Extracted from:

Build Location-Based Projects for iOS

GPS, Sensors, and Maps

This PDF file contains pages extracted from *Build Location-Based Projects for iOS*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Pragmatic exPress

Build Location-Based Projects for iOS

GPS, Sensors, and Maps

Dominik Hauser

edited by Adaobi Obi Tulton

Build Location-Based Projects for iOS

GPS, Sensors, and Maps

Dominik Hauser

The Pragmatic Bookshelf
Raleigh, North Carolina

Pragmatic Bookshelf

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Executive Editor: Dave Rankin
Copy Editor: Rachel Monaghan
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-781-2
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 2020

Creating a Host Application

We're going to build our logging library step by step, but to verify that each step brings us closer to our goal, we first need something to log. We'll build a simple demo app that acts as a host app for developing the library.

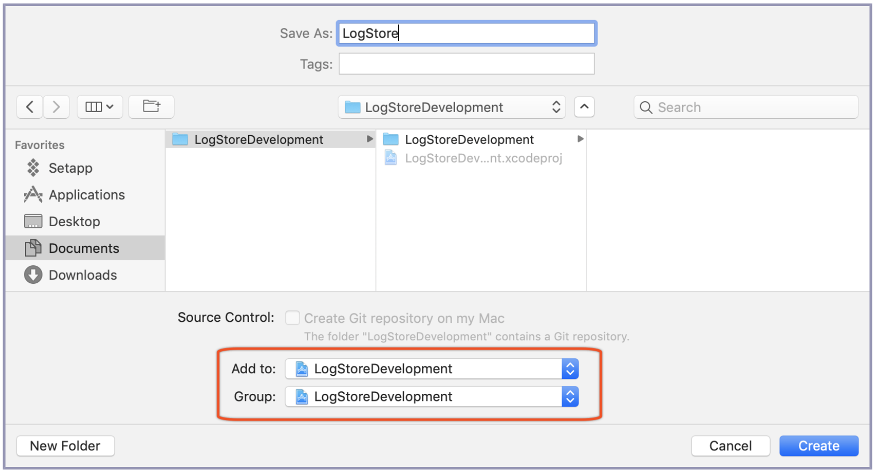
Open Xcode and create a new project by selecting the menu item File > New > Project.... Select the iOS > Single View App template and click Next. Type LogStoreDevelopment for the Product Name. For the user interface, select Storyboard from the drop-down menu and deselect the check marks for Use Core Data, Include Unit Tests, and Include UI Tests, as shown in the image on page 6. Click Next and select a place to store the project. Then click Create.

Now that we have a host app to test our library, we can work on the library itself.

Let's create a Swift package for it, because this is the preferred way to share code between projects.

Creating a Swift Package

Creating a Swift package is quite easy. In Xcode, select the menu item File > New > Swift Package.... Xcode presents a pop-up window for the initial settings of the package. Type LogStore in the Save As field. At the bottom of the window are two drop-down menus, "Add to" and "Group," as shown in the following image. Select LogStoreDevelopment from the "Add to" menu to add the package to the project and then select LogStoreDevelopment again from the "Group" menu to add it to the root folder.



Save As: LogStore

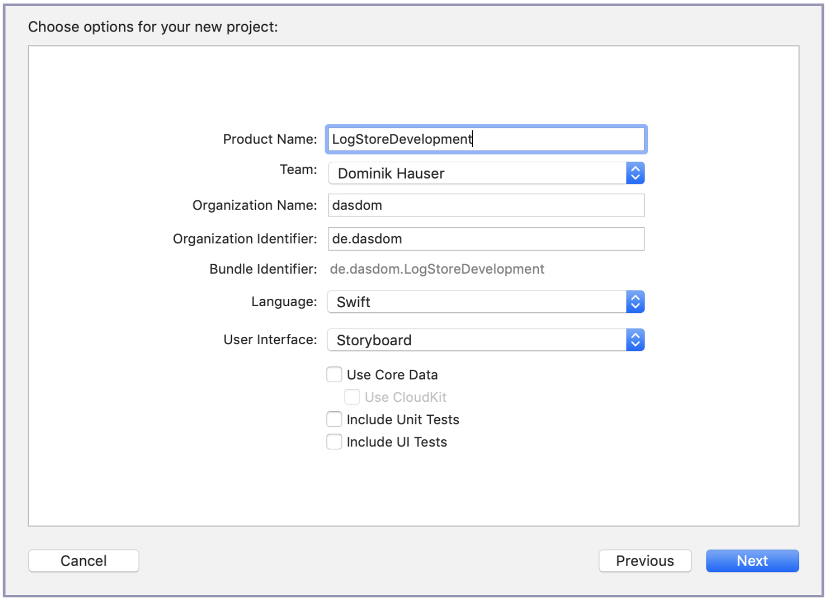
Tags:

Source Control: □ Create Git repository on my Mac

The folder "LogStoreDevelopment" contains a Git repository.

Add to: LogStoreDevelopment

Group: LogStoreDevelopment



Choose options for your new project:

Product Name: LogStoreDevelopment

Team: Dominik Hauser

Organization Name: dasdom

Organization Identifier: de.dasdom

Bundle Identifier: de.dasdom.LogStoreDevelopment

Language: Swift

User Interface: Storyboard

□ Use Core Data

□ Use CloudKit

□ Include Unit Tests

□ Include UI Tests

With these settings, we've added the newly created package to the project we're currently working on. Click Create. Xcode reloads and shows you the created package.

Let's have a look at what's in there. First, you'll see a file named Package.swift. This is the manifest, which comprises all information needed to build the package. For example, if our package depends on other packages, that information would go here.

As Swift packages are the Swift way to share code between different projects, Xcode creates a README file for you. For decades, README files have been the standard way to let developers know what a library does. Let's follow this tradition and add a sentence about what this package will do. Click README.md to open it in the Xcode editor. Replace the template text with this:

```
A tiny package to make log output accessible from within an iOS app.
```

Make a mental note that we have to update this file once we know how the package is used.

Pro Tip: Always Add a Package Description

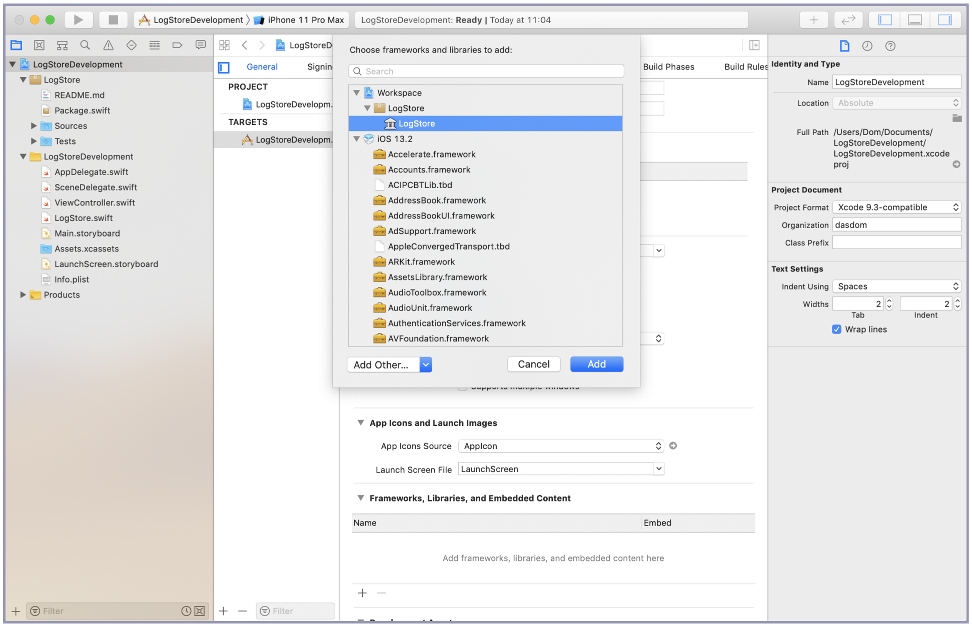
You should always take a few minutes to add information about the package. Your future self will thank you. And don't forget to update the information as the package evolves.

At the moment, the host app has no idea there's a library it can use. Even though the package is in the project of the host app, we have to add the resulting library to its *target* in a separate step. Targets are what Xcode uses to manage build products. For example, if you created an app with a Watch app companion, you'd have two or more targets in the same project. Our host app has only one target. Let's add the library generated by the Swift package to this target.

Open the project navigator and select the topmost element with the blue icon, LogStoreDevelopment. Xcode opens the project settings. To open the target settings, select the target in the TARGETS overview. Make sure the General tab is selected. Scroll down until you see the section Frameworks, Libraries, and Embedded Content. Click the + button to add a new library. In the pop-up window, select the LogStore library and click Add.

The image on page 8 shows what you should see on your screen.

Done! We can now use the package in the host app. Our Swift package is set up, so now let's fill it with life.



LogStoreDevelopment: **Ready** | Today at 11:04

Choose frameworks and libraries to add:

Name: LogStoreDevelopment

Location: Absolute

Full Path: /Users/Dom/Documents/LogStoreDevelopment/LogStoreDevelopment.xcodeproj

Project Format: Xcode 9.3-compatible

Organization: dasdom

Class Prefix:

Indent Using: Spaces

Widths: 2 Tab, 2 Indent

☑ Wrap lines

App Icons Source: AppIcon

Launch Screen File: LaunchScreen

Add frameworks, libraries, and embedded content here