

Building Mobile Applications

Creating iOS applications with jQuery Mobile, PhoneGap, and Drupal 7



Jeff Linwood

1st Chapter, Early Release

Introduction	3
Prerequisites	3
Introduction to Mobile Apps.....	4
Getting Started with PhoneGap.....	6
Prerequisites and XCode 4.....	6
Installing PhoneGap	6
Building Your First PhoneGap App for iOS	7
More to come!	15

Introduction

I'm excited about Drupal and mobile, and I hope you are too! There's a lot you can do with your existing Drupal site to bring it into the mobile world, including putting a mobile theme on it, or making your main theme responsive. In this book, I'm going to discuss another option, which is building a stand-alone mobile app for the iPhone with jQuery Mobile and PhoneGap that communicates with your Drupal server using the Services 3 module.

PREREQUISITES

I'm not going to assume that you know anything about mobile apps, or that you know anything about Drupal's Services 3 module, REST, jQuery Mobile, XCode 4 or PhoneGap.

You should have administrative access to a Drupal 7 site (either running on a server, or running locally), and know how to install modules onto your Drupal site. I'm going to assume that your Drupal site is already set up - if you haven't set up Drupal yet, there are many good books, and there is an installation guide on Drupal.org (<http://drupal.org/documentation/install>). This book covers Drupal 7, not Drupal 6, although much of the discussion is the same. Unfortunately, the data formats used for expressing nodes in JSON changed between Drupal 6 and 7, so you can't use exactly the same code.

You will need to download and install XCode 4 from Apple, which basically means that you have to have a recent Mac. You can get XCode 4 from the Mac App Store, or from <http://developer.apple.com>.

Introduction to Mobile Apps

What are mobile apps? What makes them different from a web site? These are questions I get asked frequently when I speak about building mobile apps.

There are three different kinds of mobile apps:

		
Native	HTML5	Hybrid
Android iPhone, iPad	Mobile Framework	PhoneGap Titanium

Native mobile applications will only work on one platform - Android, iOS, BlackBerry, Symbian, or Windows Phone. When you develop a native app for Android, you most likely will be using the Java programming language with the Android SDK. You can use Windows, Mac, or Linux to develop for Android, which is convenient. Writing an app for the iPhone or iPad requires a Mac and Apple's XCode 4 IDE. You need to learn the Objective-C language to write a native app for the iPhone or iPad.

HTML5 Mobile Applications work on at least one platform, but if written well, should work on more than one. Often, it's not the type of smartphone that is the problem, but the underlying browser technology it uses. For instance, both Android and iOS use WebKit for the browser, so

an HTML5 mobile app should work almost identically on both platforms. Windows Phone uses a version of Internet Explorer, and you'll need to keep that in mind if you target Windows Phone as a platform. BlackBerry has moved to WebKit in recent versions of its operating system, but earlier BlackBerry devices use a much more primitive web browser.

Typically, you'd use a mobile toolkit such as Sencha Touch, Tora Mobile, Jo, or jQuery Mobile to make your app work smoothly across browsers with an app-like feel. Each of these is evolving quickly - in this book, I'm going to discuss jQuery Mobile, but much of what I talk about applies to the others as well.

Combining HTML5 and native application development, you get into Hybrid applications. The Facebook mobile application is a great example of this - you can download the Facebook app from the Apple App Store or the Android Market, but the app itself is built with a combination of native functionality and HTML. Facebook did this to simplify cross-platform development, and to have a faster development cycle.

PhoneGap (<http://www.phonegap.com/>) is the leading open source toolkit to embed HTML5 mobile applications in a native iOS, Android, BlackBerry, Windows Phone, or WebOS app. A PhoneGap app is based around the native app's web browser control. The HTML5 mobile app runs inside the web browser control, but it can leverage more functionality than an app running in the mobile browser by using the standard PhoneGap API or by using PhoneGap plugins.

In 2011, Adobe bought Nitobi, the makers of PhoneGap, so there may be some changes to PhoneGap's product and service offerings.

Getting Started with PhoneGap

The first step we're going to take is the classic, Hello World - but as a mobile app for iOS! Using PhoneGap, this is actually very straight forward. PhoneGap lets us wrap HTML web pages in a native app wrapper, and it provides access to sensors on the device such as the camera, the accelerometer, and the current location of the device.

PREREQUISITES AND XCODE 4

You will need a recent Mac, capable of running Apple's XCode 4 IDE (integrated development environment). As of the writing of this book, the current version of XCode is 4.2, and you may download and install it from the Mac App Store.

If you want to run your app on your actual iPhone, iPod Touch, or iPad, you will need to purchase a \$99/year iOS developer subscription from Apple. This also allows you to view technical information on their developer web site, gives you early access to beta software, and allows you to publish your apps in the iOS App Store.

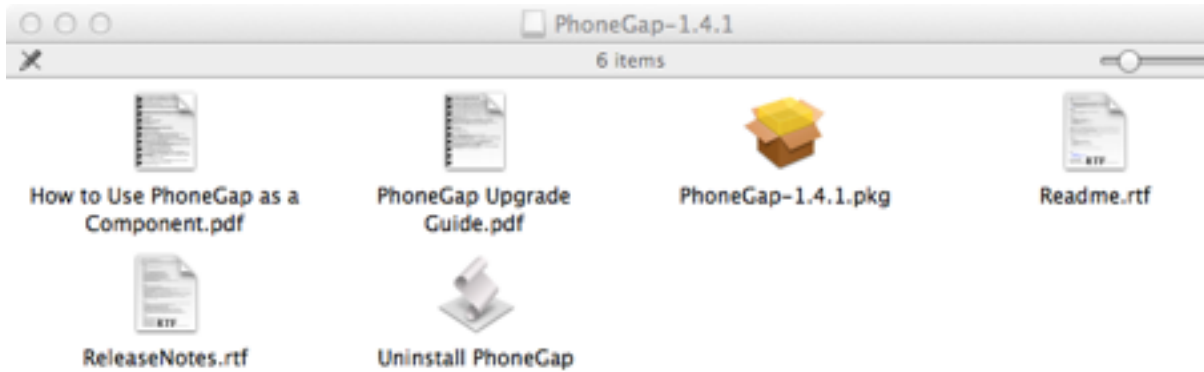
INSTALLING PHONEGAP

You can follow the directions for downloading and installing the latest version of PhoneGap from <http://phonegap.com/start/#ios-x4>

The first step is to download the PhoneGap distribution (at the time of writing, 1.4.1). The distribution contains libraries for all supported platforms:

- Android
- Bada
- BlackBerry
- iOS (iPhone, iPod Touch, and iPad)
- Symbian
- WebOS
- Windows Phone

We're only concerned with the iOS version of PhoneGap, which is in the lib directory under iOS. You'll see a DMG file. Double-click it to open it, and you'll find detailed instructions, an upgrade guide, a license, and an installer - the PKG file.



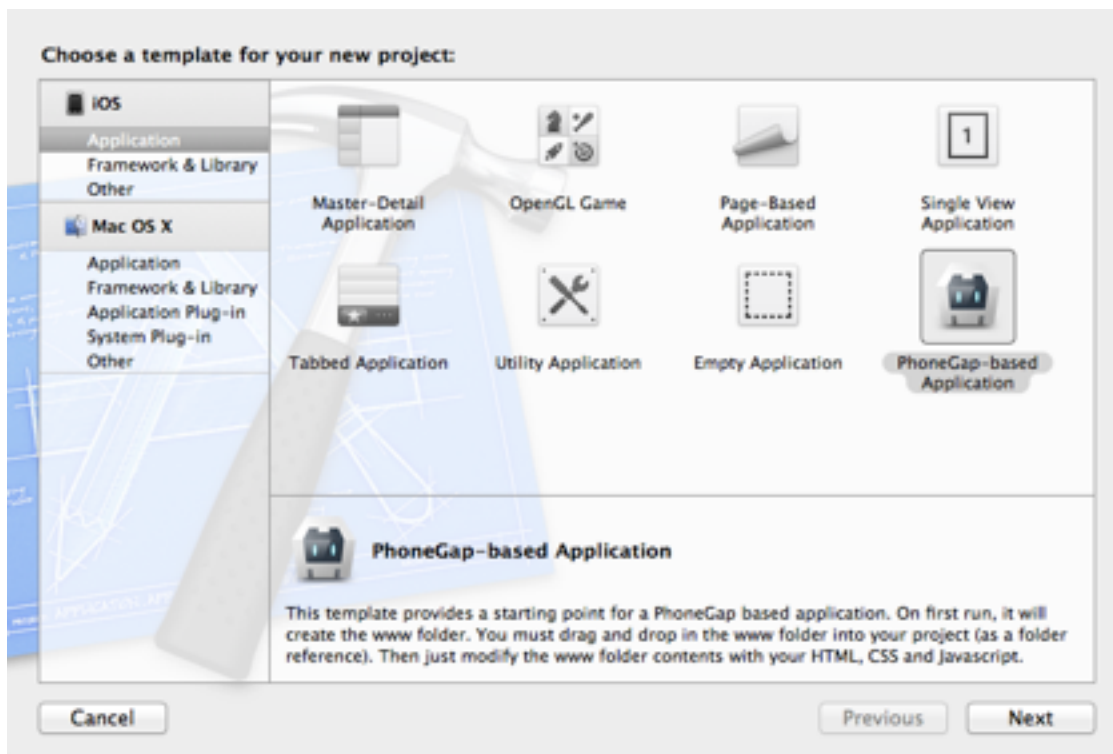
The installer will do its work and install PhoneGap so that you can use it from XCode 4.

BUILDING YOUR FIRST PHONEGAP APP FOR IOS

For this book, I'm using XCode 4.2 as the IDE for iOS, so I linked to the specific directions for that platform. I'm also going to target iOS 4.x and above, using the PhoneGap Media Capture API to access the camera.

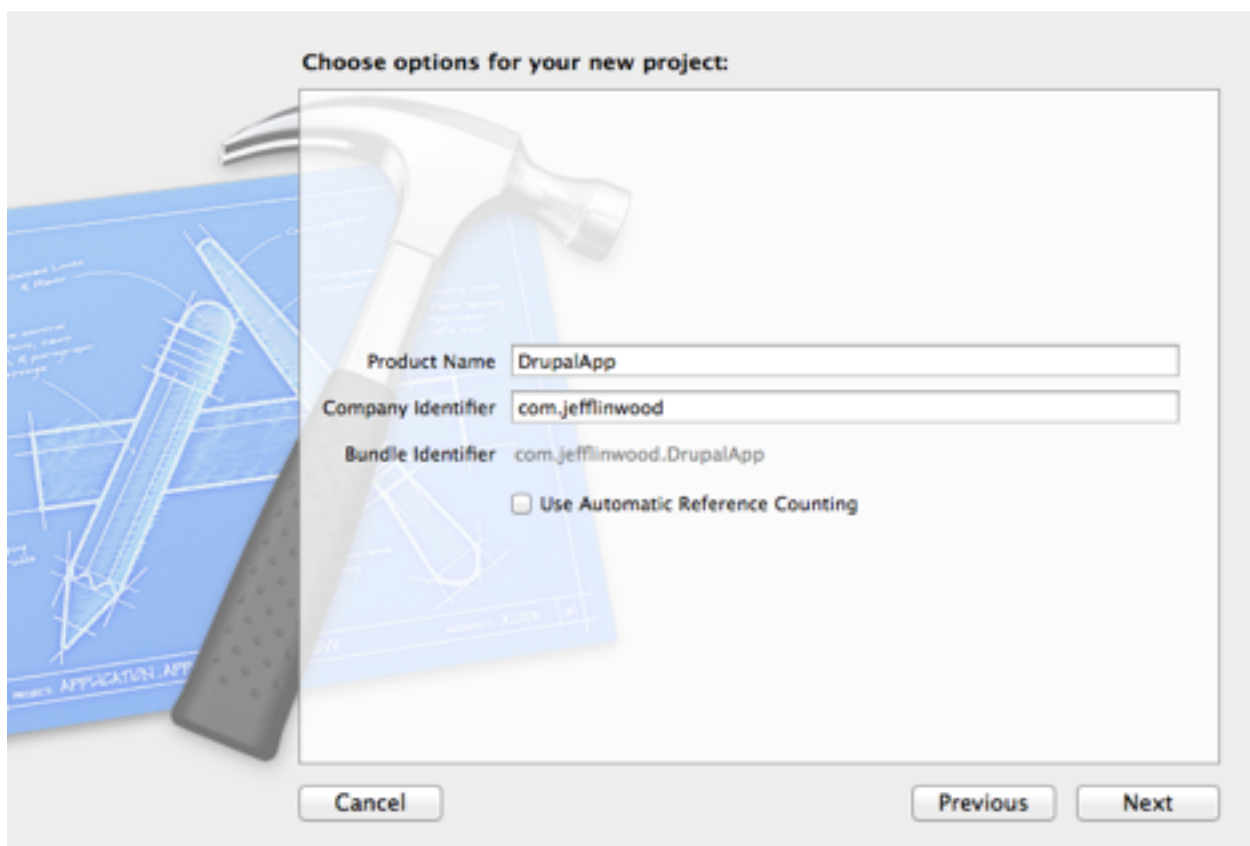
After running the PhoneGap installer, open XCode 4 and start a new project. If the startup screen doesn't appear asking you to either open an existing project or create a new project, use File->New->New Project... from the XCode 4 menu.

You should see the PhoneGap template as one of your options:



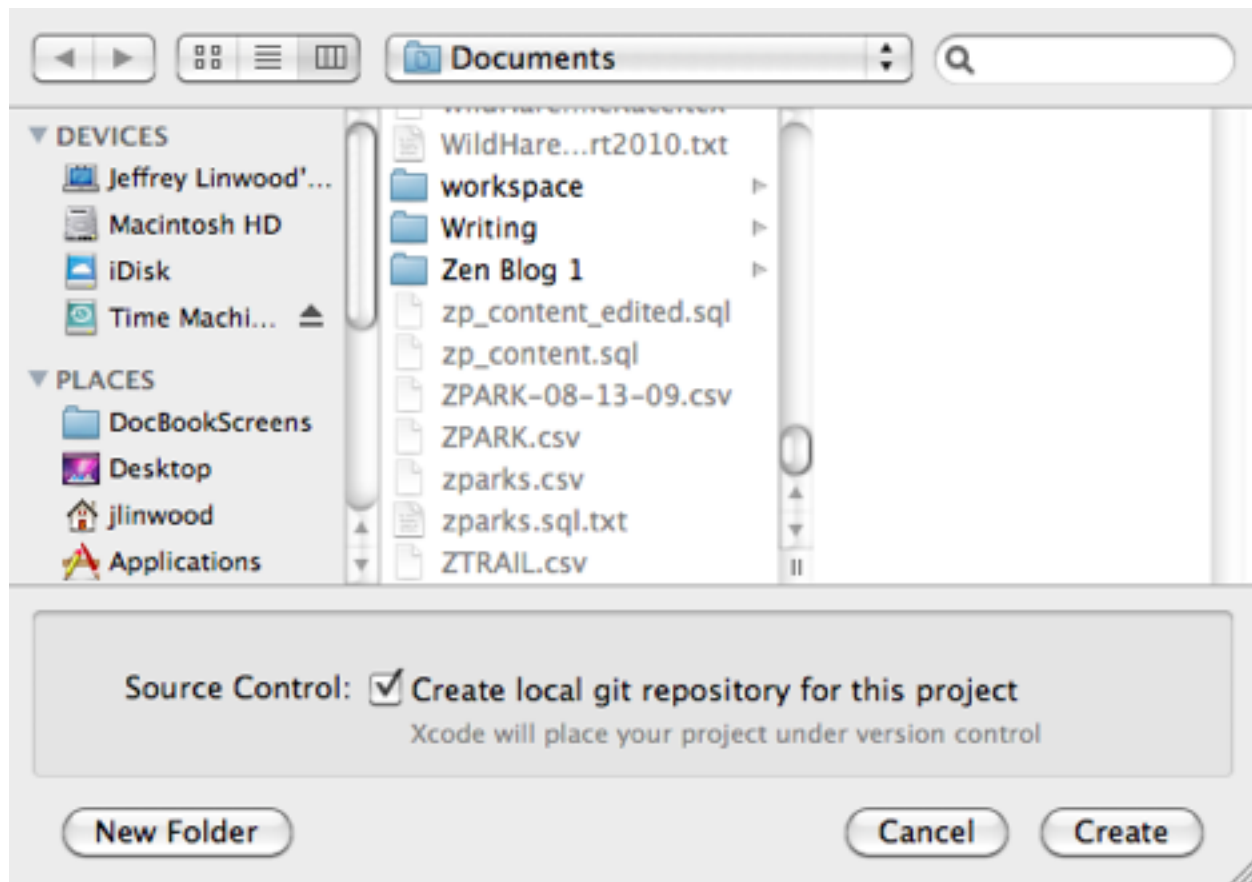
Click Next on the Choose a template dialog

I'm going to use `DrupalApp` as my Product Name, and `com.jefflinwood` as my company identifier. Uncheck Automatic Reference Counting - ARC is a feature that allows you to not have to write as much memory management code if you need to write native code. The downside is that if you use third-party plugins for PhoneGap, they have to use ARC, or you have to add a linker flag to your build settings for each source code file. Things may change in the future, but my Drupal Plugin for PhoneGap for iOS doesn't use ARC, and the third-party libraries it depends on don't either.



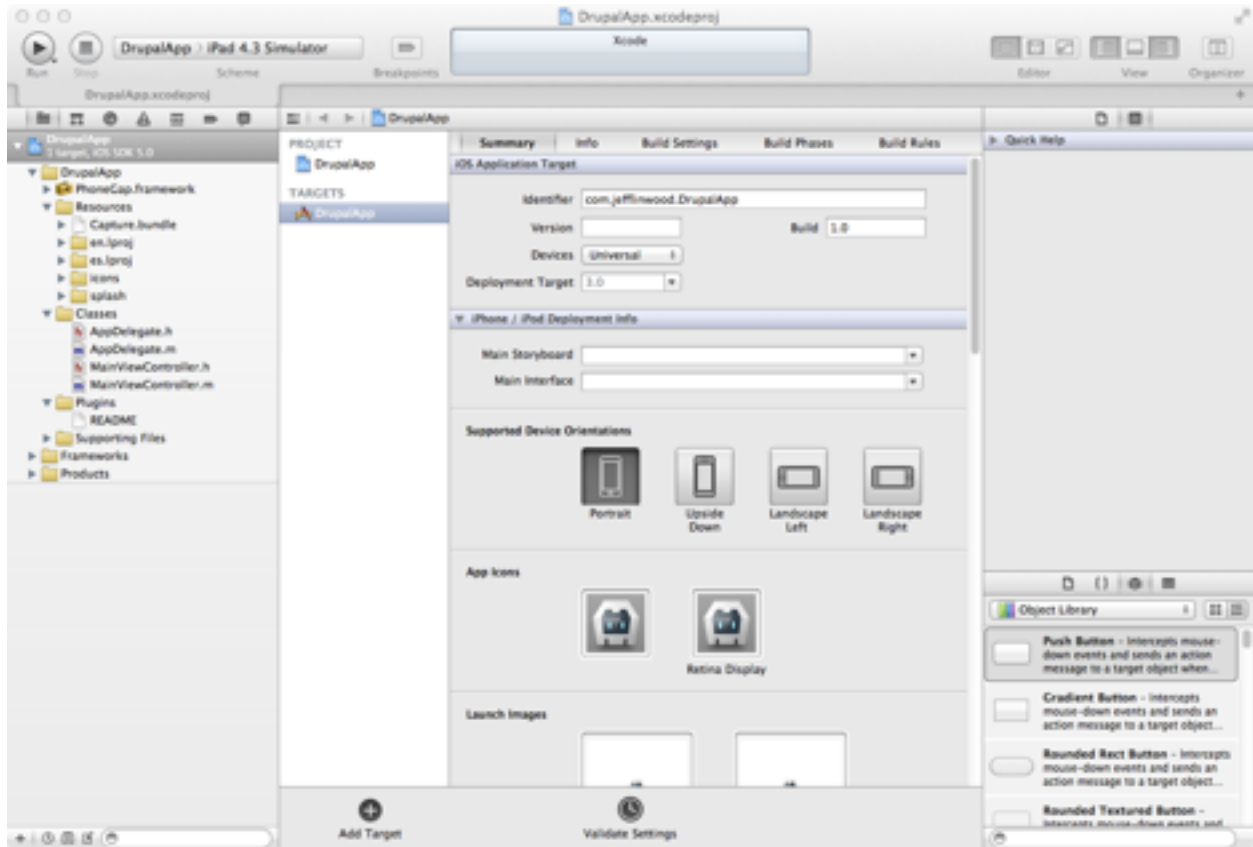
Click Next.

Now put your new project into a folder of your choice (for instance, under your home directory's Documents), and let XCode 4 create a local git repository if it offers to.



Click Create

XCode 4 will do some processing, and then you should see your new DrupalApp project in Xcode.

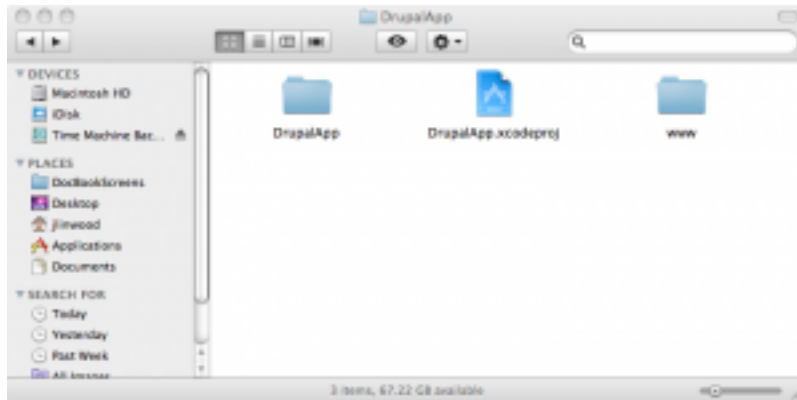


Hit the Run button in the upper left hand corner of XCode to run your project in the iPhone or iPad Simulator. PhoneGap will load, but it won't have any content to display. The first time you run your project, PhoneGap will actually create a www folder you can drag into XCode to get started. Here's what the iPhone Simulator will look like:

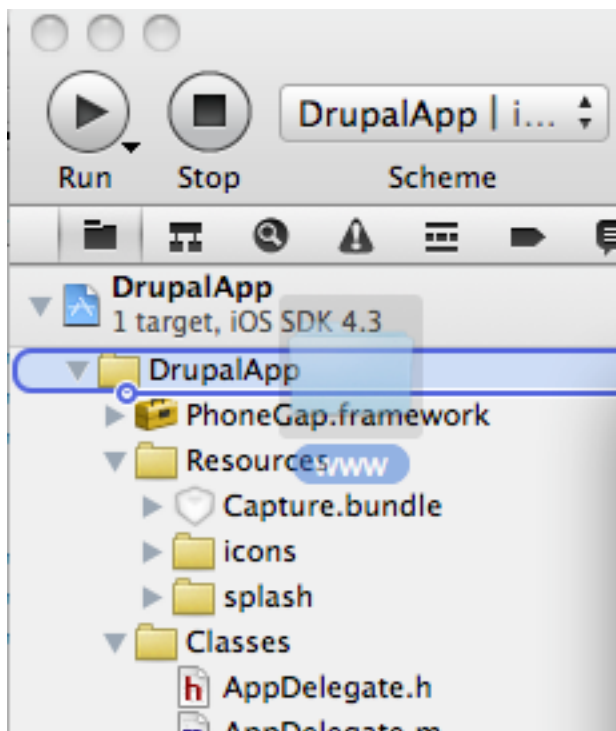


Now that we have our XCode project set up, and we've run our project, we need to add the www folder to our project. With the current version of PhoneGap, this needs to be done manually – the template isn't set up to import the www folder into the XCode project. Luckily, this is pretty easy.

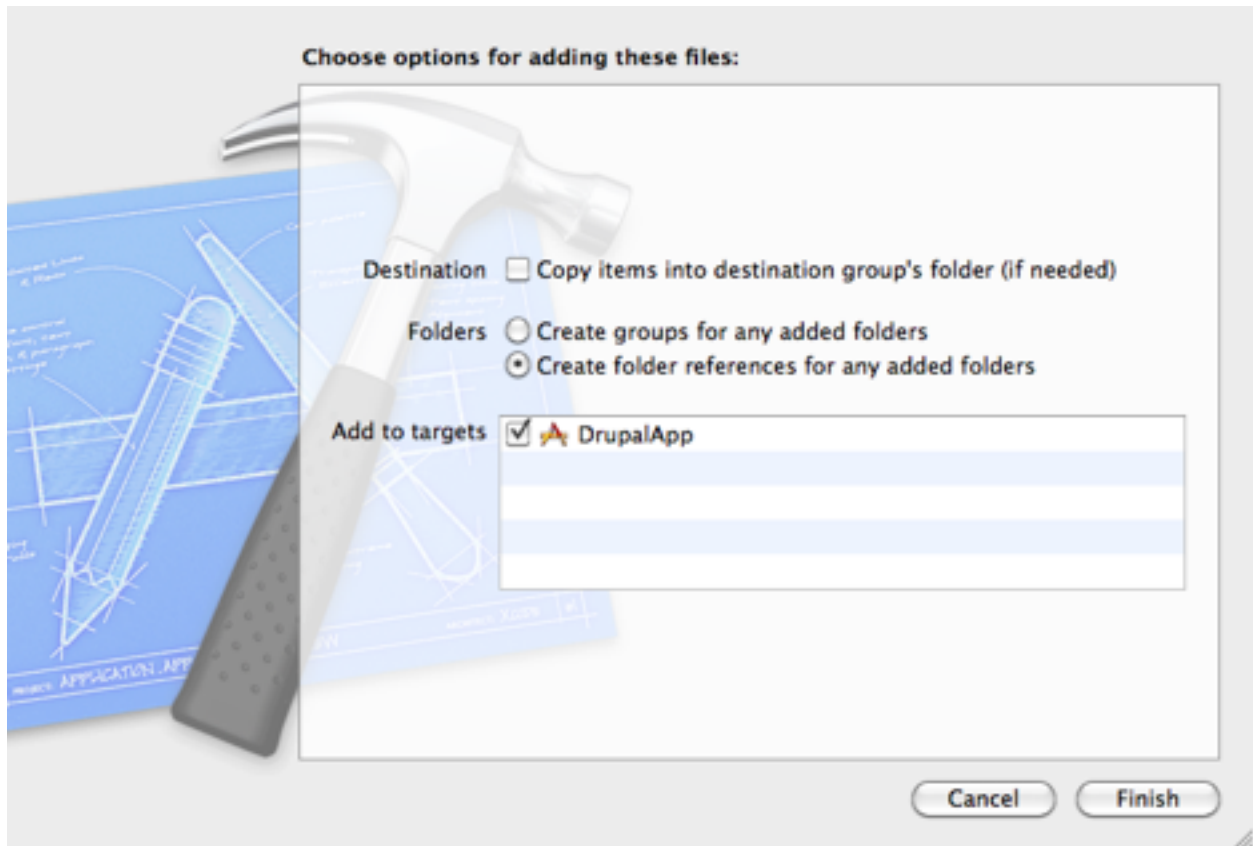
Open up the DrupalApp project folder in Finder. It should look like this:



Drag the www folder onto your XCode DrupalApp Project, right between DrupalApp and PhoneGap.framework. This is so XCode will bundle the contents of the www folder with your app when it packages the app to run on the iPhone Simulator or on an iPhone, iPod Touch, or iPad.

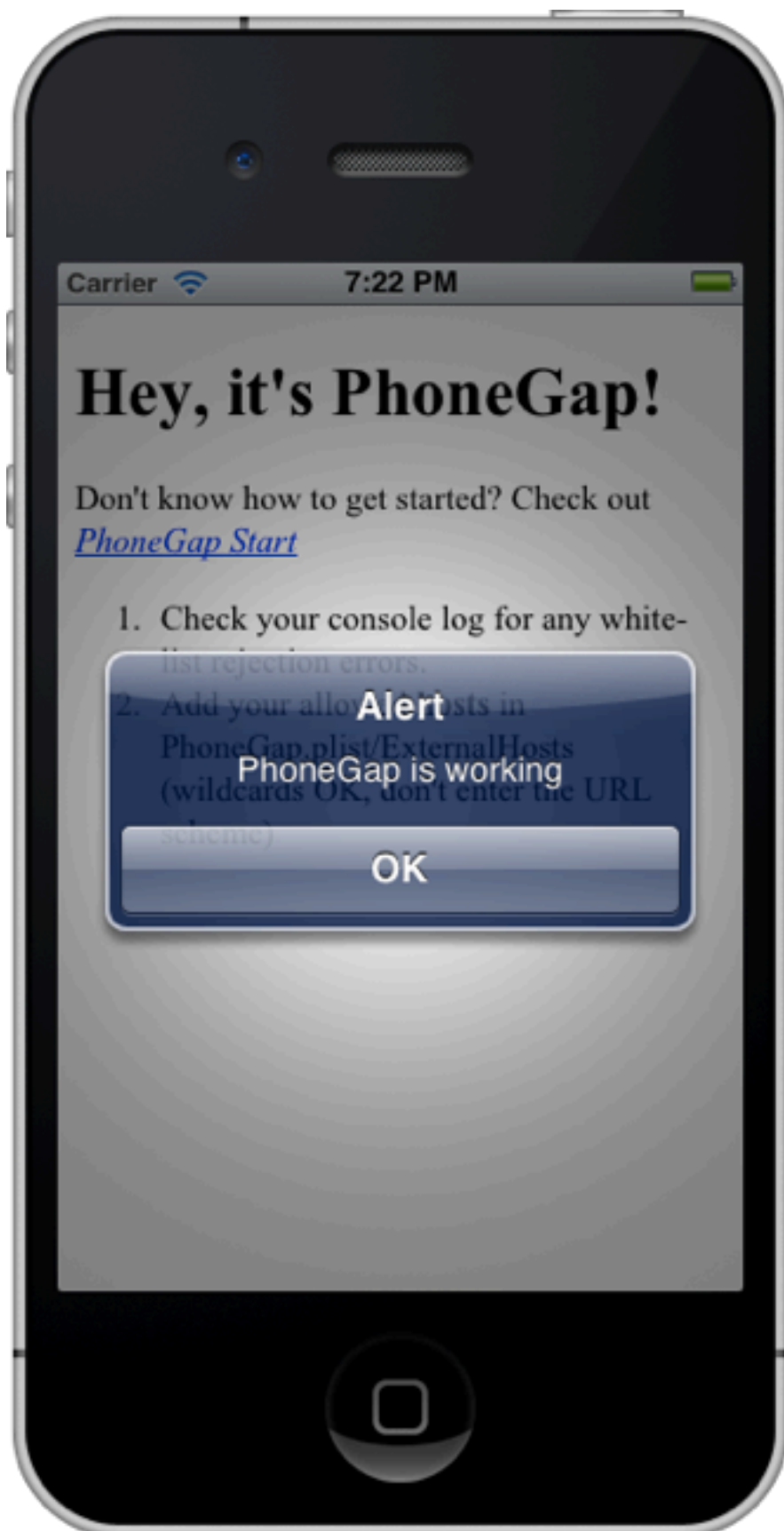


Xcode will prompt you to create folder references for any added folders – be sure to choose this option if the Create groups option is selected.



Click Finish.

Now run the project again, (stopping the currently running DrupalApp if it is still running) and PhoneGap should let you know it's working:



Hey, it's PhoneGap!

Don't know how to get started? Check out [*PhoneGap Start*](#)

1. Check your console log for any white-list rejection errors.
2. Add your allowed hosts in PhoneGap.plist/ExternalHosts (wildcards OK, don't enter the URL scheme)

Alert

PhoneGap is working

OK

More to come!

I hope you've enjoyed the first chapter of my e-book - any feedback you have, please send it to me at jlinwood@gmail.com or reach out to me through Twitter: @jefflinwood

Jeff Linwood