective C, or for teams with extensive resources and a need for the benefits.

### *1.2.2 Mobile Websites (Web Apps)*

Mobile websites are applications that work well on a mobile device, but are accessed through the mobile browser. Sometimes they are called Web Apps. Most simply, they are websites viewed on a mobile device in a mobile browser, with the exception of being designed to fit a mobile device screen size.

Some websites have a unique version of the normal website that have been developed specifically for use on a mobile device. Perhaps you've visited websites that redirect you on a mobile device to a limited feature application, often on a subdomain such as http://m.ebay.com. In other examples the design adjusts to the form factor and screen size in a technique called responsive design, such as http://www.bostonglobe.com. Depending on the site of the browser window, the website reflows the page to fit better on a smaller screen and perhaps even hides content.

#### Mobile Websites Advantages

Mobile websites enjoy a number of benefits, primarily in the level of effort and compatibility on devices.

- **Maintainability.** They are easy to update and maintain without the need to go through an approval process or updating installations on devices.
- **No installation.** Since it exists on the internet, it doesn't require installation on mobile devices.
- **Cross platform.** Any mobile device has a browser, allowing your application to be accessible from any device.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

#### MOBILE WEBSITES DISADVANTAGES

Mobile websites run inside of a mobile browser, which is the major cause of limitations and disadvantages.

- **No native access.** Since it is run in the browser, it has no access to the native APIs or the platform, just the APIs provided by the browser.
- **Requires keyboard to load.** The user has to type address in a browser to find or use your mobile website, and mobile browsing is on the decline.
- **Limited user interface.** It is difficult to create touch friendly applications, especially if you have a responsive site that has to work well on desktops.

Mobile websites can be important even if you have a mobile app, depending on your product or service. Research shows users spend much more time using apps compared to the mobile browser, so mobile websites tend to have a lower engagement.

### *1.2.3 Hybrid Apps*

A hybrid app is a mobile app that contains a web view (essentially an isolated browser instance) to run a web application inside of a native app, using a native app wrapper that can communicate with the native device platform and the web view. This means web applications can run on a mobile device and have access to the device, such as the camera or GPS features. Hybrid apps are possible because of tools that have been created that facilitate the communication between the web view and the native platform. These tools are not part of the official iOS or Android platforms, but are third party tools such as Apache Cordova, which is used in this book. When a hybrid app is built, it will be compiled, transforming your web application into a native app.

#### HYBRID APP ADVANTAGES

- **Cross platform.** You can build your app once, and deploy it to multiple platforms with minimal effort.
- **Same skills as web development.** It allows you to build mobile apps using the same skills already used to develop websites and web applications.
- **Access to device.** Since the web view is wrapped in a native app, your app has access to all of the device features available to a native app.
- **Ease of development.** They are easy and fast to develop, without the need to constantly rebuild to preview. You also have access to the same development tools used for building websites.

Hybrid apps provide a robust base for mobile app development while still being able to use the web platform. You can build the majority of your app as a website, but anytime you need access to a native API the hybrid app framework can provide a bridge to access that API by using JavaScript. You can detect swipes, pinches, and other gestures like you can detect clicks or keyboard events.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

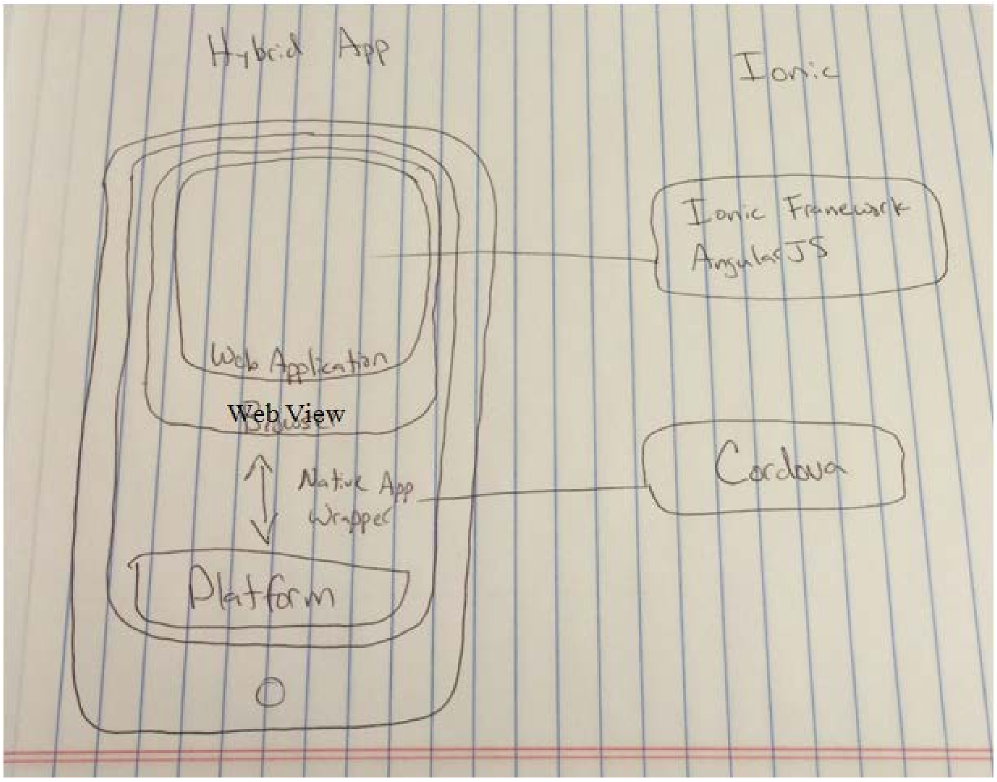
### Hybrid App Disadvantages

- **Web view limitations.** The application can only run as well as the web view instance, which means performance is tied to the quality of the platform's browser.
- **Native via plugins.** Access to the native APIs you need may not be currently available, and may require additional development to make a plugin to support it.
- **No native user interface controls.** Without a tool like Ionic, developers would have to create all of the user interface elements. Luckily Ionic provides a rich set of controls that look and feel like the native controls.

With Ionic, we'll be building hybrid apps so we can leverage the knowledge and skills with which web developers are already familiar.

## 1.3 Understanding how the Ionic stack works

There are several technologies that can be used when building hybrid apps, but with Ionic there are three primary technologies: the Ionic Framework, AngularJS, and Cordova. Let's look at each one more closely.



How Ionic, Angular, and Cordova work together for a hybrid app

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders. http://www.manning-sandbox.com/forum.jspa?forumID=931

### 1.3.1 Ionic Framework

The Ionic Framework (often referred to as Ionic) is a number of things, but the primary feature is that it provides a set of user interface controls that are missing from HTML but are common on mobile apps. Imagine we are building a weather app that shows current conditions based on the user's location. Ionic provides a number of user interface components such as a slidebox that allows a user to swipe between several boxes of information like temperature, forecasts, and weather maps. These components are built with a combination of CSS, HTML, and JavaScript, and they behave like the native controls you are accustomed to using. Common examples include:

- Side menus that slide in from the side
- Toggle buttons
- Mobile tabs

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

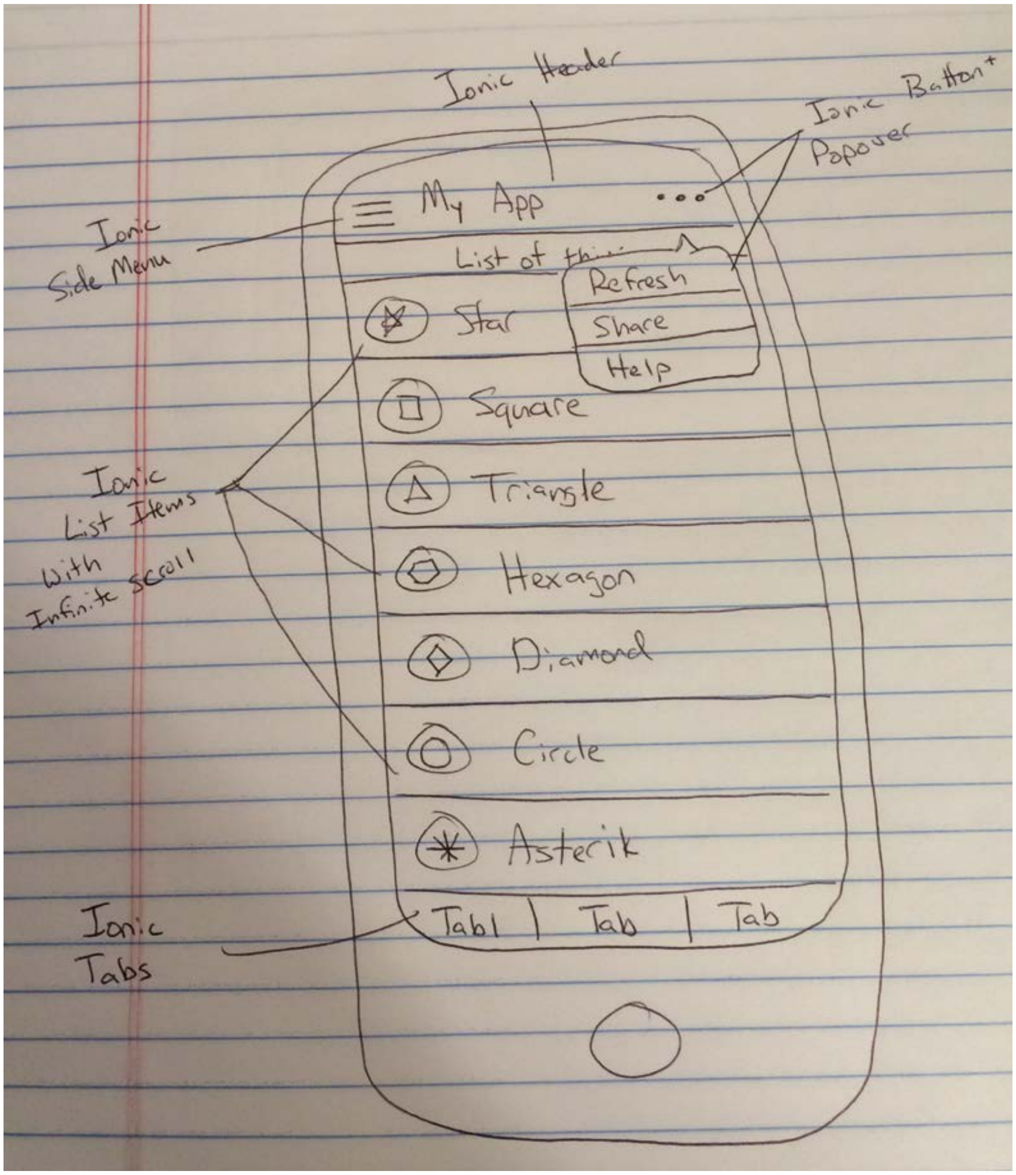


Diagram showing how parts of Ionic work together to create a useable interface

Ionic is an open source project that is primarily developed by Drifty. It was launched in November 2013 and has grown in popularity very quickly to become a primary choice when

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders. http://www.manning-sandbox.com/forum.jspa?forumID=931

building hybrid apps. Over 20,000 apps are launched with Ionic a month. Ionic is provided under the MIT license.

Ionic also has built a command line tool that provides some helpful developer tools. I'll refer to it as the Ionic CLI tool. This CLI tool can help generate starter projects, preview, build, and deploy your app. I'll demonstrate most of the features of the CLI tool as we go through our examples.

Ionic also has an included font icon library. This gives you access to a decent number of useful and common icons for your application. It is optional, but is provided by default and we will use it regularly in our examples.

Ionic also has a number of services in development that can aid in the development of apps such as a visual drag and drop app creator and deployment tooling.

These user interface controls are the primary feature of Ionic, but the Ionic team has worked hard to ensure its tools and processes work well with the following two projects, AngularJS and Cordova.

### 1.3.2 AngularJS:

AngularJS (often called Angular) is a Google open source project that has become quite popular with web application developers. It provides web developers the ability to write complete applications quickly and provides good structure for your applications. In our weather app example, Angular is used to help manage the user's data and loading information from our weather service.

Gone are the days where you have to use a server-based language (like PHP, Ruby, or Java) to build complex applications, today JavaScript web application frameworks like AngularJS allow you to build complex applications in the browser. This is an obvious advantage for hybrid app developers, since the browser is the platform we use to create our apps. If you are familiar with AngularJS (or other JavaScript application frameworks such as Ember) you will be able to apply your knowledge easily into mobile apps with Ionic.

Miško Hevery and Adam Abrons started AngularJS in 2009. Eventually Hevery joined Google and brought Angular with him. The project is immensely popular with developers today, and has been adopted by a number of large sites such as www.stackoverflow.com and www.nasa.gov. AngularJS is licensed under the MIT license.

In this book I will also use additional Angular modules that have been developed by third party developers. The primary example is a module called `ui.bootstrap`, which is an open source Angular module that provides better application routing and navigation compared to the default Angular routing module.

### 1.3.3 Cordova: Hybrid App Framework

In this book, we will be using Apache Cordova as our hybrid app framework. This is the layer that takes care of managing the communication between the browser window and the native APIs. Our weather app example needs access to the device's GPS information to know what

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

location to load data for, and Cordova is able to bridge the gap between Angular and the device to retrieve that information.

The core of Cordova provides a lot features, and it also provides a plugin system for developers to create new features such as native API integrations with the phone camera. It is actively maintained and regularly releases new version with improvements and new features. You can find out more about Cordova at http://cordova.apache.org. Later in chapter 11 I'll cover more detail about Cordova and plugins.

You may also have heard of Phonegap. Adobe contributed Phonegap to the Apache Software Foundation under the name Cordova. Today, Phonegap is a distribution of Cordova, or in other words Phonegap is essentially Cordova with support for a few additional commercial features from Adobe. For the purposes of this book we will use Cordova, but you could use Phonegap and its commercial features if you desire.

Cordova is an open source Apache project that has a large community around it. Adobe continues to be a major developer of the framework. Cordova is licensed under Apache 2.0 license.

## 1.4 Why Ionic?

Hybrid app frameworks are not new, there are a number of options available, but Ionic brings a new and important set of improvements. Until recently, mobile devices were still relatively sluggish and only a native app could deliver the performance and experience many developers wanted or needed. Mobile platform makers had not made browsers as fast as the native platforms. All of that has changed as devices have become more powerful, platforms have improved, and new tools like Ionic make it possible to build amazing hybrid apps.

### 1.4.1 Why Ionic is good for developers

Ionic is able to provide an experience built into the hybrid app that looks, feels, and performs like a native app. The long-standing argument that native apps are the only way to get fast and richly featured apps has been proven wrong. People expect their mobile apps to be fast, smooth, and intuitive, which Ionic apps can provide.

- **Build apps with the web platform.** Using HTML, CSS, and JavaScript, you can make hybrid apps that behave like native like mobile apps.
- **Built with AngularJS.** For people familiar with AngularJS (or even another JavaScript framework like Ember), Ionic is a perfect choice. Ionic is built alongside Angular, which allows you access to all of Angular's features as well as any of the many thousands of Angular modules for additional features. Ionic also uses the popular `ui.router` module for navigation, because Angular's default routing module lacks some features.
- **Uses modern techniques.** Ionic was designed to work with modern CSS3 features, like animations. Mobile browsers generally have better support for the latest web platform specifications, which allows you to use those features as well.
- **Powerful CLI tools.** With the Ionic command line tool, you can quickly manage development tasks such as previewing the app in a browser, emulating the app, or

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

deploying an app to a connected device. It helps with setting up and starting a project as well.

- **Ionic ecosystem.** Ionic also provides a rich ecosystem of features that make development much easier. The Ionic Creator service allows you to use a drag and drop interface to design and export an app. An upcoming service for remotely building and deploying apps is also in development. In short, Ionic is all about creating not just the basic tools for making hybrid apps but also the development tools that will help you create them efficiently.

You could even consider just using the parts of Ionic that you want, such as the UI components for a mobile app but use a different hybrid app build tool. While Ionic has a lot of features that work well together, you can select the parts you want to use and apply them to an existing project or workflow.

### *1.4.2 Why Ionic is good for managers/owners*

When I began to use Ionic I had to make the argument about why it was the right choice over other options. In a short presentation I made the following case for Ionic.

- **Ionic covers all the bases.** Ionic was able to provide a complete set of features needed to build a hybrid mobile app, with the current developer team skills available.
- **Ionic is open source.** Many developers are very fond of open source, but managers should also be interested in it. Ionic has permissive open source licensing that doesn't get in the way of our own copyrights.
- **Ionic has a dedicated team.** Open source projects can be difficult to select, because you can't be sure if it will be properly developed or supported. Ionic has a dedicated team that has a vested interest in keeping the platform on the leading edge.
- **Ionic development is fast.** Within minutes you can generate a base app and very shortly build a solid prototype. This is important to time management, costs, and maintainability.
- **Ionic is fun.** Making your developers happy is pretty important to their productivity. This is a bit more subjective statement, but the developers I know who have used Ionic enjoy it.

Selecting Ionic is a good move for managers who want to ensure they have a solid platform that is well supported and their developers can quickly leverage to build mobile apps.

### *1.4.3 Why Ionic is good for your customers*

Ionic provides developers the foundation and tools needed to build up great apps. Building on the benefits above, it allows hybrid apps to offer:

- **Native experience.** With Ionic, you can create a look and feel that is like the native apps, making it easier for your customers to use the app.
- **Performance.** The performance with Ionic is comparable to a native app, and the better

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

the app performs the happier customers will be.

- **Beautiful design.** Customers are accustomed to having apps that look great, and with Ionic as the base you can create an amazing experience.

With Ionic, you can design amazing apps that your customers will be delighted to use and take you far less time and effort to create.

## 1.5 Prerequisites for building apps with Ionic

In order to build hybrid apps, there are a few skills that you should already have which are not covered in this book. You don't need to be an expert in any of the following areas, but you should be prepared to use them all together.

### 1.5.1 Experience with the Web Platform

If you've built a website, you've used the Web Platform. The browser is like the operating system that we will be using to develop our mobile app. HTML, CSS, and JavaScript are the key languages the browser understands. HTML gives structure to the content, while CSS provides the design. JavaScript then provides for the interaction and logic necessary for our web application.

jQuery is often used in web development, but in this book we will be using raw JavaScript. You will need to be familiar with JavaScript syntax and concepts such as asynchronous calls and events.

### 1.5.2 Experience with web applications and AngularJS

You should have a fundamental understanding of web applications, since we will be building them inside of our mobile apps. There are a number of technologies and libraries that developers use to build web applications, and familiarity with the concepts will help you greatly.

In this book, our web applications will be written in JavaScript using the popular AngularJS framework. Ionic is built specifically to work with Angular, and developers who have experience building applications with Angular will be able to apply their experience easily. You might have experience with another framework, such as Ember or Backbone, which can provide a foundation as you learn the Angular specific approach.

I will cover a bit about Angular as we go since it is used heavily, but this is not a book about Angular. You'll want to refer to Angular In Action to learn everything you want about Angular beyond the scope of this book.

### 1.5.3 Access to a Mobile Device

Having a mobile device is extremely important when building a mobile app. I recommend that you have at least one device for every platform in order to test on an actual device. There are emulators that let you see what your apps should look like on a mobile device, but they are not full substitutes for the real thing.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

You will have to register these devices with your developer accounts as well, so it’s not practical to borrow. If you need a device, you can check for refurbished or used items online and use it just for development testing. The more types of devices you can test your app with, the better.

These three prerequisites will help you be more successful at designing, testing and building mobile apps across multiple platforms. Let’s take a look at the mobile platforms Ionic supports.

## 1.6 Supported Mobile Devices and Platforms

There are a number of mobile platforms: iOS, Android, Windows 8, Firefox OS, Tizen, Blackberry, and more. With Ionic, we can build for both iOS and Android. Support for Windows 8 and Firefox OS are planned for the future, but not currently available.

While it may be possible to develop an app by previewing only on a simulator, devices can act differently in the real world. When possible, it is recommended to have an actual device available for the platform to which you are building. It also helps to test apps on a device because you are able to experience how the app responds to touch gestures that are difficult to simulate. Let’s take a closer look at these two primary platforms and requirements.

### 1.6.1 Apple iOS

Apple makes the popular iPhone and iPad devices and they share a common platform called iOS. Apple has strong control over the entire experience from the devices to software to the apps, essentially making it a closed system. This has made iOS a strong platform from the perspective of users and developers.

Apple provides Xcode as the primary development program for iOS and Mac development. Xcode is free and available in the AppStore if you do not already have it downloaded. I will cover setting up for iOS development in the next chapter.

Xcode comes with a set of simulators, which allow you to simulate different versions of their iPhones and iPads. The simulators are fairly good at giving a realistic experience, which is helpful when targeting multiple versions of iOS with the same app.

Apple has one major requirement for building iOS mobile apps: you will need a Mac computer. Apple has only designed its development tools to work on Apple’s operating system OSX, and it is also recommended to be running the latest version. Xcode is the program used to manage mobile app development, which is provided for free by Apple, but only runs on OSX.

For those of you who aren’t using a Mac, it is worth considering purchasing one if you plan to do iOS development. If you just need to build mobile apps, you’ll be able to take advantage of any of the Mac computers. Any new Mac will have enough processing power to manage the simulation and build process. If you consider purchasing a used machine, you should verify that it is able run the latest version of OSX.

If you don’t have a Mac, there are some options available that can help build your apps. (Possible Ionic tool here, need to do more research into options. Most are only able to provide a build, not simulators.)

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

The Apple Developer Program has two types of membership, iOS and Mac development. You will need to sign up at http://developer.apple.com and join the iOS program. It costs $99 USD per year, but you only need to sign up when you are ready to sign and deploy your app to the AppStore. You are able to work through this entire book without this account until the point I show you how to deploy an app into the AppStore.

### 1.6.2 Google Android

Google created Android as an open source mobile platform, and has allowed mobile device makers to take Android and integrate it into their devices. Compared to Apple's approach, Android has a very diverse set of devices. Older devices may also have Android forks specifically designed for a mobile carrier. This open system has encouraged adoption and also has been the leading platform in emerging markets, by allowing lower cost devices due to the absence of licensing fees for the operating system.

Android provides a number of tools for developing, which are freely available for download from Android's site. Google has also been working on additional tools that are being built into Chrome, Google's browser, providing useful development support for hybrid app developers. I'll cover how to setup your computer for Android development in the next chapter. Android has a simulator that can emulate the screen size and resolution of most Android devices.

Android development is supported on Mac, Linux, and Windows computers. You can review the exact requirements for Android developer tools https://developer.android.com/sdk/index.html.

Google also has a Developer Program, which is $25 USD per year. Just like with iOS, you don't have to sign up until you are ready to publish your app into the Play Store. You can register at https://play.google.com/apps/publish/signup/.

There are a few other Android app stores, notably the Amazon Web Store, which may also charge for a developer program. These are not covered in this book. However you will be able to build and deploy apps for any Android based device, even if the app is distributed through a different store.

## 1.7 Summary

Through this chapter we've looked at details about how Ionic provides a powerful set of tools for building hybrid apps. Let's review the major topics covered in this chapter.

- Ionic is a solid choice that benefits developers, managers, and users.
- Hybrid apps are an advantage for developers who already are familiar with the web platform, and do not require learning additional programming languages.
- Hybrid apps use a web view inside of a native app to run web applications that have access to the native APIs.
- Ionic is designed to work with AngularJS for web application development and Cordova for integration with the device platform.
- Android and iOS are supported and both require developer subscriptions. iOS can only

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931

be developed from a Mac.

©Manning Publications Co. We welcome reader comments about anything in the manuscript - other than typos and other simple mistakes. These will be cleaned up during production of the book by copyeditors and proofreaders.
http://www.manning-sandbox.com/forum.jspa?forumID=931