SAMPLE CHAPTER

Updated for Swift 3

Anyone Can Create an App

Beginning iPhone and iPad programming

Wendy L. Wise

manning

www.itbook.store/books/9781617292651

Anyone Can Create an App

by Wendy L. Wise

Chapter 2

Copyright 2017 Manning Publications

www.itbook.store/books/9781617292651

brief contents

www.itbook.store/books/9781617292651

www.itbook.store/books/9781617292651

2 Building your first app

This chapter covers

- Launching Xcode for the first time
- Creating your first application
- Running your first application

You've done all the setup (if you haven't done it yet, please go to appendix A to set up the tools you'll need to complete this chapter). Now you're probably eager to start creating your first app. We're going to start off as almost every programmer ever has: with a Hello World application—a rite of passage for coding. A Hello World application is the first app almost every programmer writes when learning a new programming language, because it's a simple app that accomplishes two goals:

- It gets the programmer using the new tools.
- It proves that the programming tools are set up correctly.

Those are the goals of the application, but what does it actually do? Hello World will launch and display the words "Hello World!" That's it. It may not seem like much, but it's important to take small steps while learning to program.

I'm going to write my own version of pseudocode here so you know what you want to accomplish. As I explained in chapter 1, *pseudocode* is your road map—logical

www.itbook.store/books/9781617292651

steps required to make the app do what you want it to do. This pseudocode will be a bit more detailed than future chapters, because it's your first app. In my version of the pseudocode, the steps needed to build the Hello World app are as follows:

1. Launch Xcode.
2. Create a new project.
3. Set up your project options.
4. Run the blank app.
5. Add the Hello World text.
6. Run the app.
7. Pat yourself on the back.

It's going to take seven small steps to create your first app. Get ready—we're starting!

2.1 Launching Xcode for the first time

I'm going to quickly walk through Xcode, the integrated development environment (IDE) introduced in chapter 1. I'll show you only what you need to know to create your first app in this chapter. Chapter 4 goes into detail about Xcode if you must jump ahead.

2.1.1 Step 1: Launch Xcode

Step 1 in the pseudocode was to launch Xcode, so find the Xcode icon in Launchpad on your Mac. If you're not familiar with Launchpad, it is the easiest way to find applications installed on your Mac. It looks like a rocket launching (fancy that…Launchpad, a rocket…launching), as shown in figure 2.1. Once Launchpad is on your screen, look for the Xcode icon and click it.

Helpful shortcut alert

You can also find Xcode by holding down the Command key (⌘) and the spacebar key to launch Spotlight Search. Type *xcode* into the Search bar. Spotlight will help you find anything on your Mac.

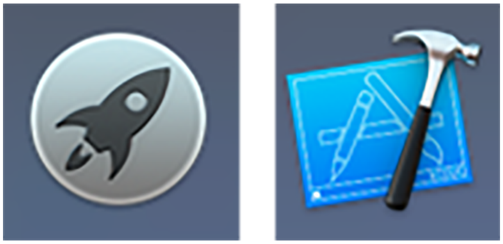
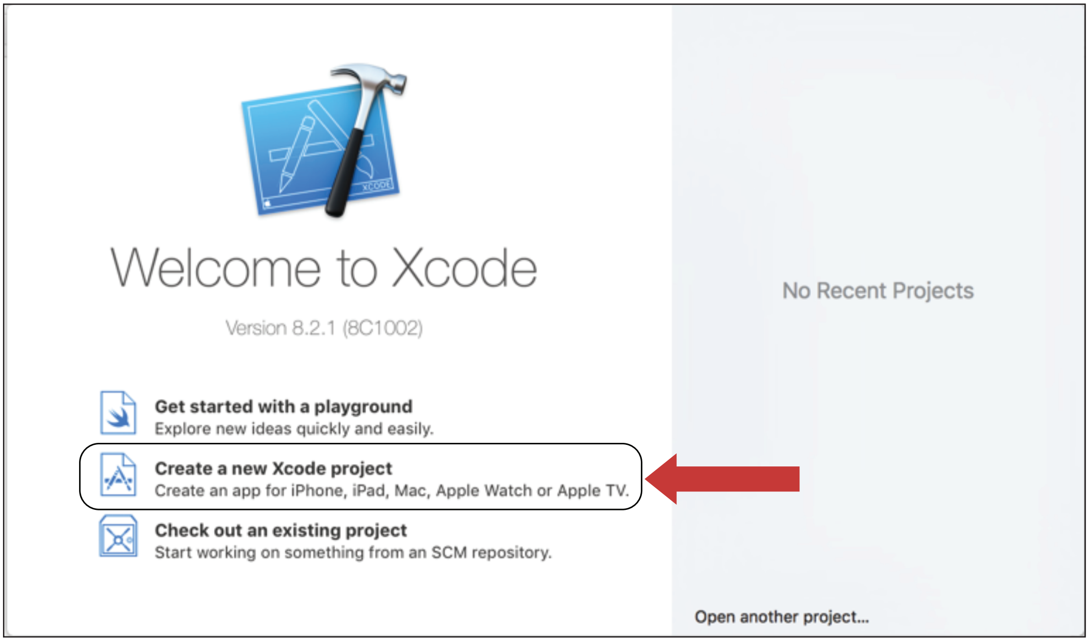


Figure 2.1 Use Launchpad (left) to launch Xcode (right).

www.itbook.store/books/9781617292651

2.1.2 Step 2: Create a new project

The next step in the pseudocode is to create a new project. Once you've launched Xcode, you should see an initial window that says Welcome to Xcode, as shown in figure 2.2.



Welcome to Xcode

Version 8.2.1 (8C1002)

Get started with a playground
Explore new ideas quickly and easily.

Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.

Check out an existing project
Start working on something from an SCM repository.

No Recent Projects

Open another project...

Figure 2.2 Click Create a New Xcode Project to start your first project.

When this appears, click to select the second option, Create a New Xcode Project. The Choose a Template dialog box appears, as shown in figure 2.3.

We'll walk through each option a little later. For now, click Application in the left window pane, if it isn't already selected. On the right, you now have multiple options. Click Single View Application, and then click Next. The Choose Options dialog box will open.

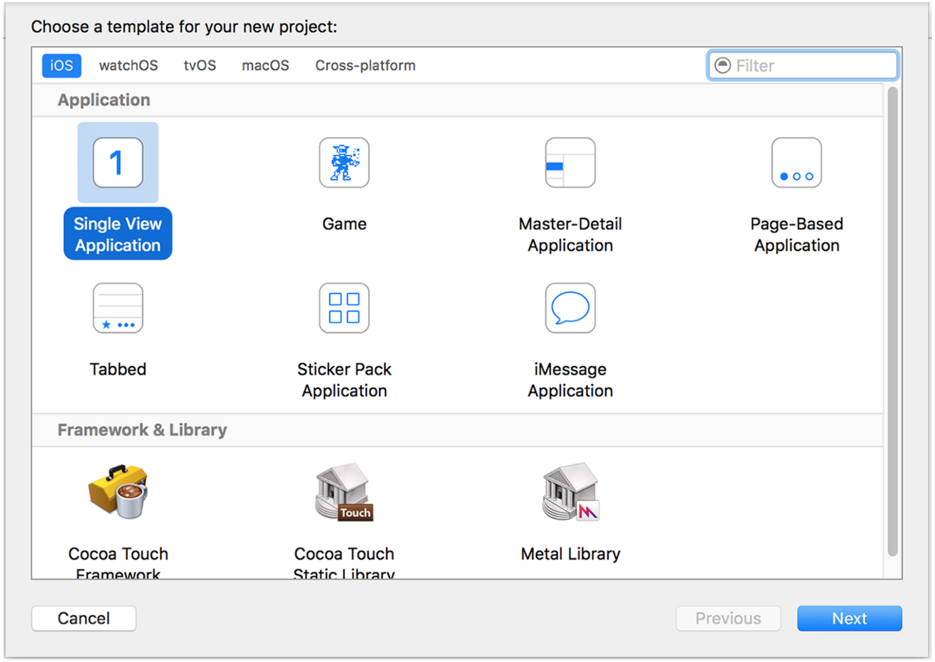
2.1.3 Step 3: Set up your project options

The Choose Options dialog box is used to set up the basic information for your new app, as shown in figure 2.4. You'll see these options for each new app you create, so let's go over them so you know what they do.

Product Name is where you enter the name of your new application. In this case, we're calling it HelloWorld. This product name will be used by Xcode to fill in several areas of the application (I show you those areas in later chapters).

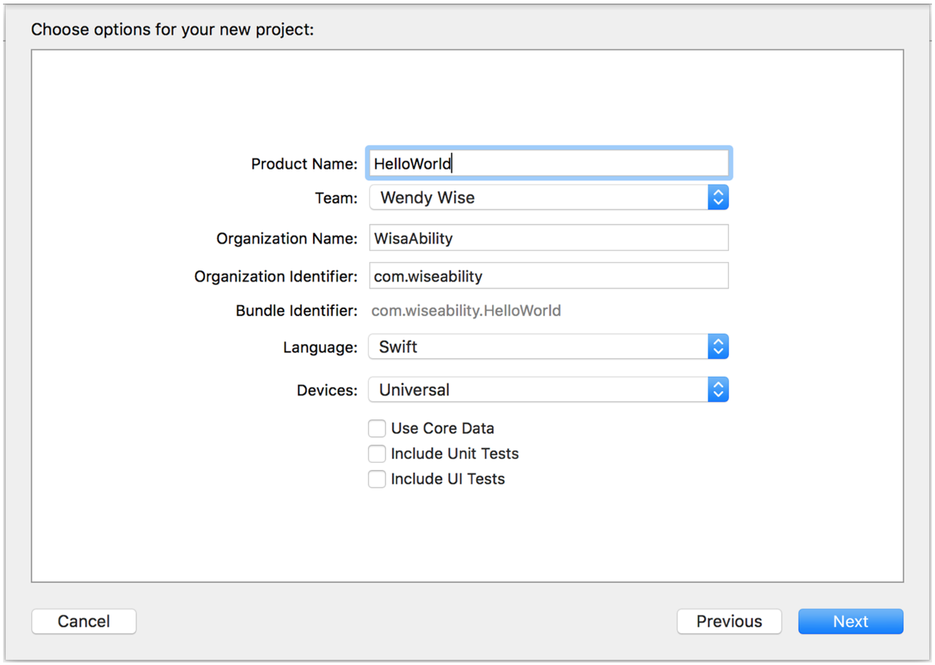
The next box is Organization Name. This field assumes that you are developing for a company or organization. I do have my own company, so I entered *WiseAbility* in this

www.itbook.store/books/9781617292651



Choose a template for your new project:

Figure 2.3 Click Single View Application in the iOS Application section.



Choose options for your new project:

Product Name:	HelloWorld
Team:	Wendy Wise
Organization Name:	WisaAbility
Organization Identifier:	com.wiseability
Bundle Identifier:	com.wiseability.HelloWorld
Language:	Swift
Devices:	Universal

□ Use Core Data
□ Include Unit Tests
□ Include UI Tests

Figure 2.4 Configuring your project options—add HelloWorld to the Product Name, add an Organization Name, and click Next.

field. You don't actually need to be affiliated with a company to develop for iPhones and iPads, so if you don't have a company or aren't affiliated with one, put something here as a placeholder. You can enter a fictional dream company name if you want, or your own name, or leave it blank.

The Organizational Identifier field is used to create unique names for your projects in the event that you want to publish your app to the App Store. It's common practice to use your website domain in reverse for this field. My website is www.wiseability.com, so I use *com.wiseability* for the organizational identifier. You must fill in this field, and if you don't have a website right now, you can use *com.yourlastname* (filling in your last name). You can't have any spaces in this field, so if your last name has spaces, use dashes (-) to replace the spaces.

Once you fill in the Organization Identifier, the Bundle Identifier should show *[your-organization-identifier].HelloWorld*, as mine does: *com.wiseability.HelloWorld*. It isn't important to know the details of these settings right now—we still have a lot of the elephant left to eat that I mentioned in chapter 1.

> **NOTE** If you name your project Hello World, the Bundle Identifier will show *Hello-World*—what gives? Well, Xcode doesn't allow spaces in the Bundle Identifier field, so if you enter spaces in the product name, it replaces them with dashes (-).

Finally, make sure the Language selection is Swift, and Devices is set to iPhone. Leave the Use Core Data selection blank, and make sure the Include Unit Tests and Include UI Tests options are unchecked. Click Next.

You can now choose where to save your project files on your hard drive. I recommend creating a folder called dev and putting all your work there, so it's easy to keep track of.

Click Create. Xcode will now create a new project for you and open it. This is pretty easy so far, right?

2.1.4 Step 4: Run the blank app

I know it doesn't feel like you've actually built an application yet, but Xcode has taken care of the infrastructure and framework for you. How cool is that?

Your next step is to run the blank app. At top left of the current screen is the Run button. It looks like a familiar, triangular play button, as shown in figure 2.5.

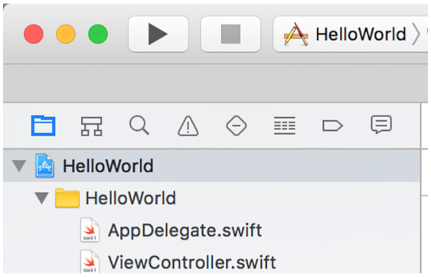


Figure 2.5 Xcode's Run button looks like the play button on most electronics.

This is the universal coding sign for Run. Click it once. The iOS Simulator should pop up with a blank screen, as shown in figure 2.6.

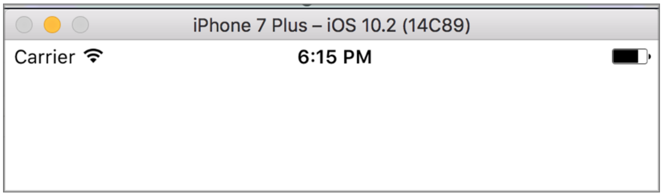


Figure 2.6 If you see a blank simulated iPhone screen, you've set up your project correctly.

The iOS Simulator is an application that Apple built to help developers (like you) test their applications before installing them on an actual iPhone or iPad. Chapter 3 talks a lot more about the iOS Simulator. For now, you're going to launch it so you can see it in action. If you see the blank screen, you're golden.

2.1.5 Step 5: Add the Hello World text

Step 5 of the pseudocode was to add the Hello World text, so let's get to it. Back in Xcode, on the left side of the screen, you should see several filenames, including Main.storyboard.

Click the Main.storyboard file. You should see a screen like figure 2.7.

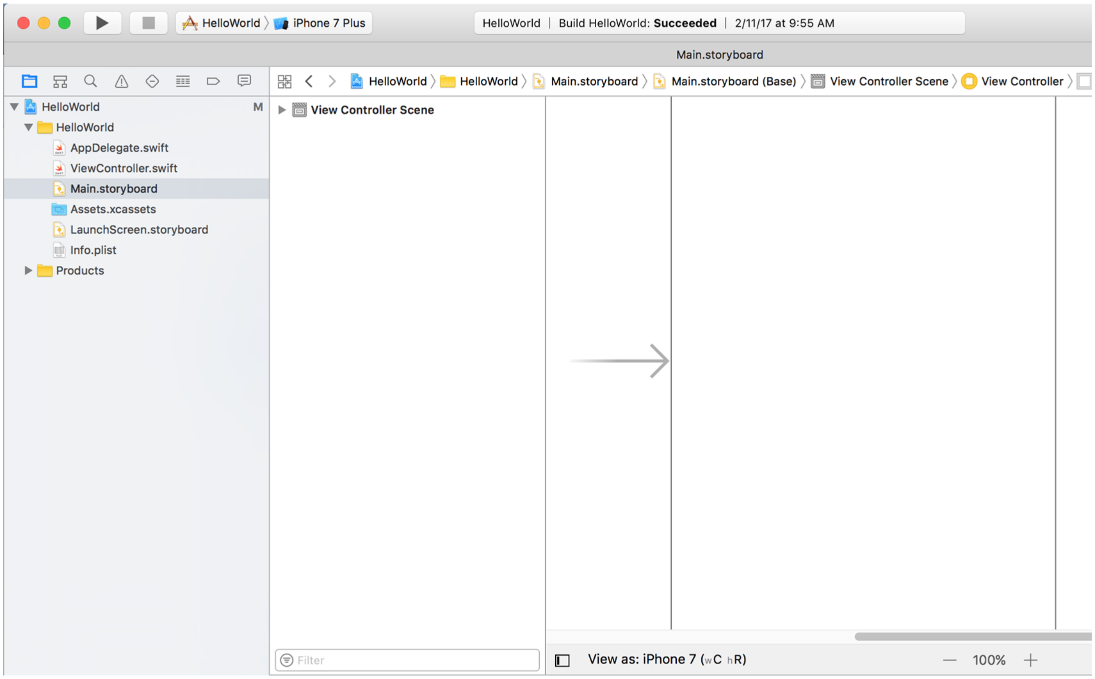
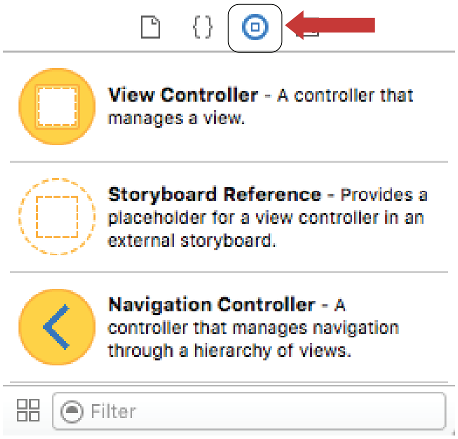


Figure 2.7 Select the Main.storyboard file on the left side of Xcode.

www.itbook.store/books/9781617292651

At bottom right on your Xcode screen, if it isn't already selected, select the third button—the one that looks like a circle with a tiny square in it—as shown in figure 2.8. This is the Object Library, which is like a toolbox with all the things that you'll need to build your app. The toolbox is full of items like nuts, bolts, screws, wires, and more. Chapter 4 talks a lot more about the Object Library, but for now, you're going to select a *label* as the item you need for this app.



View Controller - A controller that manages a view.

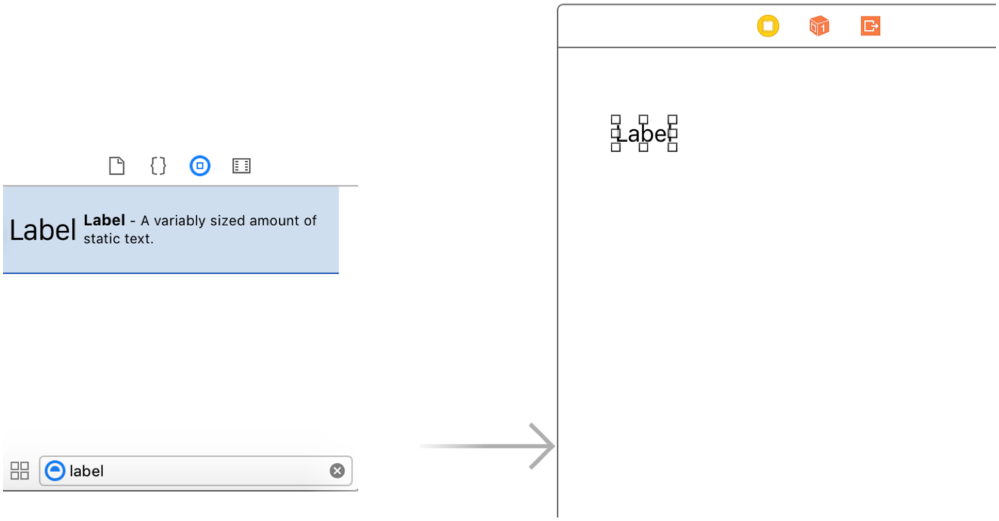
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Figure 2.8 Showing the Object Library

At the bottom of the Object Library window, click in the search area, and type *label*. The list should be filtered down to the word *Label* and an icon.

Click and drag the Label icon (the bigger word Label on the left) to the other screen, called the *storyboard* screen, as shown in figure 2.9. Position it near the top-left corner in the storyboard screen, and then drop it by releasing the mouse button.



Label **Label** - A variably sized amount of static text.

Figure 2.9 Dragging a label onto the storyboard

www.itbook.store/books/9781617292651

Once your label is on your storyboard, double-click it so you can edit the text. Type *Hello World!* and then click somewhere else on the storyboard to stop editing the label.

2.1.6 Step 6: Run the app

We're finally at step 6 of the pseudocode list: Run the app. You already ran the app once earlier in step 4, but now you'll run it again to see the actual Hello World text.

Click the Run button again. If everything is set up correctly, you should see the iOS Simulator window pop up with the words "Hello World!" displayed, as shown in figure 2.10. The size of the storyboard and the size of the Simulator are different, but you'll fix that in Chapter 7.

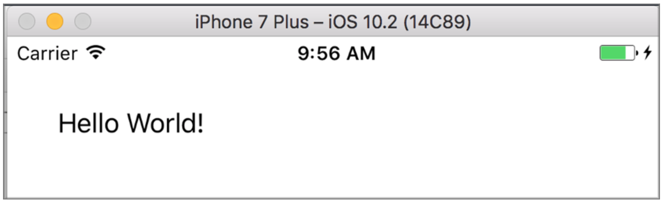


Figure 2.10 Your first app running—congratulations!

> **Troubleshooting**
>
> If you don't see your label or your app doesn't look like figure 2.10, go back to your storyboard in Xcode, move your label closer to the top left of the storyboard screen, and run the app again.

Congratulations! You wrote your first iOS application. Exciting, right? You're already a success! Revel in it.

> **Terminology alert**
>
> You are now a *newbie*, or *noob*, to iOS programming. That means you are a new, inexperienced developer. Don't let anyone tell you it's a negative term—you're learning. Wear your newbie badge with pride.

2.1.7 Step 7: Pat yourself on the back (and review)

Step 7 in the pseudocode steps was to pat yourself on the back. Go for it!

This example was pretty easy, and it's true that we glossed over a lot of stuff. The next chapter will explain the details of what you did and why you did it. Remember, it's important for you to understand the underlying concepts of development, but the details can drown you if you don't wade in slowly. At this point, I want you to focus on

www.itbook.store/books/9781617292651

remembering some of the basics: how to create an app, where the Object Library is, and how to build and run (with the play button) your code.

Concepts to remember

- How to launch Xcode
- How to start a new project—you're going to do this a lot
- How to run an application—you're going to do this a lot, too
- How to drag and drop objects to your storyboard

If you don't remember how to do all of these things, walk through this chapter again. You'll do these things many, many times in the upcoming chapters, so you'll have lots of practice. It's good to get the basics down early.

I hope you feel like you accomplished something, because you did. You wrote your first iOS application. It may not have done much, but still—you did it. You may also feel like there's a *lot* you don't know, and that Xcode looks scary with all those buttons, tabs, and icons. Don't worry. I'm going to cover the details of it so it won't look so daunting. We'll do it in small chunks, though, and we'll do it on a *just-in-time* basis so you don't feel like you're drowning.

2.2 Summary

I told you in chapter 1 that I'd distinguish key programming concepts in each chapter. This chapter was mostly a how-to—remember, I also told you that you'd learn what you need when you need it. This chapter taught you the actions to take, and the next chapter will tell you why you took them. I hope you're excited about your progress and curious about what you've done so far.

www.itbook.store/books/9781617292651

MOBILE DEVELOPMENT

Anyone Can Create an App

Beginning iPhone and iPad programming

Wendy L. Wise

Free eBook
SEE INSERT

Do you have a fantastic idea for an iPhone app but no idea how to bring it to life? Great news! With the right tools and a little practice, anyone can create an app. This book will get you started, even if you've never written a line of computer code.

Anyone Can Create an App begins with the basics by introducing programming concepts, the Swift language, and the tools you'll need to write iOS apps. As you explore the interesting examples, illuminating illustrations, and crystal-clear step-by-step instructions, you'll learn to

- Get started programming, no experience necessary!
- Add controls like text boxes and buttons
- Keep track of your favorite things by creating the Like It or Not (LioN) app

By the end, you'll be able to create and run your own apps, and you'll have the confidence to learn more on your own.

This book is written especially for nonprogrammers —no experience needed!

Wendy Wise is a web and mobile developer with a talent for teaching app development to new programmers.

Updated for Swift 3

"An excellent way to get started on iOS development. A beginner-friendly approach that lets you learn at your own pace."

—Stephen Byrne, Dell

"A clear, concise guide for the uninitiated. Quick results guaranteed in simple steps!"

—Mark Cooper, Cooper and Lansbury Associates

"Highly recommended for anyone wanting to get their first app into the App Store."

—Jason Pike
Atlas RFID Solutions

"Approachable, easy to read, and easy to follow."

—Jocelyn Jeriah
MBO Partners

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/books/anyone-can-create-an-app

ISBN-13: 978-1-61729-265-1
ISBN-10: 1-61729-265-6
52999
9 781617 292651

manning

US $29.99 / Can $39.99
[including eBook]

www.itbook.store/books/9781617292651