Caffeine Integration: How-To

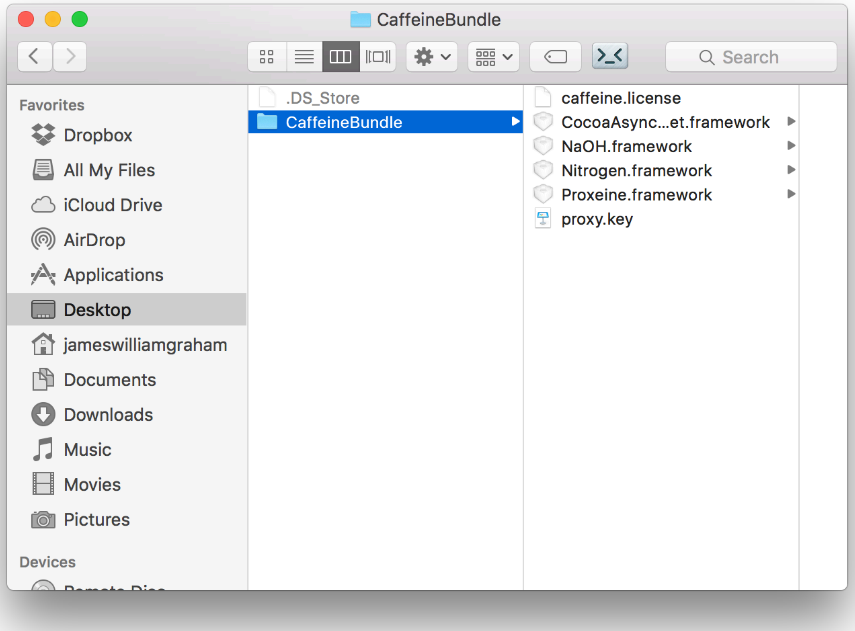
If at any time you require help, please contact our engineers at help@caffei.net.

Server Side Integration

- Currently no changes need be made to your server or backend. We have a server close to yours that interprets Caffeine as HTTP/S and makes the request to your existing backend.

Client Side Integration

You should have received a Caffeine bundle containing six items: four frameworks, a proxy.key, and a caffeine.license (pictured below)



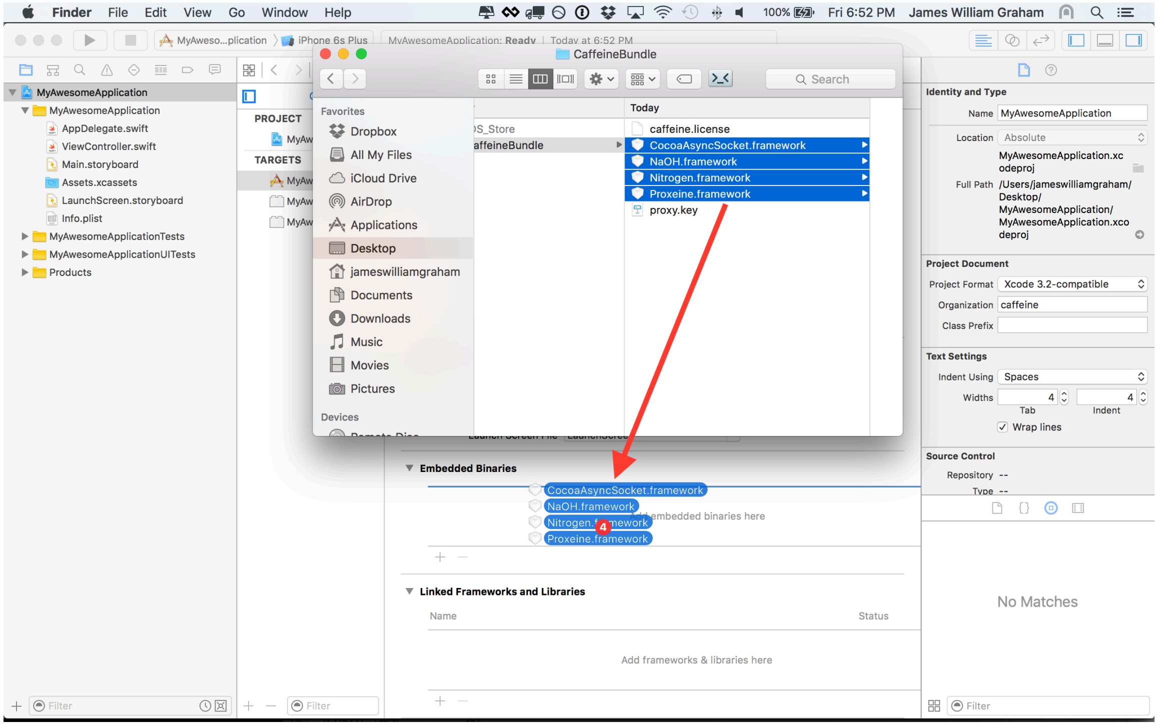
.DS_Store
CaffeineBundle

caffeine.license
CocoaAsync...et.framework
NaOH.framework
Nitrogen.framework
Proxeine.framework
proxy.key

- The caffeine.license file is required to use Caffeine. It contains information that uniquely identifies your organization.
- CocoaAsyncSocket.framework is an open source library that Caffeine uses for fast-UDP diagnostic logging.
- NaOH.framework is an open source library built and maintained by us - forked from libsodium, implementing Caffeine's security.
- Nitrogen.framework contains our proprietary protocol.
- Proxeine.framework is the interpreter that converts whatever networking library is used on the client into Caffeine.
- Proxy.key contains the public key the client will use to authenticate with our proxy server.

Client side integration (continued)

1. Drag and drop the four frameworks into the Embedded Binaries section in Xcode (found on the project's general tab).



Identity and Type

Name: MyAwesomeApplication

Location: Absolute

MyAwesomeApplication.xcodeproj

Full Path: /Users/jameswilliamgraham/Desktop/MyAwesomeApplication/MyAwesomeApplication.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible

Organization: caffeine

Class Prefix:

Text Settings

Indent Using: Spaces

Widths: 4 (Tab) 4 (Indent)

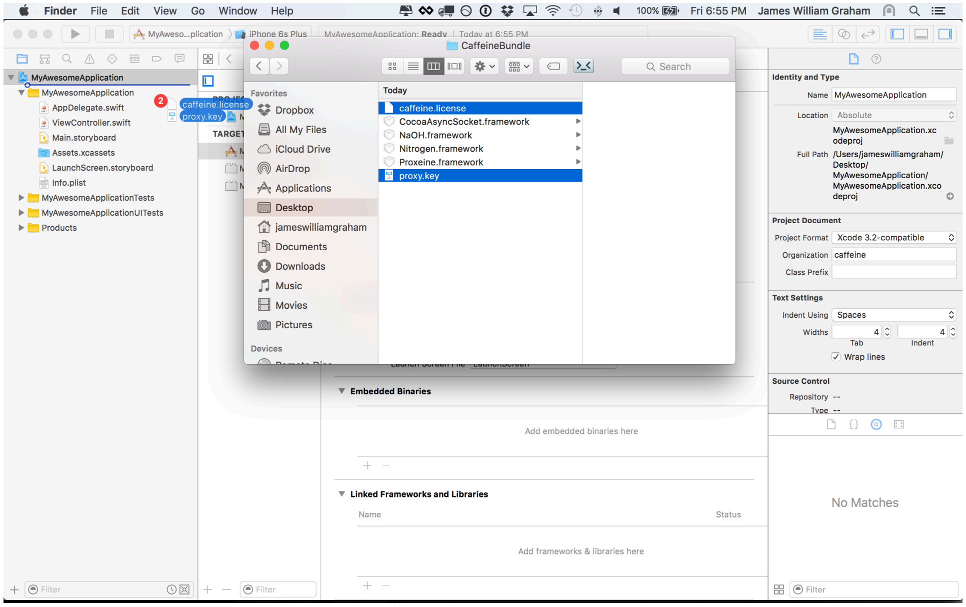
✓ Wrap lines

Source Control

Repository: --

Type: --

2. Drag and drop the proxy.key and caffeine.license into the Xcode project and add to the main application target.



Identity and Type

Name: MyAwesomeApplication

Location: Absolute

MyAwesomeApplication.xcodeproj

Full Path: /Users/jameswilliamgraham/Desktop/MyAwesomeApplication/MyAwesomeApplication.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible

Organization: caffeine

Class Prefix:

Text Settings

Indent Using: Spaces

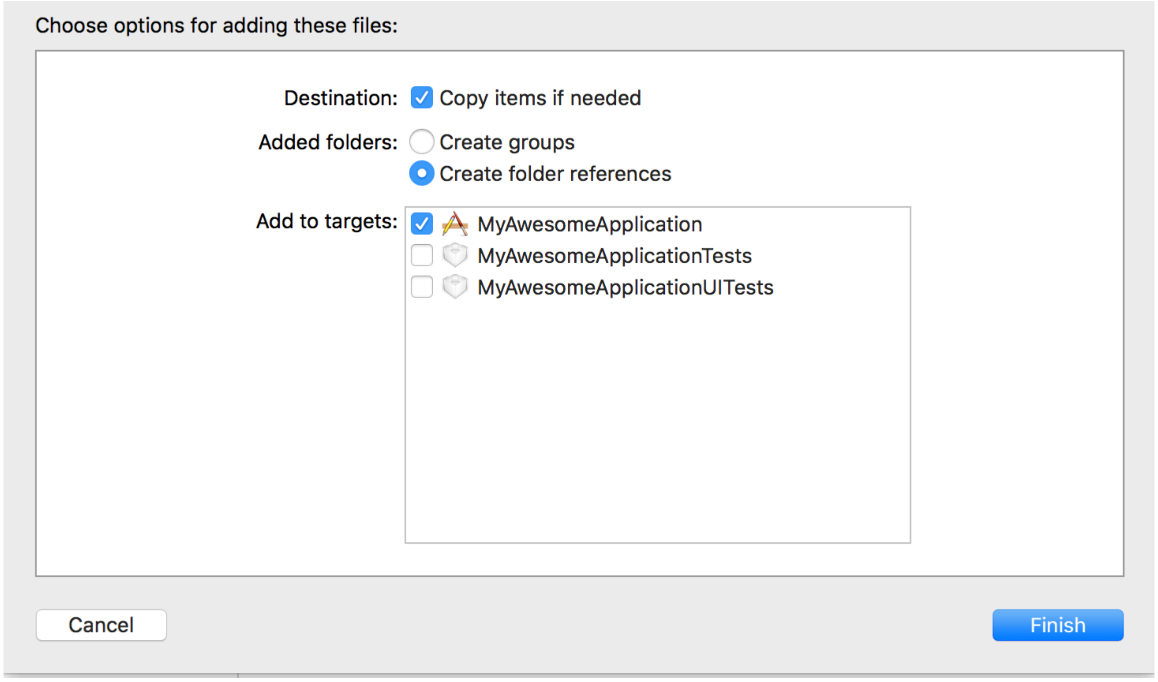
Widths: 4 (Tab) 4 (Indent)

✓ Wrap lines

Source Control

Repository: --

Type: --



Choose options for adding these files:

Destination: ☑ Copy items if needed

Added folders: ○ Create groups
● Create folder references

Add to targets:

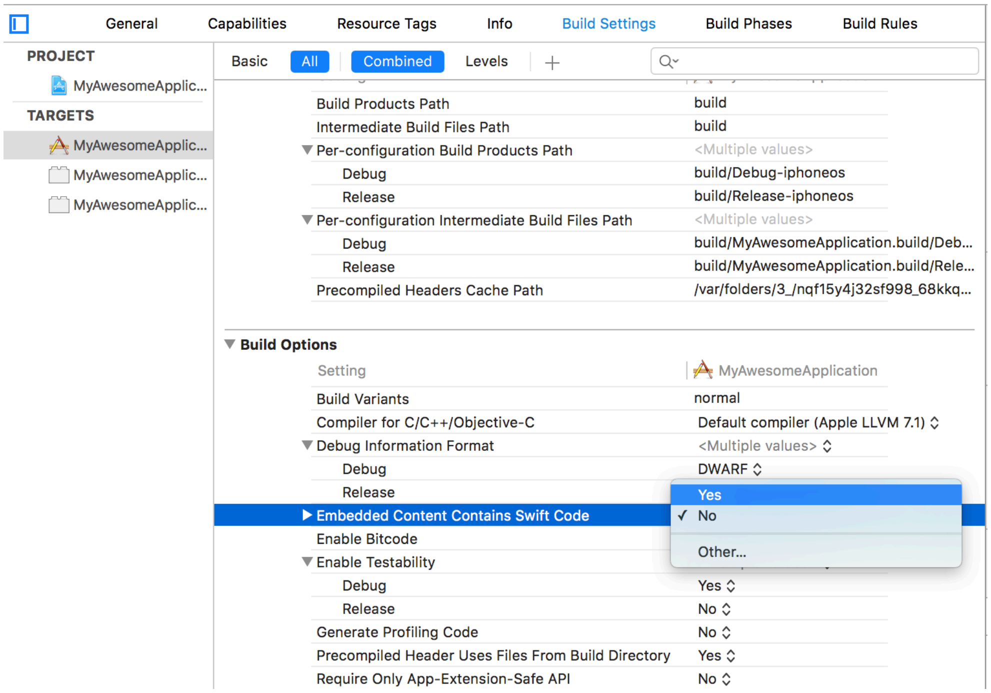
☑ MyAwesomeApplication

☐ MyAwesomeApplicationTests

☐ MyAwesomeApplicationUITests

(Adding to main application target)

3. In the main target's build settings, set "Embedded Content Contains Swift Code" to Yes



Build Products Path	build
Intermediate Build Files Path	build
▼ Per-configuration Build Products Path	<Multiple values>
Debug	build/Debug-iphoneos
Release	build/Release-iphoneos
▼ Per-configuration Intermediate Build Files Path	<Multiple values>
Debug	build/MyAwesomeApplication.build/Deb...
Release	build/MyAwesomeApplication.build/Rele...
Precompiled Headers Cache Path	/var/folders/3_/nqf15y4j32sf998_68kkq...

▼ Build Options

Setting	MyAwesomeApplication
Build Variants	normal
Compiler for C/C++/Objective-C	Default compiler (Apple LLVM 7.1) ◇
▼ Debug Information Format	<Multiple values> ◇
Debug	DWARF ◇
Release	
▶ Embedded Content Contains Swift Code	Yes ✓ No Other...
Enable Bitcode	
▼ Enable Testability	
Debug	Yes ◇
Release	No ◇
Generate Profiling Code	No ◇
Precompiled Header Uses Files From Build Directory	Yes ◇
Require Only App-Extension-Safe API	No ◇

4. In your App Delegate, import the Proxeine framework. Locate the *application:didFinishLaunchingWithOptions:* function and call the function:

Swift:
CaffeineHTTPProxy.start()

Objective-C:
[CaffeineHTTPProxy start];

5. Caffeine defaults to off, so no traffic will be accelerated until specific URLs or sets of URLs are specified. This is done through CaffeineHTTPProxy's *allow* and *deny* functions.

   - To accelerate a specific URL, add

Swift:
CaffeineHTTPProxy.allow("http://google.com")

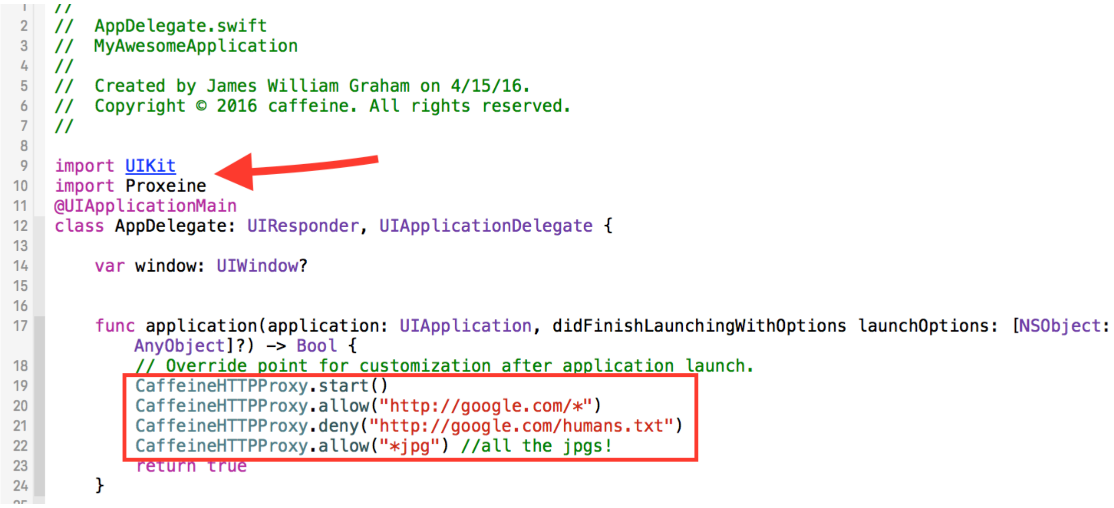
Objective-C:
[CaffeineHTTPProxy allow:@"http://google.com"];

- To accelerate all URLs in a subdirectory, add

Swift:
CaffeineHTTPProxy.allow(http://google.com/*)

Objective-C:
[CaffeineHTTPProxy allow:@"http://google.com/*];

As long as the URL starts with http://google.com, the request will be Caffeinated.



```
//
//  AppDelegate.swift
//  MyAwesomeApplication
//
//  Created by James William Graham on 4/15/16.
//  Copyright © 2016 caffeine. All rights reserved.
//

import UIKit
import Proxeine
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject:
        AnyObject]?) -> Bool {
        // Override point for customization after application launch.
        CaffeineHTTPProxy.start()
        CaffeineHTTPProxy.allow("http://google.com/*")
        CaffeineHTTPProxy.deny("http://google.com/humans.txt")
        CaffeineHTTPProxy.allow("*jpg") //all the jpgs!
        return true
    }
```