GETTING STARTED WITH XCODE

Nick Chen

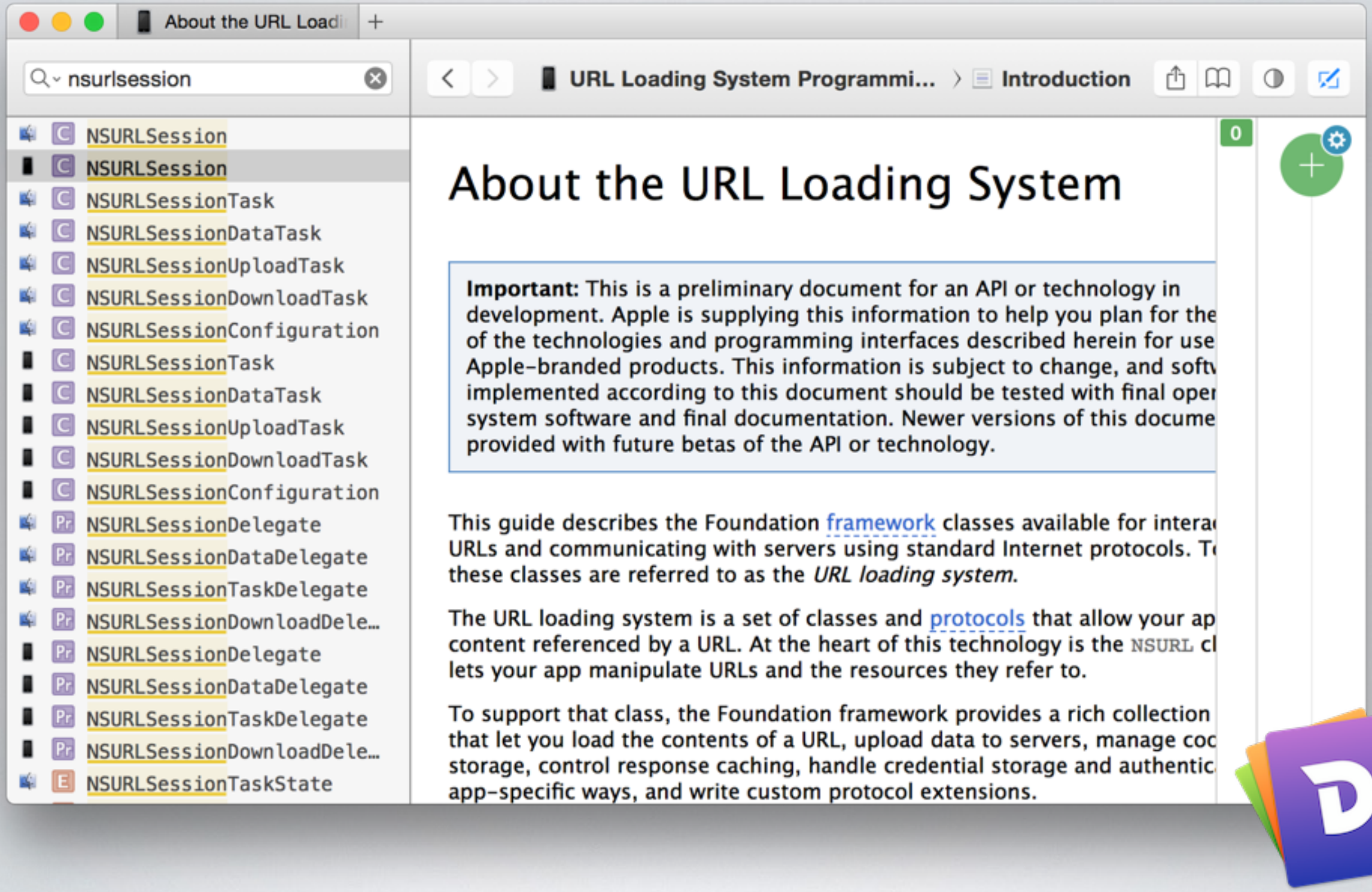
Fall 2015

talentspark.io

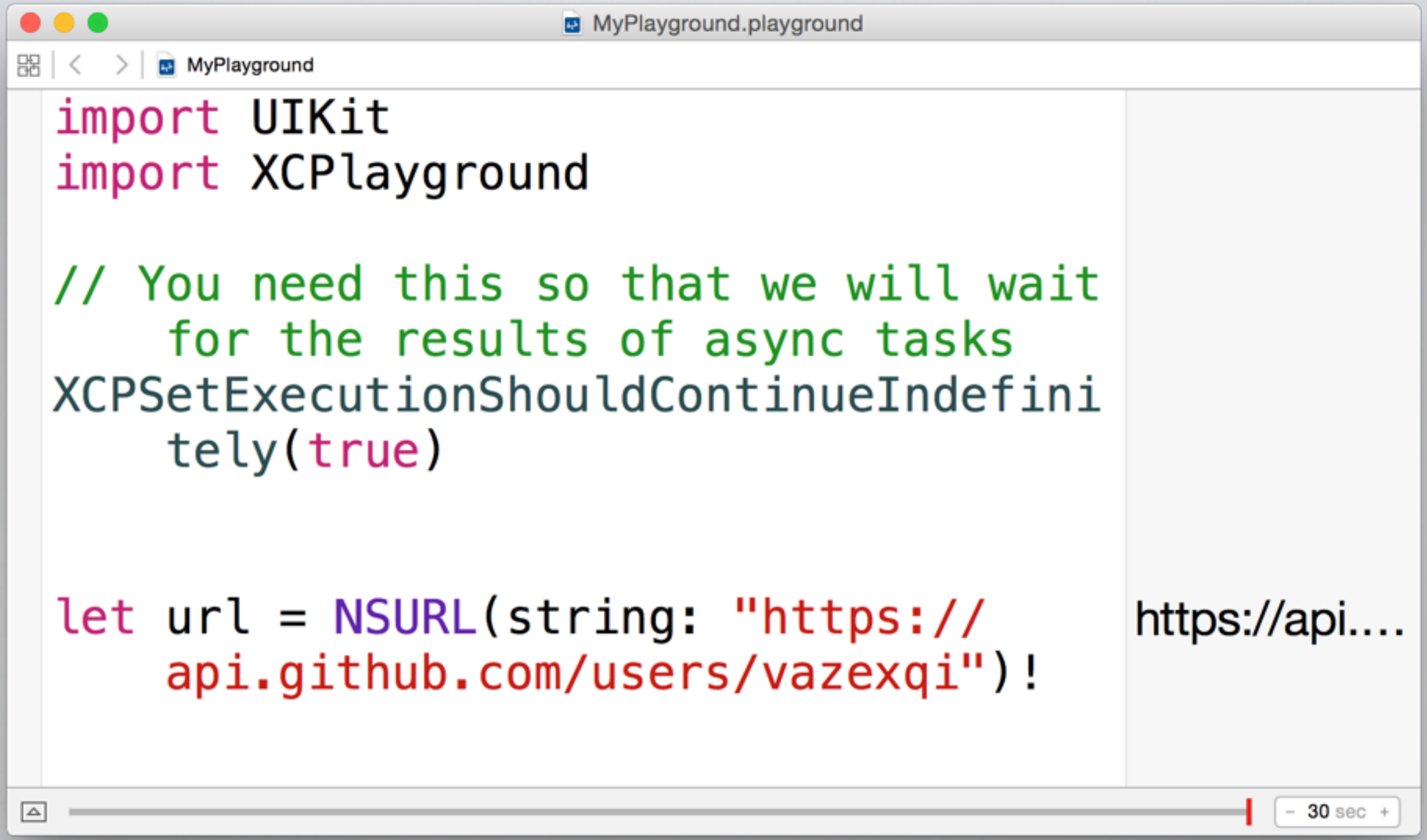
GOALS

- Refresh Swift knowledge
- Learn how developers use CocoaPods + Xcode
- Pick up Xcode tips and tricks

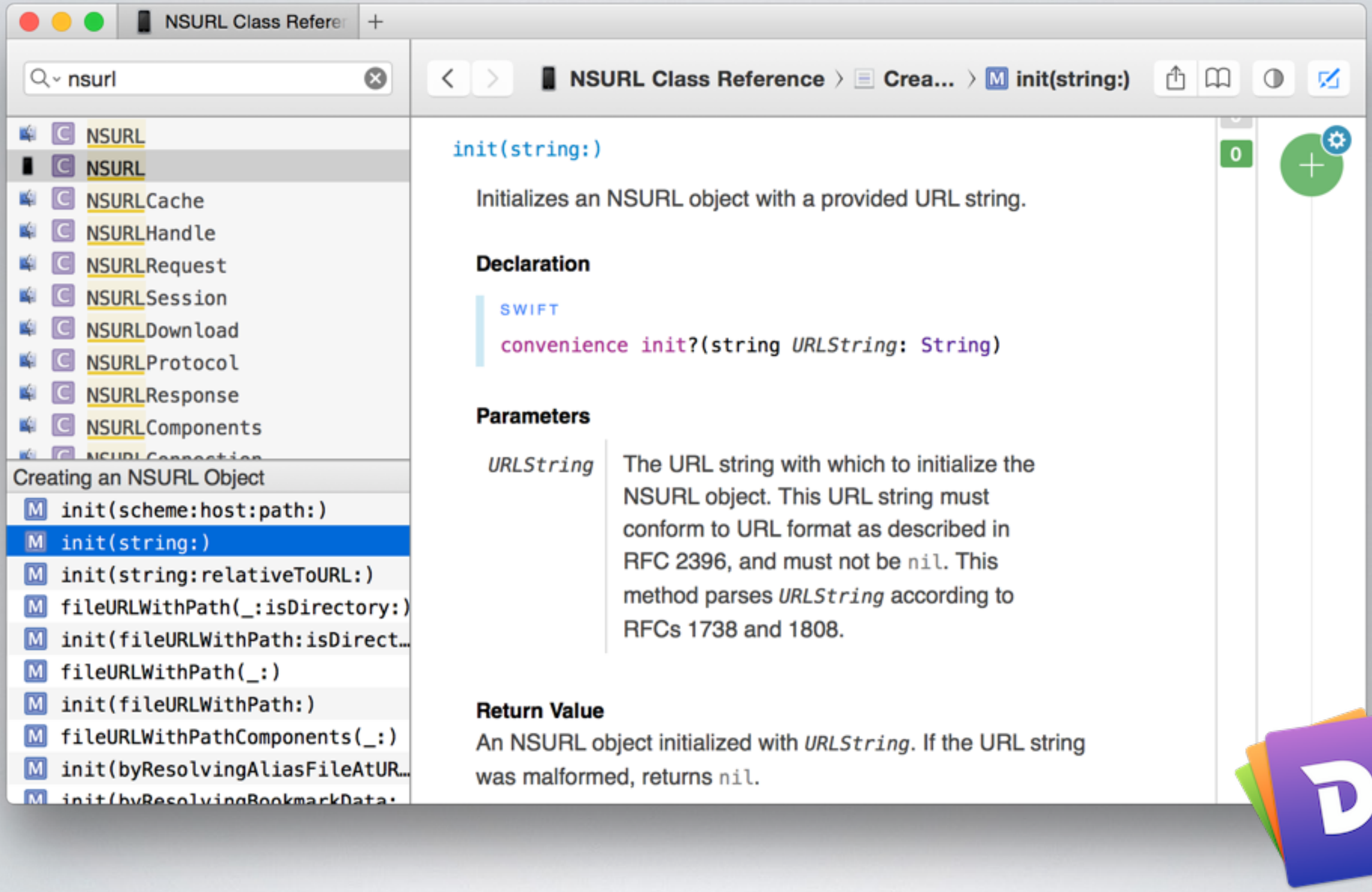
LET'S LOAD A URL



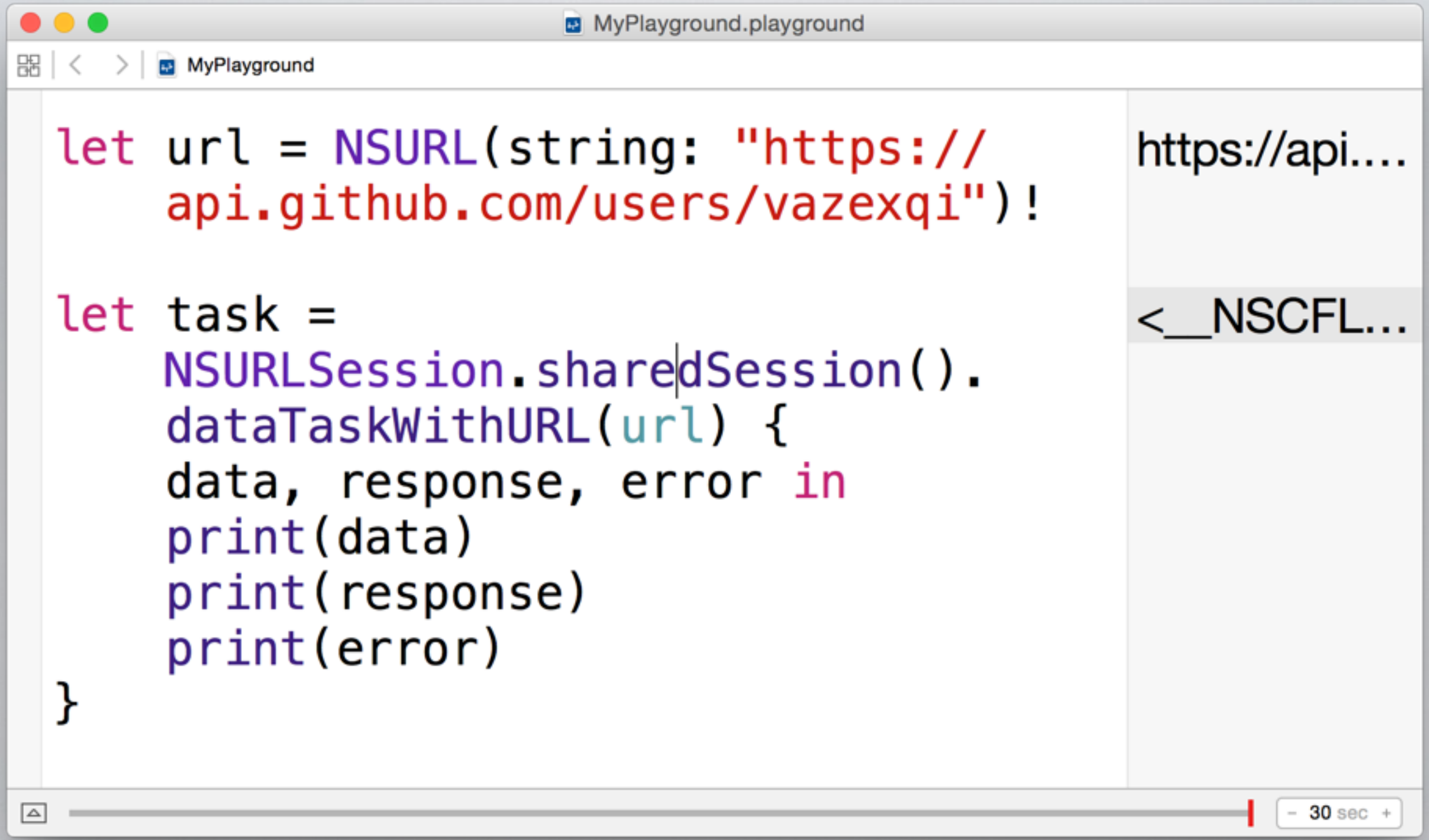
NSURL


MyPlayground.playground
MyPlayground
import UIKit
import XCPlayground
// You need this so that we will wait
for the results of async tasks
XCPSetExecutionShouldContinueIndefini
tely(true)
let url = NSURL(string: "https://
api.github.com/users/vazexqi")!
https://api....
30 sec

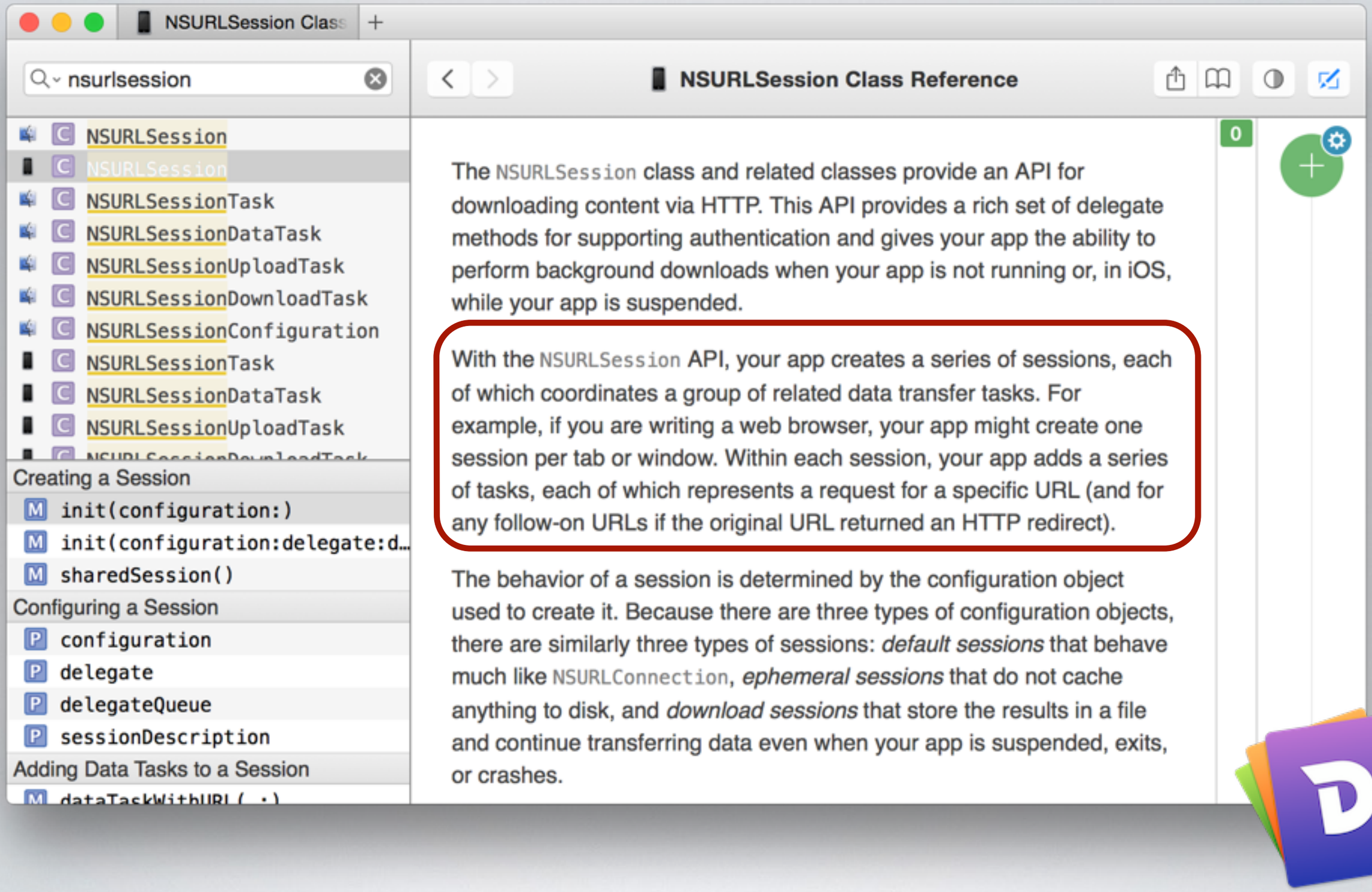
NSURL


NSURL Class Refere
nsurl
NSURL Class Reference
Crea...
init(string:)
NSURL
NSURL
NSURLCache
NSURLHandle
NSURLRequest
NSURLSession
NSURLDownload
NSURLProtocol
NSURLResponse
NSURLComponents
Creating an NSURL Object
init(scheme:host:path:)
init(string:)
init(string:relativeToURL:)
fileURLWithPath(_:isDirectory:)
init(fileURLWithPath:isDirect…
fileURLWithPath(_:)
init(fileURLWithPath:)
fileURLWithPathComponents(_:)
init(byResolvingAliasFileAtUR…
init(string:)
Initializes an NSURL object with a provided URL string.
Declaration
SWIFT
convenience init?(string URLString: String)
Parameters
URLString
The URL string with which to initialize the NSURL object. This URL string must conform to URL format as described in RFC 2396, and must not be nil. This method parses URLString according to RFCs 1738 and 1808.
Return Value
An NSURL object initialized with URLString. If the URL string was malformed, returns nil.

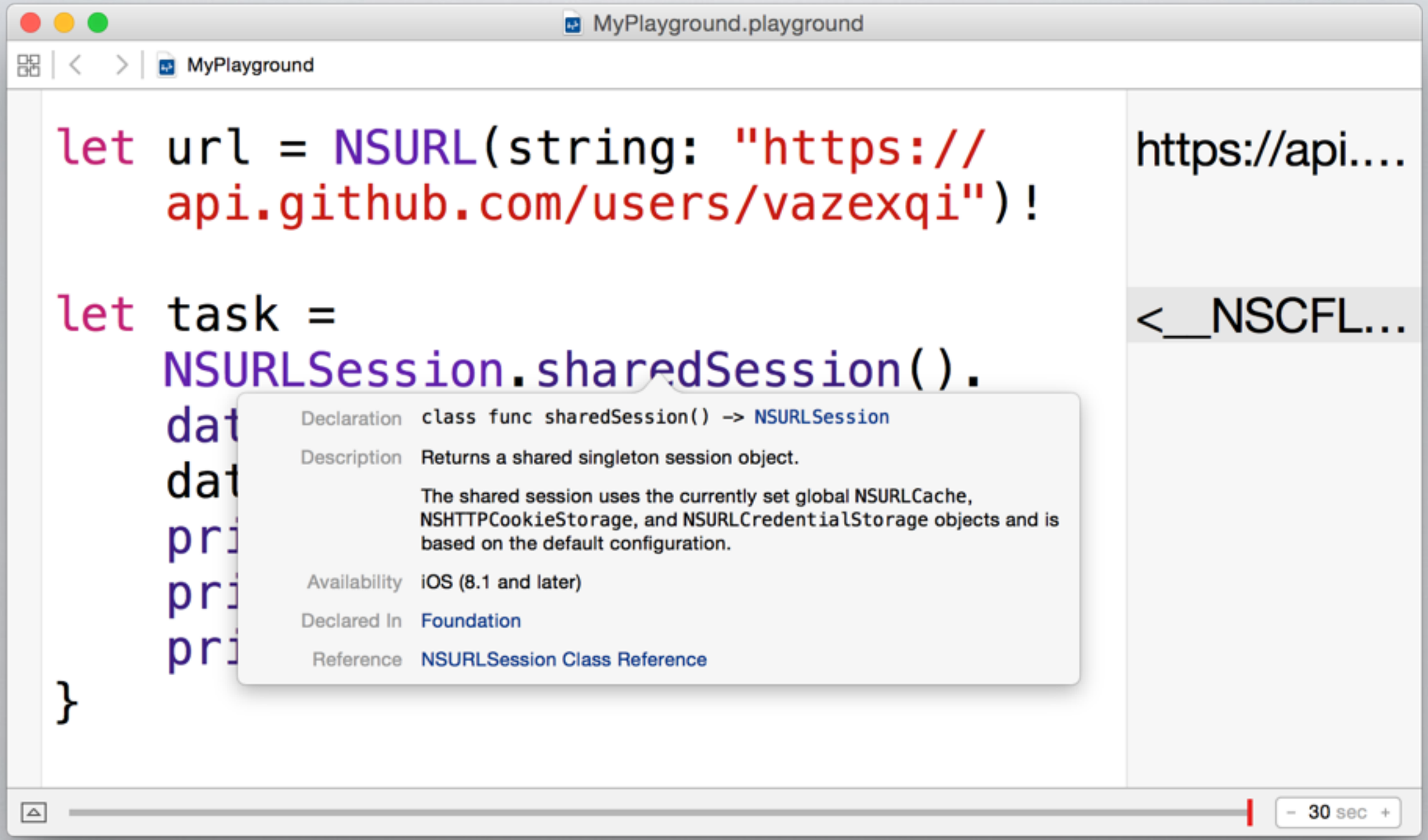
NSURLSESSION



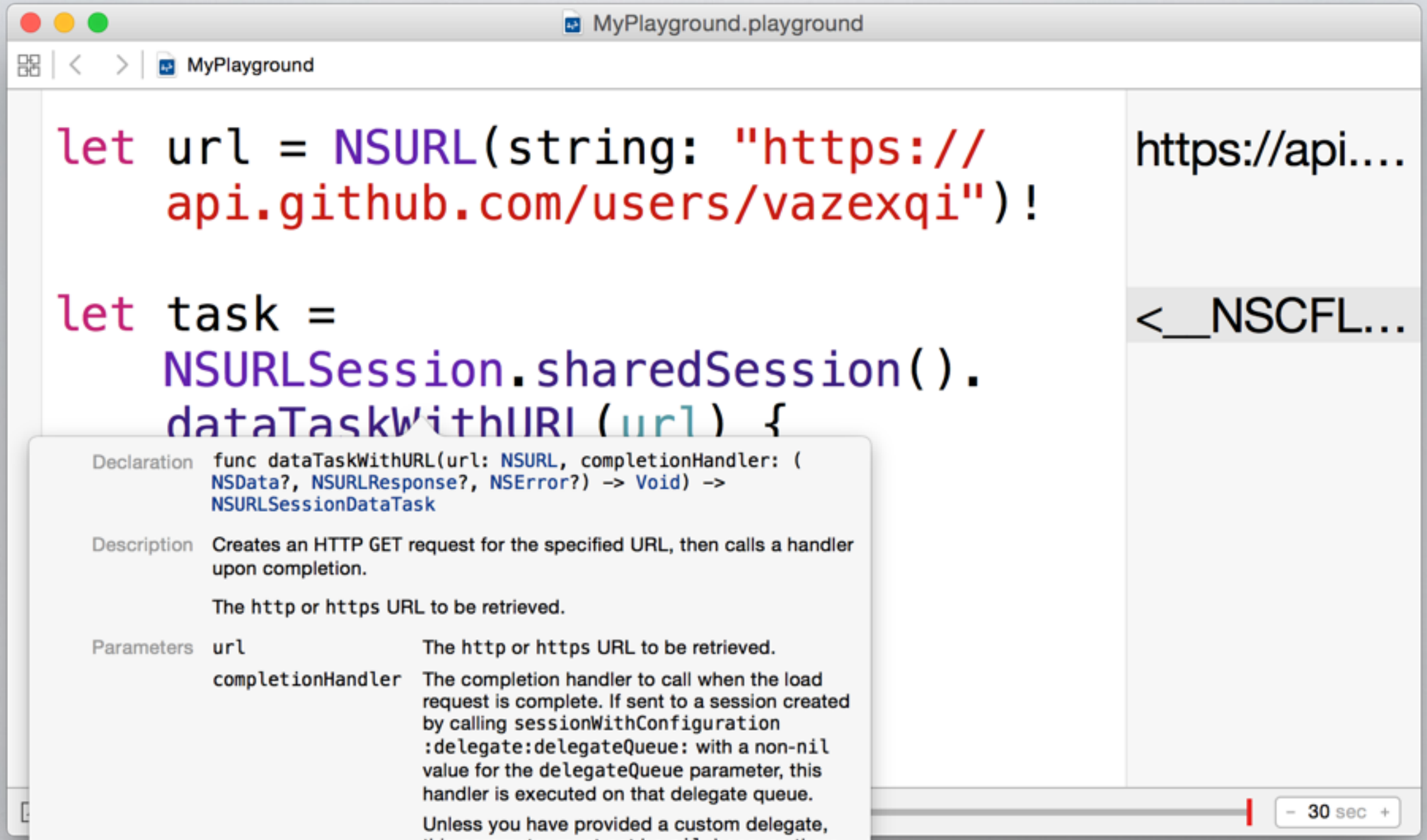
NSURLSESSION



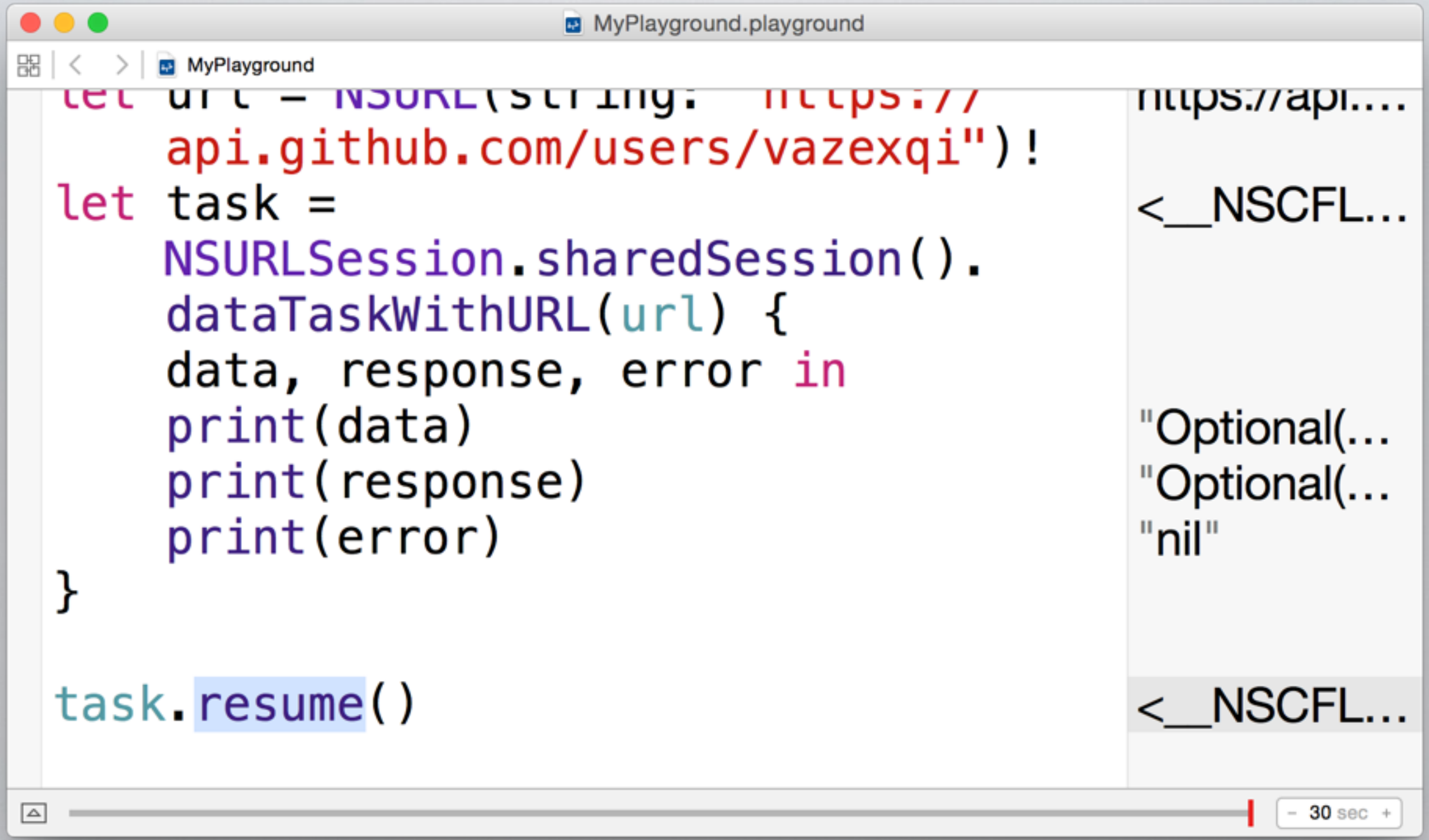
NSURLSESSION



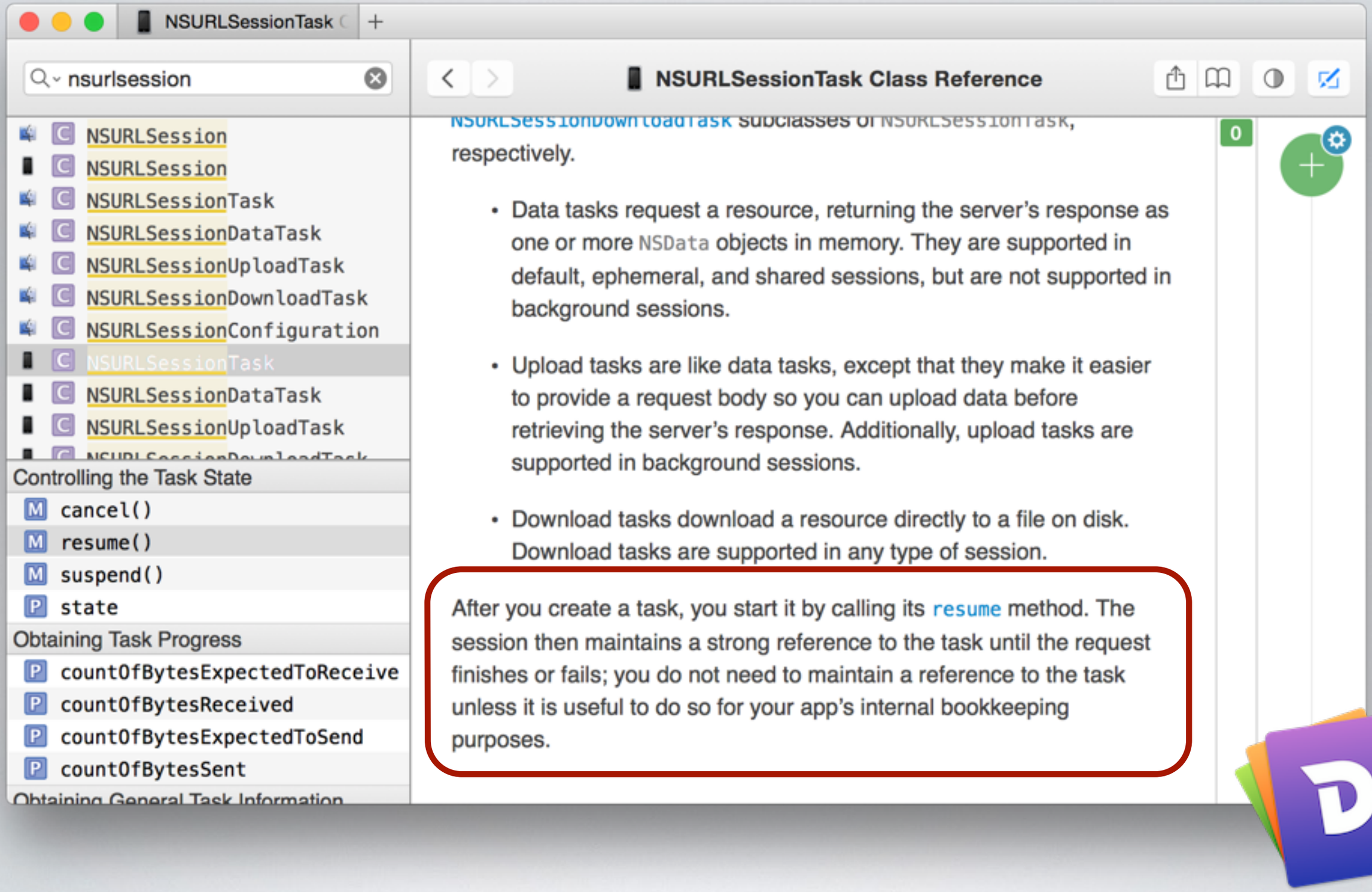
NSURLSESSION



TASK


MyPlayground.playground
MyPlayground
let url = NSURL(string: "https://
api.github.com/users/vazexqi")!
let task =
NSURLSession.sharedSession().
dataTaskWithURL(url) {
data, response, error in
print(data)
print(response)
print(error)
}
task.resume()
https://api....
<__NSCFL...
"Optional(...
"Optional(...
"nil"
<__NSCFL...
30 sec

TASK


NSURLSessionTask
nsurlsession
NSURLSessionTask Class Reference
NSURLSession
NSURLSession
NSURLSessionTask
NSURLSessionDataTask
NSURLSessionUploadTask
NSURLSessionDownloadTask
NSURLSessionConfiguration
NSURLSessionTask
NSURLSessionDataTask
NSURLSessionUploadTask
Controlling the Task State
cancel()
resume()
suspend()
state
Obtaining Task Progress
countOfBytesExpectedToReceive
countOfBytesReceived
countOfBytesExpectedToSend
countOfBytesSent
Obtaining General Task Information
NSURLSessionDownloadTask subclasses of NSURLSessionTask,
respectively.
Data tasks request a resource, returning the server's response as one or more NSData objects in memory. They are supported in default, ephemeral, and shared sessions, but are not supported in background sessions.
Upload tasks are like data tasks, except that they make it easier to provide a request body so you can upload data before retrieving the server's response. Additionally, upload tasks are supported in background sessions.
Download tasks download a resource directly to a file on disk. Download tasks are supported in any type of session.
After you create a task, you start it by calling its resume method. The session then maintains a strong reference to the task until the request finishes or fails; you do not need to maintain a reference to the task unless it is useful to do so for your app's internal bookkeeping purposes.
0

RECAP

RECAP

NSURL

RECAP

NSURLSession

NSURL

RECAP

NSURLSessionDataTask

NSURLSession

NSURL

RECAP

NSURLSessionDataTask

resume()

NSURLSession

NSURL

RECAP

NSURLSessionUploadTask

NSURLSessionDataTask

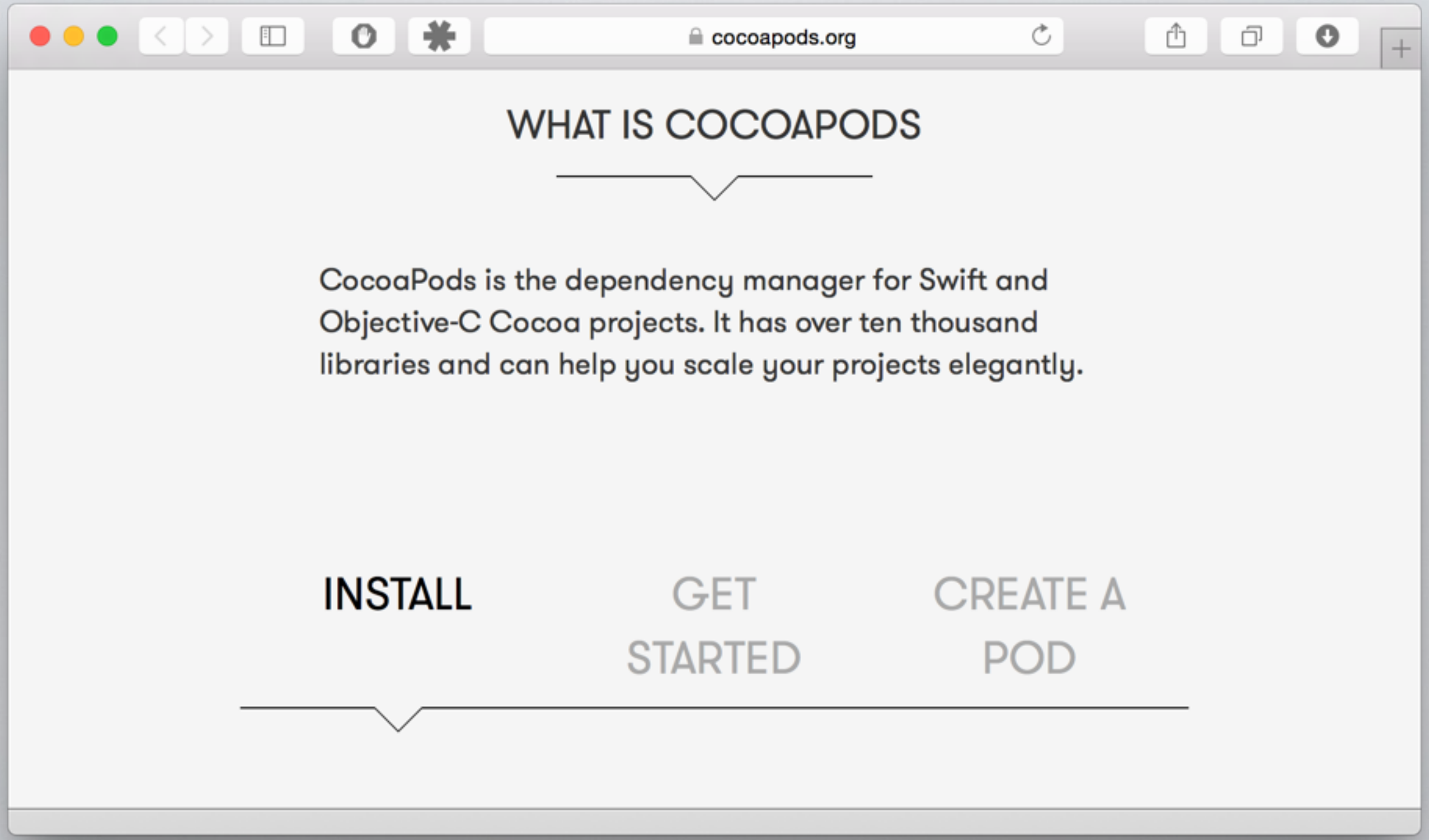
resume()

NSURLSession

NSURL

NSURLSessionDownloadTask

COCOAPODS


cocoapods.org
WHAT IS COCOAPODS
CocoaPods is the dependency manager for Swift and Objective-C Cocoa projects. It has over ten thousand libraries and can help you scale your projects elegantly.
INSTALL
GET STARTED
CREATE A POD

COCOAPODS

COCOAPODS

My Super Cool Project

COCOAPODS

Framework #1

My Super Cool Project

COCOAPODS

Framework #1

My Super Cool Project

Framework #2

COCOAPODS

Framework #1

My Super Cool Project

Framework #1.a

Framework #2

COCOAPODS

Framework #1

Framework #2.a

My Super Cool Project

Framework #1.a

Framework #2

COCOAPODS

Framework #1

My Super Cool Project

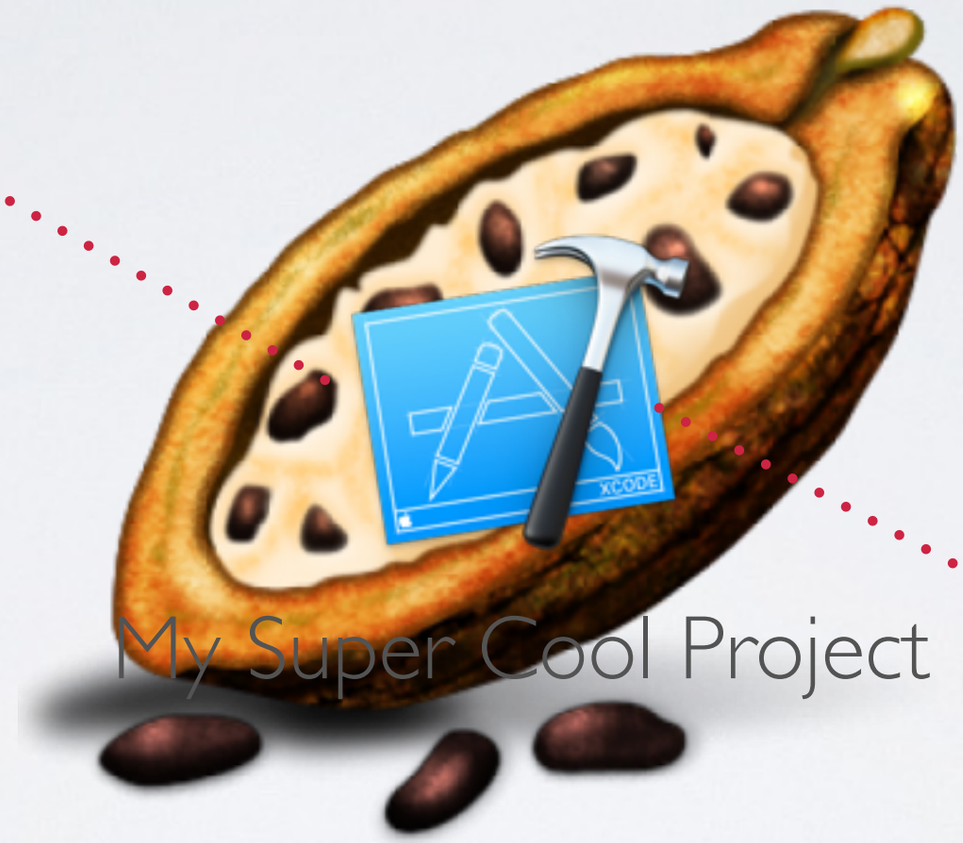
Framework #2.a

Framework #1.a

Framework #2

COCOAPODS

Framework #1

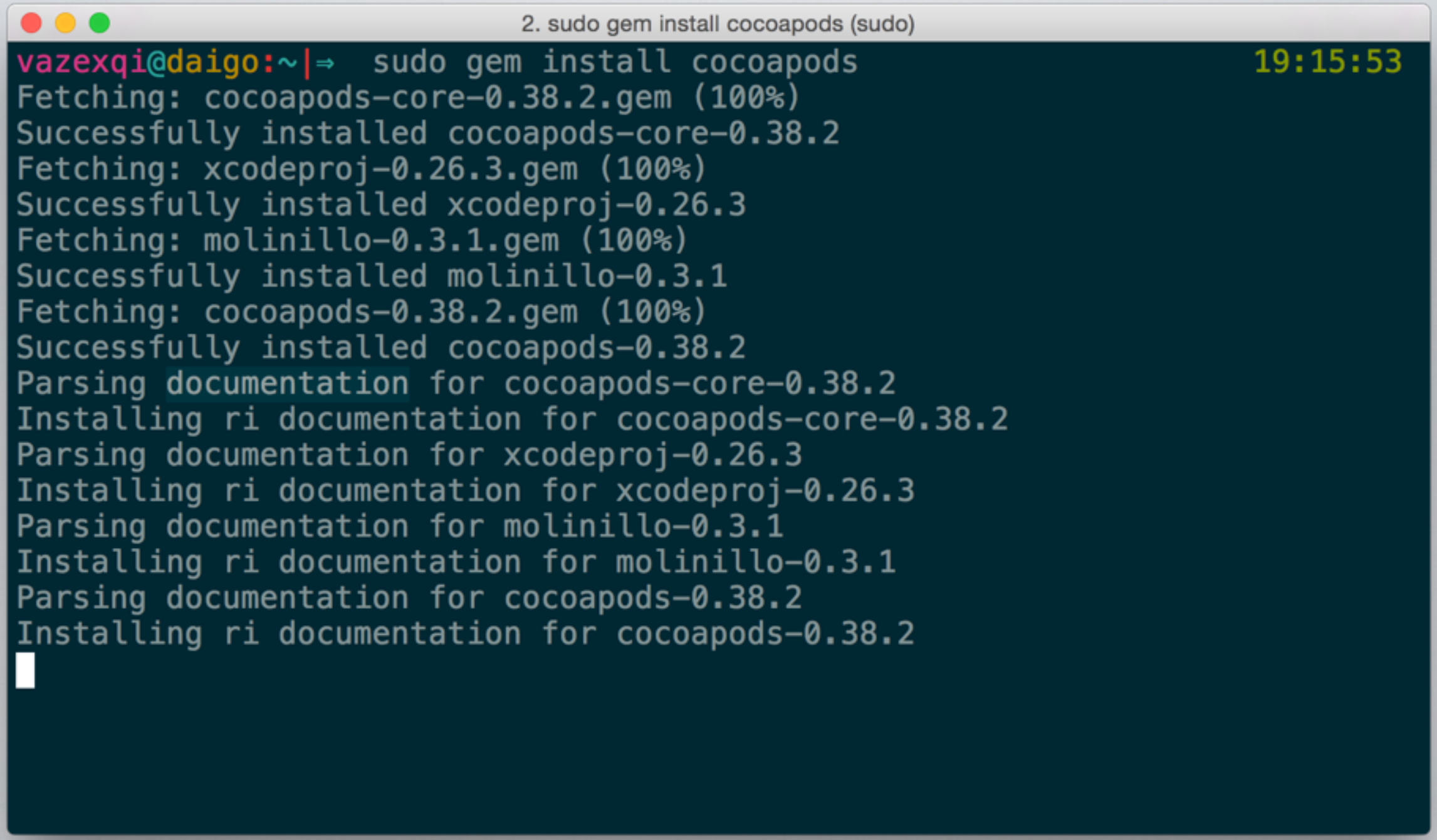


Framework #2.a

Framework #1.a

Framework #2

COCOAPODS

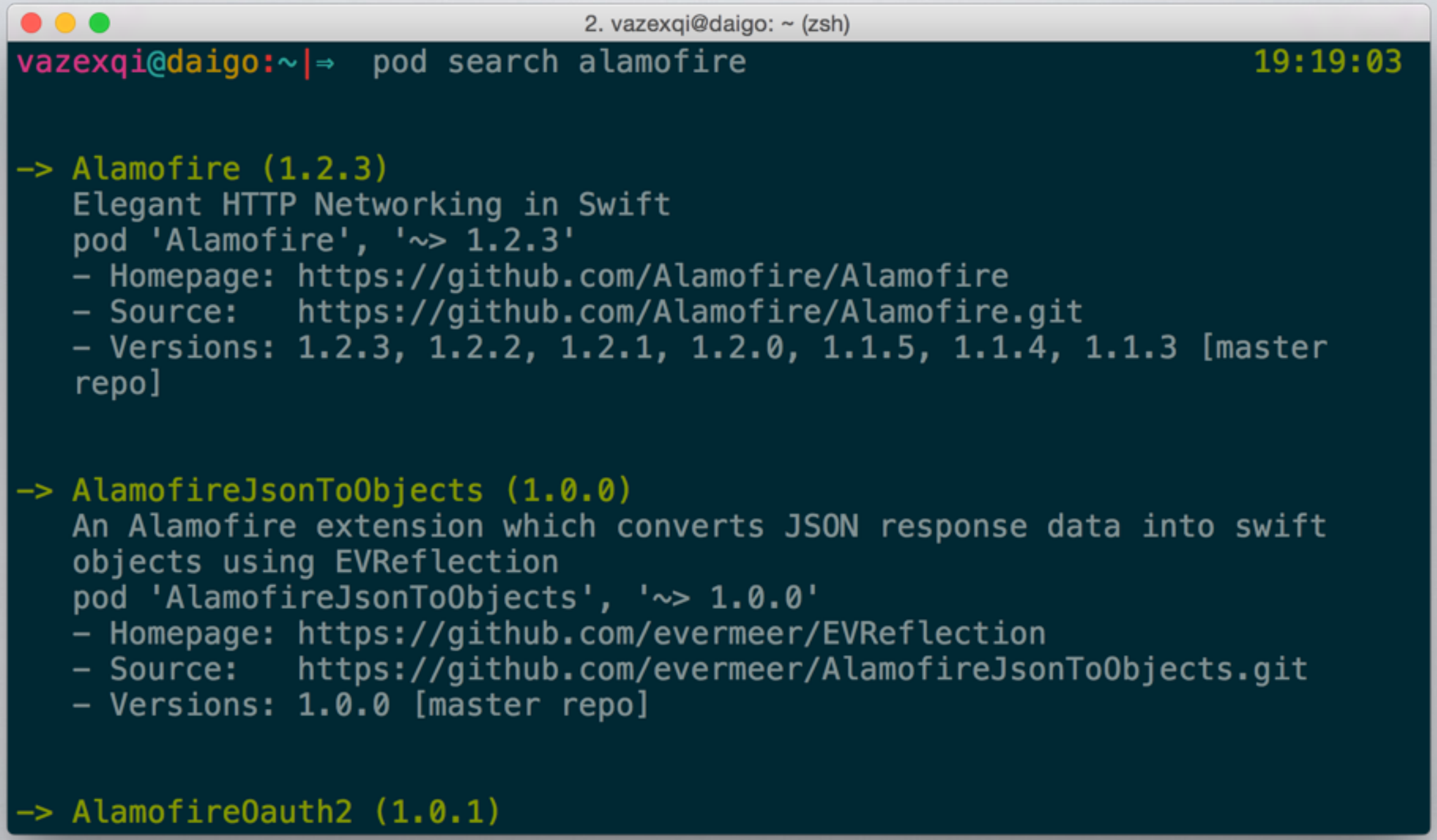

2. sudo gem install cocoapods (sudo)
vazexqi@daigo:~|⇒ sudo gem install cocoapods 19:15:53
Fetching: cocoapods-core-0.38.2.gem (100%)
Successfully installed cocoapods-core-0.38.2
Fetching: xcodeproj-0.26.3.gem (100%)
Successfully installed xcodeproj-0.26.3
Fetching: molinillo-0.3.1.gem (100%)
Successfully installed molinillo-0.3.1
Fetching: cocoapods-0.38.2.gem (100%)
Successfully installed cocoapods-0.38.2
Parsing documentation for cocoapods-core-0.38.2
Installing ri documentation for cocoapods-core-0.38.2
Parsing documentation for xcodeproj-0.26.3
Installing ri documentation for xcodeproj-0.26.3
Parsing documentation for molinillo-0.3.1
Installing ri documentation for molinillo-0.3.1
Parsing documentation for cocoapods-0.38.2
Installing ri documentation for cocoapods-0.38.2

COCOAPODS

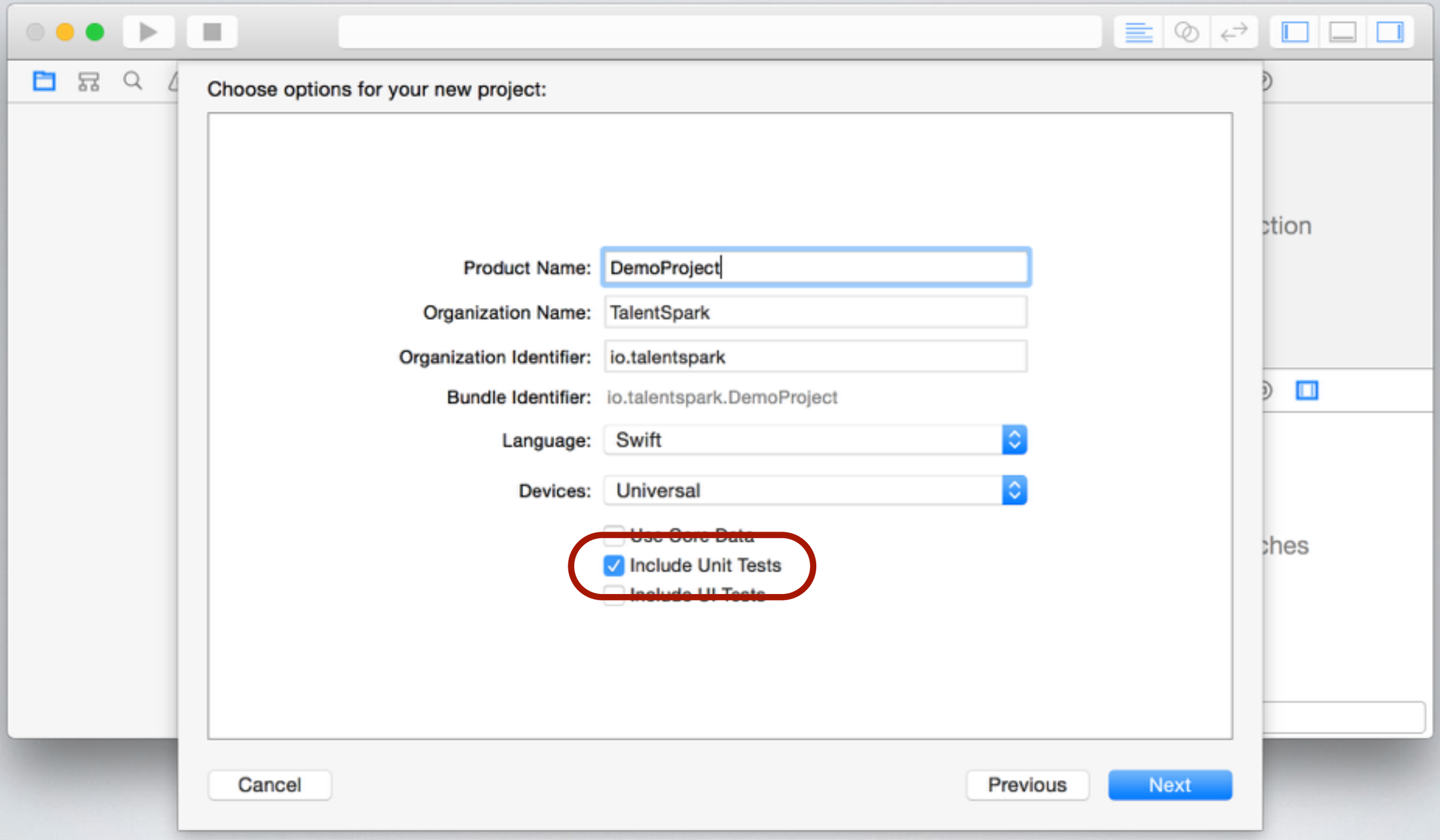
2. vazexqi@daigo: ~ (zsh)

```
vazexqi@daigo:~|⇒  pod --help                                    19:17:10
Usage:

    $ pod COMMAND

      CocoaPods, the Cocoa library package manager.

Commands:

    + cache       Manipulate the CocoaPods cache
    + init        Generate a Podfile for the current directory.
    + install     Install project dependencies to Podfile.lock versions
    + ipc         Inter-process communication
    + lib         Develop pods
    + list        List pods
    + outdated    Show outdated project dependencies
    + plugins     Show available CocoaPods plugins
    + repo        Manage spec-repositories
    + search      Searches for pods
    + setup       Setup the CocoaPods environment
    + spec        Manage pod specs
    + trunk       Interact with the CocoaPods API (e.g. publishing new
```

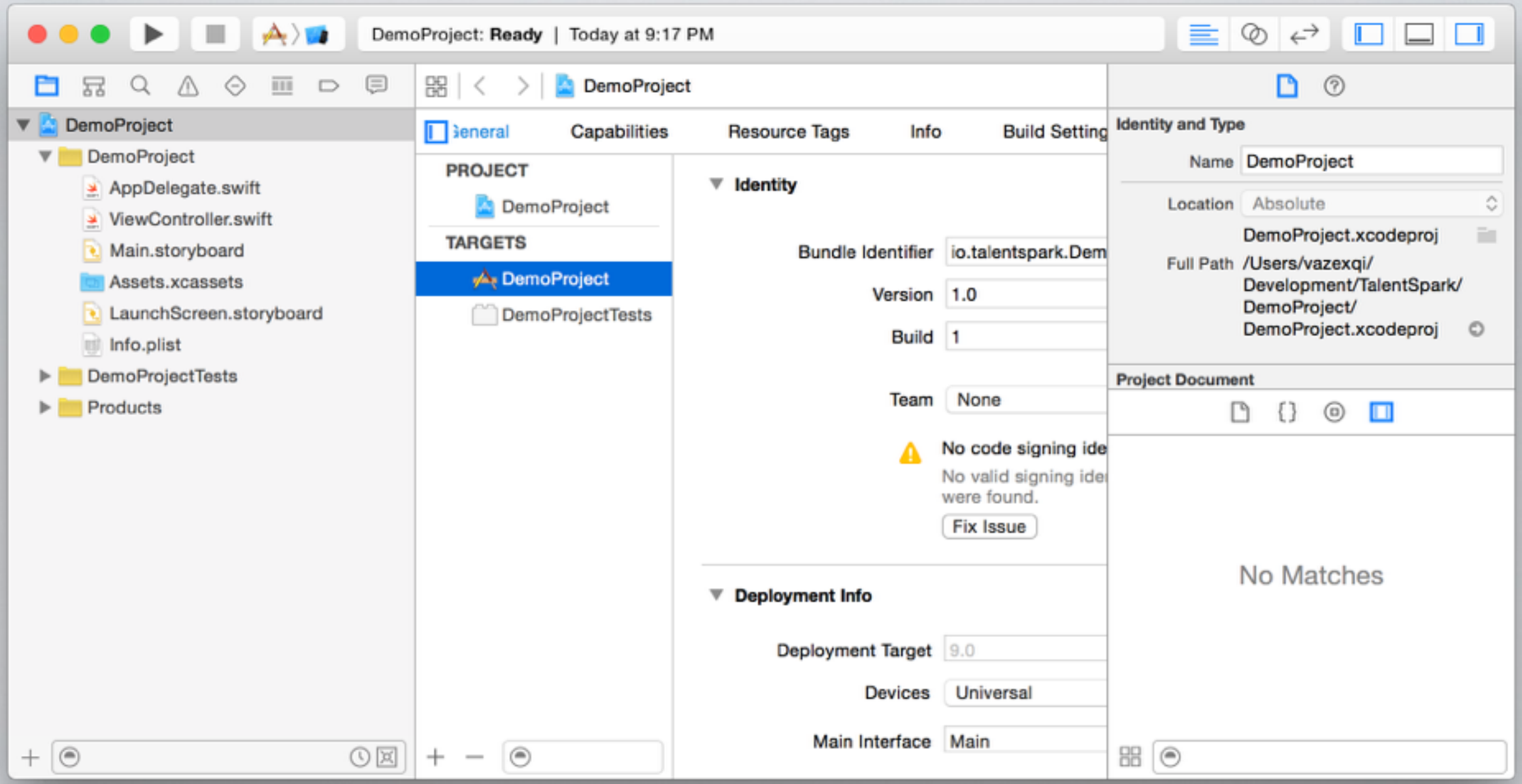
COCOAPODS


2. vazexqi@daigo: ~ (zsh)
vazexqi@daigo:~|⇒ pod search alamofire
19:19:03
-> Alamofire (1.2.3)
Elegant HTTP Networking in Swift
pod 'Alamofire', '~> 1.2.3'
- Homepage: https://github.com/Alamofire/Alamofire
- Source: https://github.com/Alamofire/Alamofire.git
- Versions: 1.2.3, 1.2.2, 1.2.1, 1.2.0, 1.1.5, 1.1.4, 1.1.3 [master repo]
-> AlamofireJsonToObjects (1.0.0)
An Alamofire extension which converts JSON response data into swift objects using EVReflection
pod 'AlamofireJsonToObjects', '~> 1.0.0'
- Homepage: https://github.com/evermeer/EVReflection
- Source: https://github.com/evermeer/AlamofireJsonToObjects.git
- Versions: 1.0.0 [master repo]
-> AlamofireOauth2 (1.0.1)

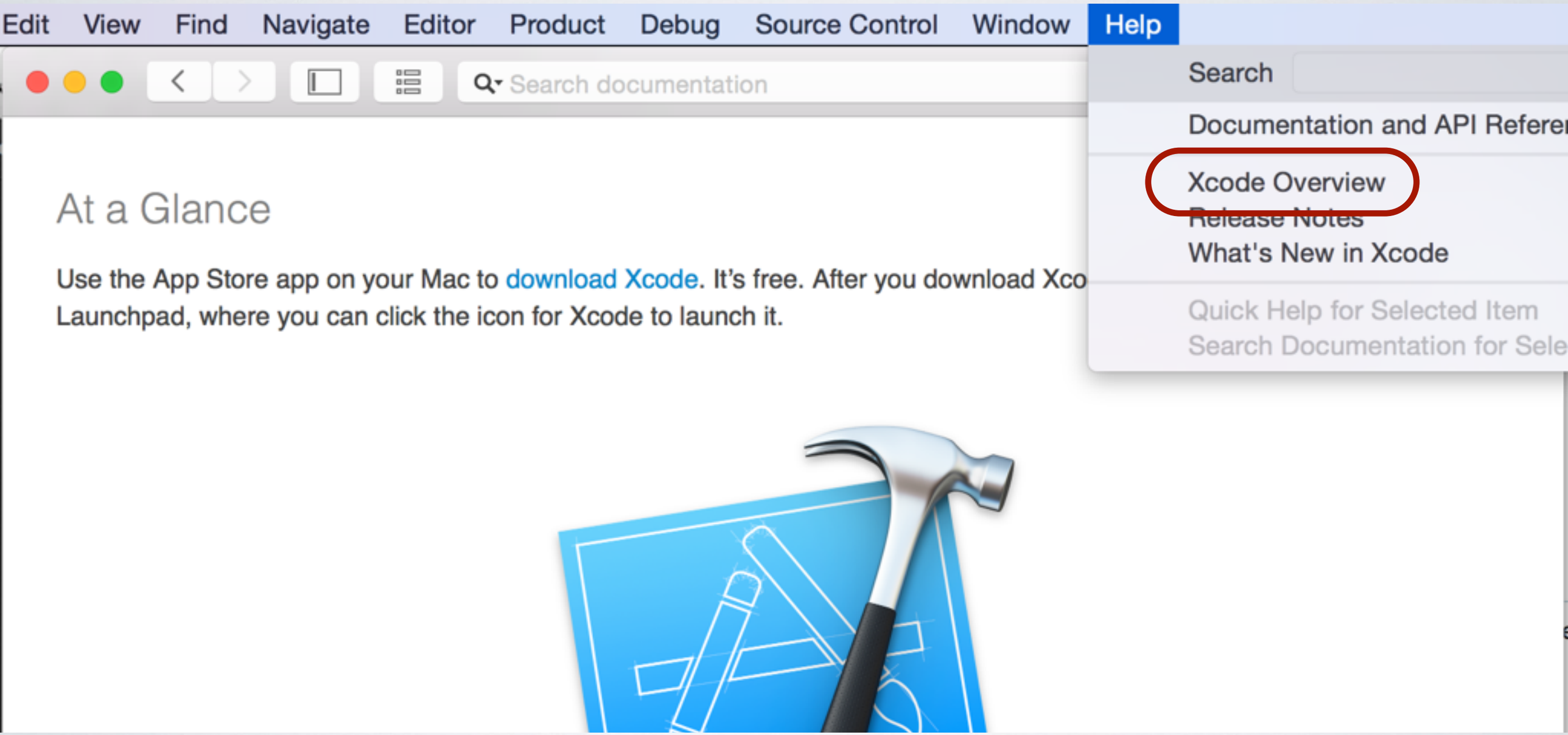
XCODE


Choose options for your new project:
Product Name: DemoProject
Organization Name: TalentSpark
Organization Identifier: io.talentspark
Bundle Identifier: io.talentspark.DemoProject
Language: Swift
Devices: Universal
Include Unit Tests
Cancel
Previous
Next

XCODE


DemoProject: Ready | Today at 9:17 PM
DemoProject
DemoProject
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
DemoProjectTests
Products
General
Capabilities
Resource Tags
Info
Build Setting
PROJECT
DemoProject
TARGETS
DemoProject
DemoProjectTests
Identity
Bundle Identifier
io.talentspark.Dem
Version
1.0
Build
1
Team
None
No code signing ide
No valid signing ide
were found.
Fix Issue
Deployment Info
Deployment Target
9.0
Devices
Universal
Main Interface
Main
Identity and Type
Name
DemoProject
Location
Absolute
DemoProject.xcodeproj
Full Path
/Users/vazexqi/
Development/TalentSpark/
DemoProject/
DemoProject.xcodeproj
Project Document
No Matches

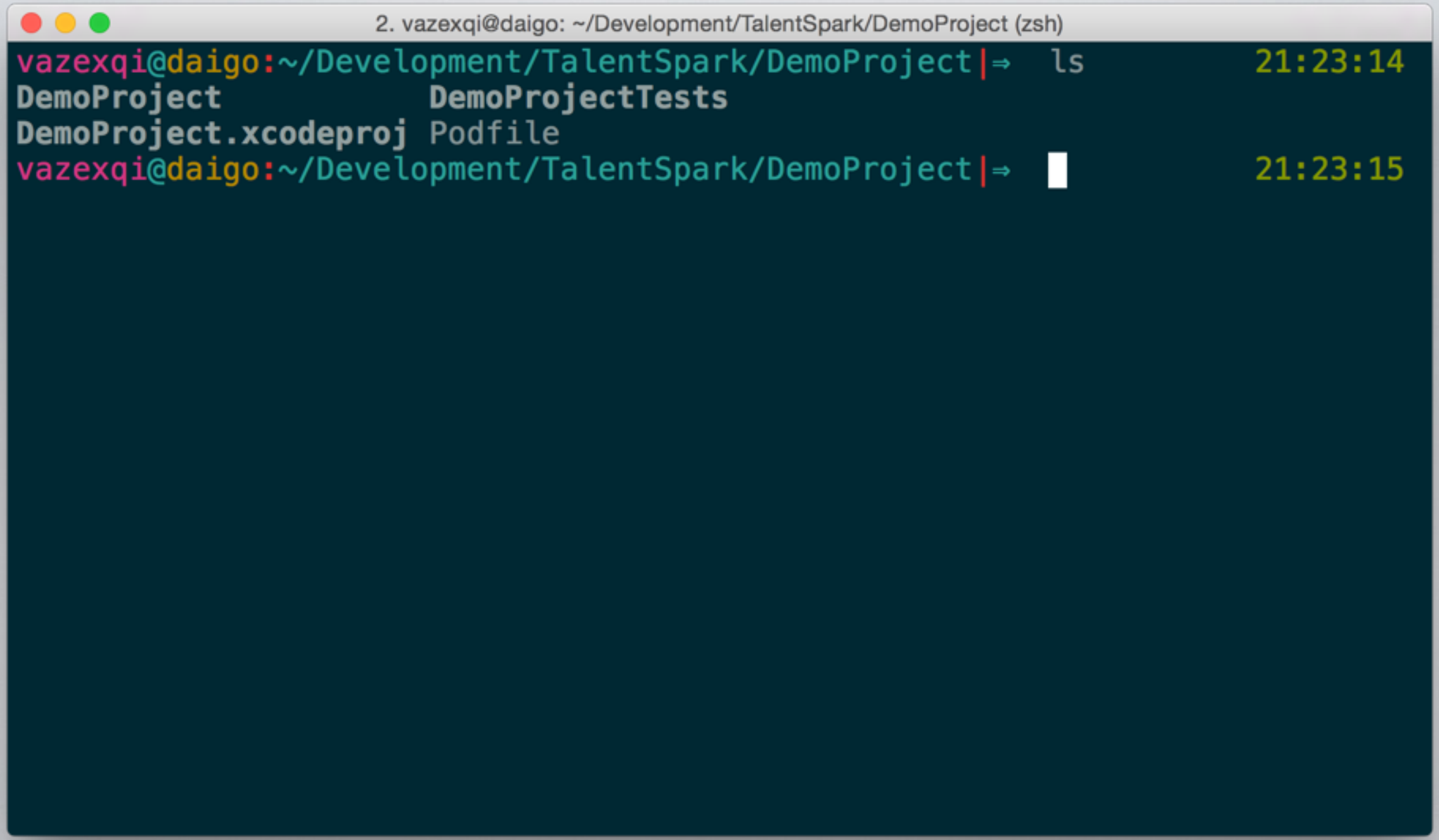
XCODE


Edit
View
Find
Navigate
Editor
Product
Debug
Source Control
Window
Help
Search documentation
Search
Documentation and API Referen
Xcode Overview
Release Notes
What's New in Xcode
Quick Help for Selected Item
Search Documentation for Sele
At a Glance
Use the App Store app on your Mac to download Xcode. It's free. After you download Xco
Launchpad, where you can click the icon for Xcode to launch it.

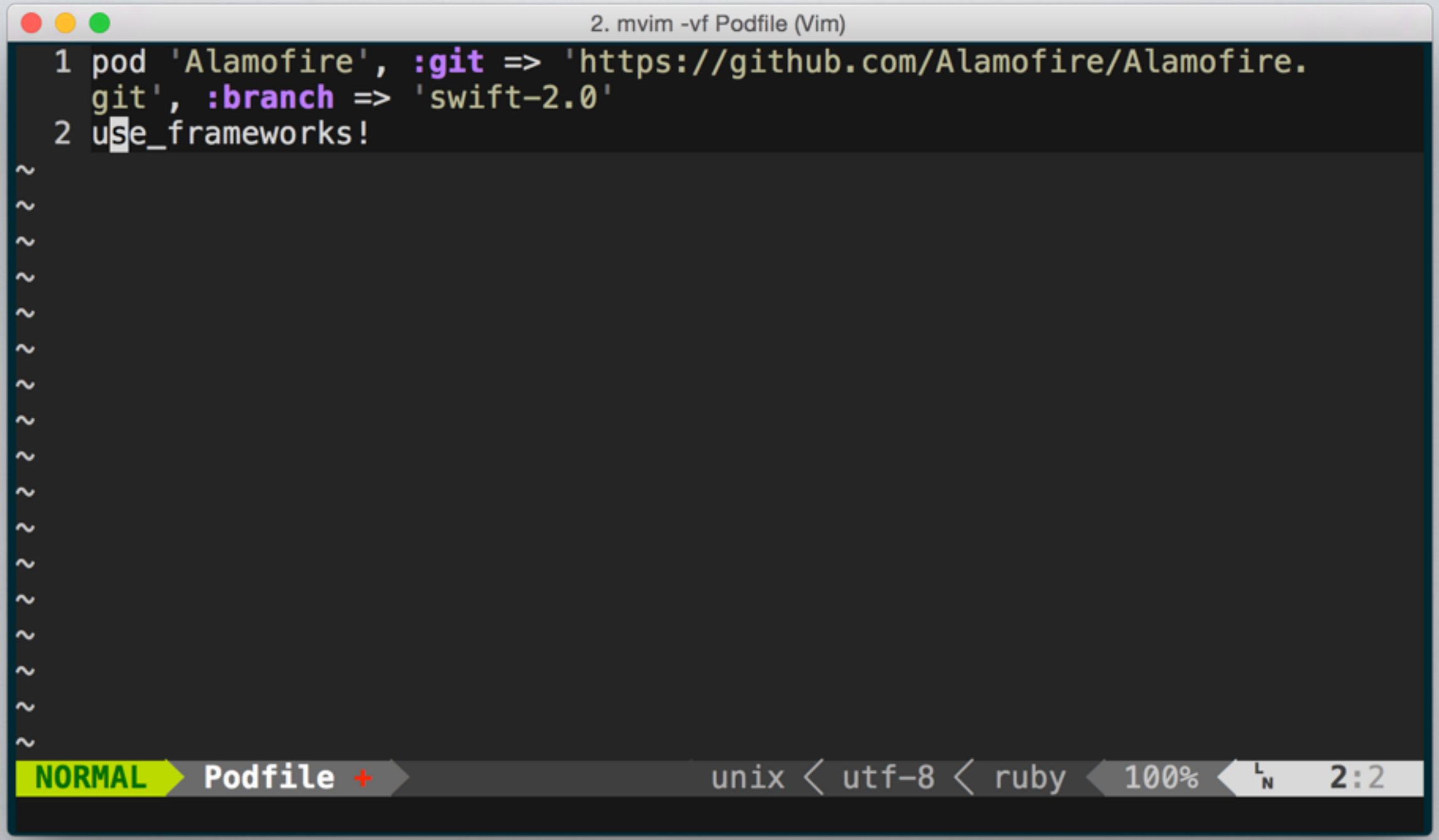
CHEAT SHEET


www.git-tower.com/blog/xcode-cheat-
fournova
XCODE CHEAT SHEET
presented by TOWER › Version control with Git - made easy
SEARCH
Find in File ⌘ F
Find & Replace in File ⌘ ⌥ F
Find in Project ⌘ ⇧ F
Find & Replace in Project ⌘ ⌥ ⇧ F
TABS
New Tab ⌘ T
Previous Tab ⌘ }
Next Tab ⌘ {
NAVIGATION
Open Quickly ⌘ ⇧ O
Move Focus to Editor ⌘ J
Next Counterpart ⌘ ^ UP
Previous Counterpart ⌘ ^ DOWN
Next Recent File ⌘ ^ RIGHT
Previous Recent File ⌘ ^ LEFT
Definition of Symbol ⌘ click
Go to Line ⌘ L
Fold Section ⌘ ⌥ LEFT
Unfold Section ⌘ ⌥ RIGHT
EDITING
Show Assistant Editor ⌘ ⌥ RETURN
Hide Assistant Editor ⌘ RETURN
Toggle Completions ^ SPACE
Edit All in Scope ⌘ ^ E
Indent Selection ⌘]
Outdent Selection ⌘ [
Fix Indentation ^ i
Comment / Uncomment ⌘ /
Move Line Up ⌘ ⌥ [
Move Line Down ⌘ ⌥]

TERMINAL


2. vazexqi@daigo: ~/Development/TalentSpark/DemoProject (zsh)
vazexqi@daigo:~/Development/TalentSpark/DemoProject|⇒ ls 21:23:14
DemoProject DemoProjectTests
DemoProject.xcodeproj Podfile
vazexqi@daigo:~/Development/TalentSpark/DemoProject|⇒ 21:23:15

TERMINAL

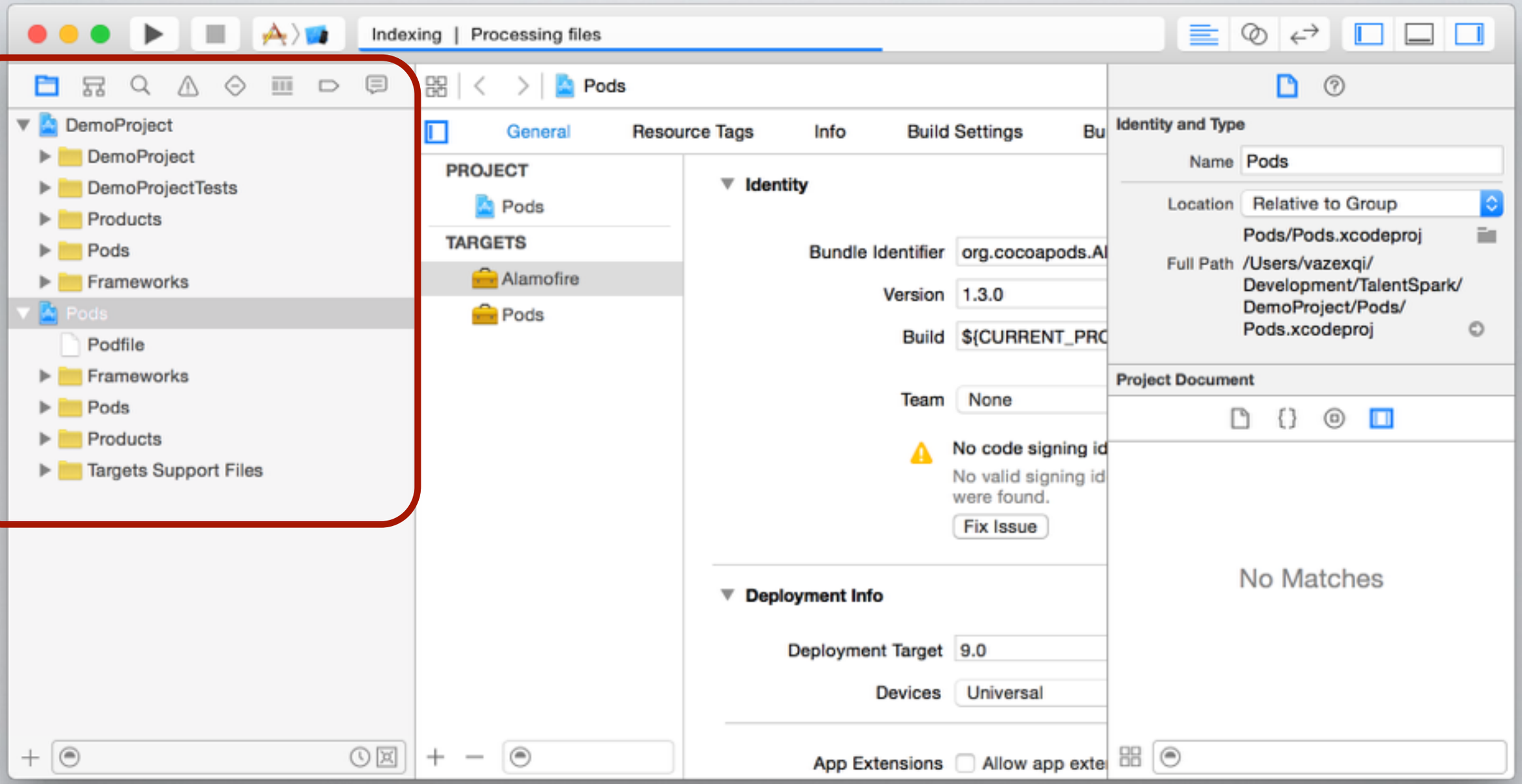

2. mvim -vf Podfile (Vim)
1 pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.
git', :branch => 'swift-2.0'
2 use_frameworks!
NORMAL Podfile + unix < utf-8 < ruby 100% LN 2:2

TERMINAL

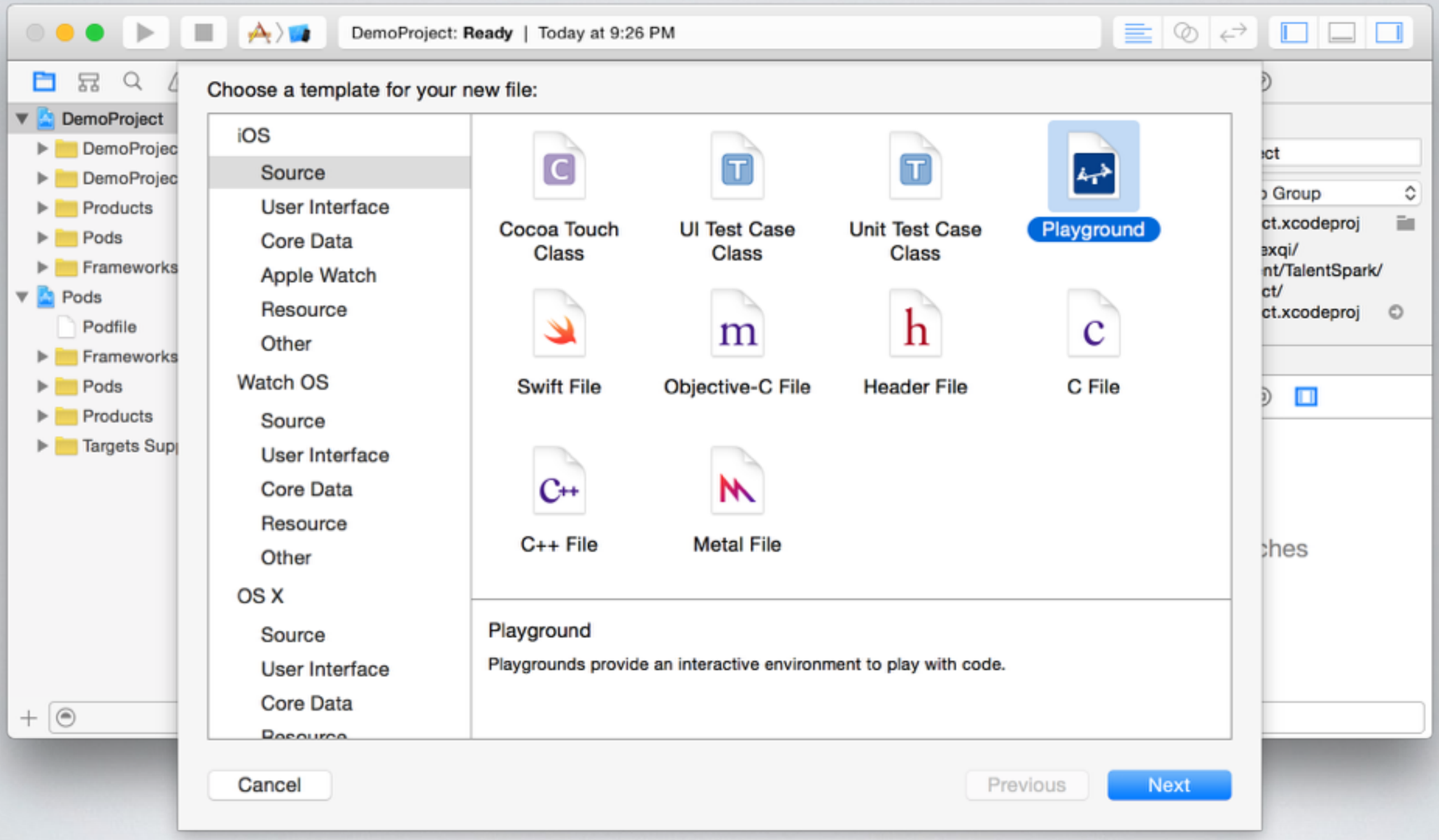
2. vazexqi@daigo: ~/Development/TalentSpark/DemoProject (zsh)

```
vazexqi@daigo:~/Development/TalentSpark/DemoProject|⇒  ls          21:23:14
DemoProject              DemoProjectTests
DemoProject.xcodeproj Podfile
vazexqi@daigo:~/Development/TalentSpark/DemoProject|⇒  pod install
Updating local specs repositories
Analyzing dependencies
Pre-downloading: `Alamofire` from `https://github.com/Alamofire/Alamofire.g
it`, branch `swift-2.0`
Downloading dependencies
Installing Alamofire (1.3.0)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `DemoProject.xcworkspac
e` for this project from now on.
Sending stats
vazexqi@daigo:~/Development/TalentSpark/DemoProject|⇒                21:23:45
```

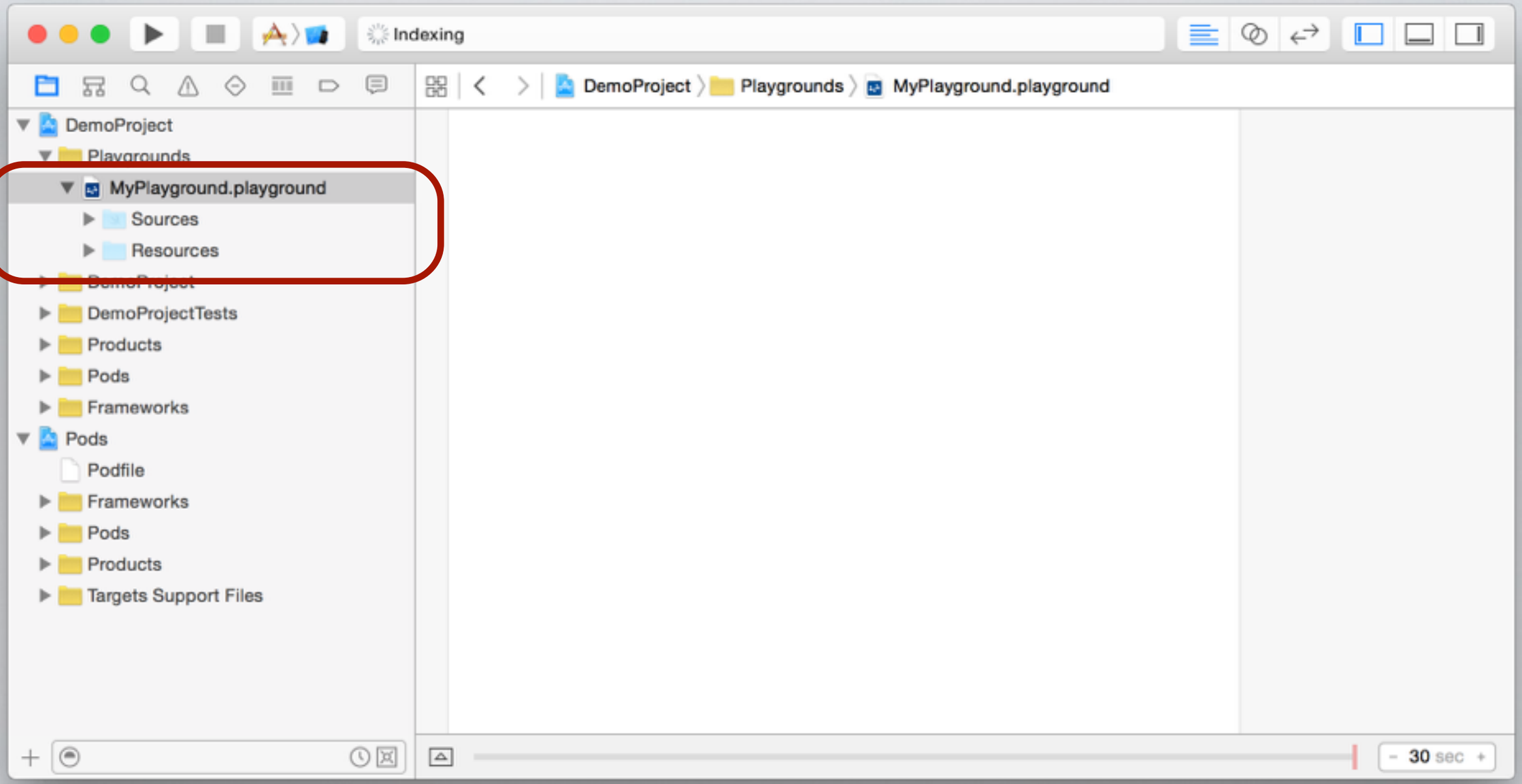
XCODE


Indexing | Processing files
DemoProject
DemoProject
DemoProjectTests
Products
Pods
Frameworks
Pods
Podfile
Frameworks
Pods
Products
Targets Support Files
Pods
General
Resource Tags
Info
Build Settings
Bu
PROJECT
Pods
TARGETS
Alamofire
Pods
Identity
Bundle Identifier
org.cocoapods.Al
Version
1.3.0
Build
${CURRENT_PRO
Team
None
No code signing id
No valid signing id
were found.
Fix Issue
Deployment Info
Deployment Target
9.0
Devices
Universal
App Extensions
Allow app exte
Identity and Type
Name
Pods
Location
Relative to Group
Pods/Pods.xcodeproj
Full Path
/Users/vazexqi/
Development/TalentSpark/
DemoProject/Pods/
Pods.xcodeproj
Project Document
No Matches

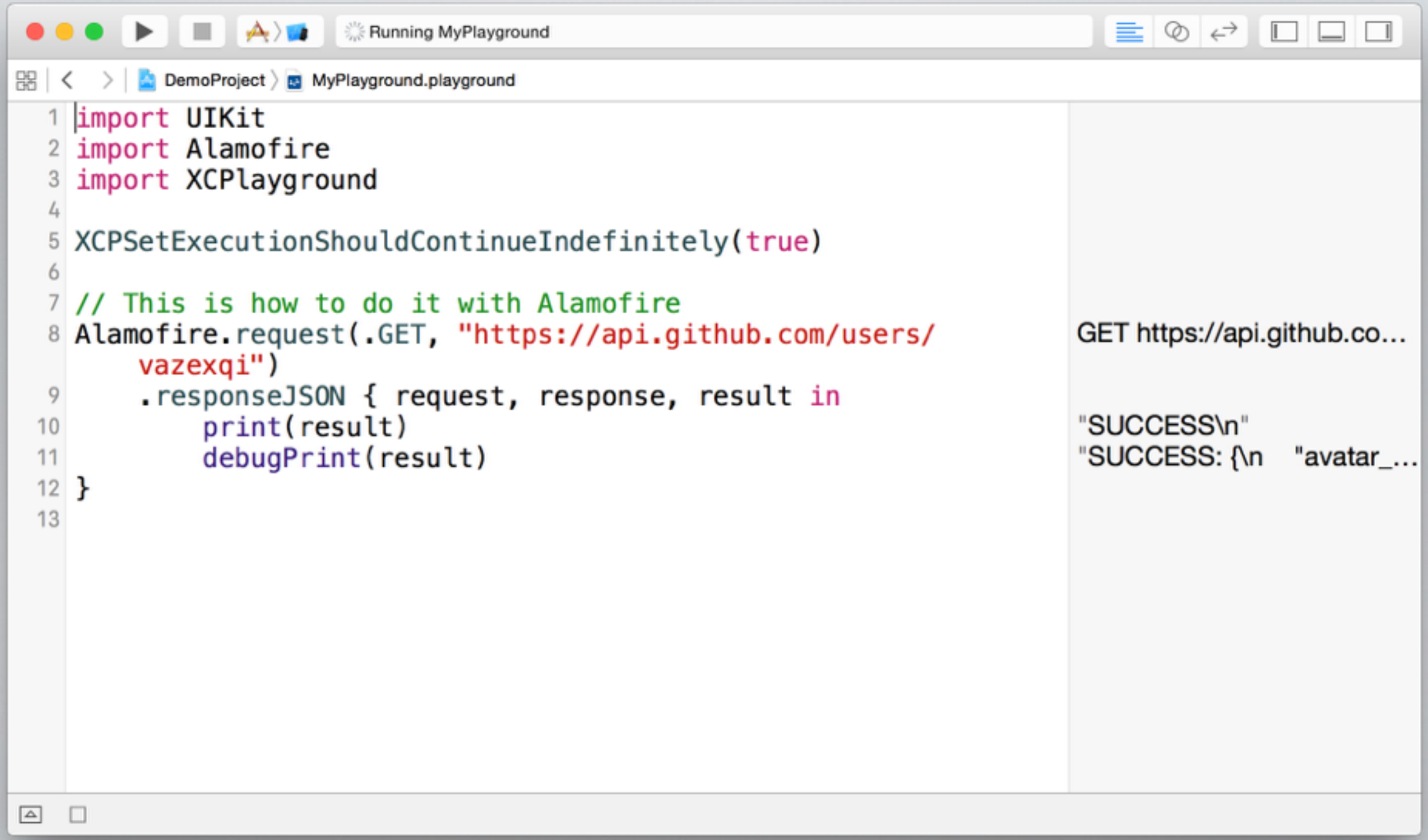
XCODE


DemoProject: Ready | Today at 9:26 PM
Choose a template for your new file:
iOS
Source
User Interface
Core Data
Apple Watch
Resource
Other
Watch OS
Source
User Interface
Core Data
Resource
Other
OS X
Source
User Interface
Core Data
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Playground
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
Playground
Playgrounds provide an interactive environment to play with code.
Cancel
Previous
Next
DemoProject
Pods
Podfile
Frameworks
Products
Targets Supp

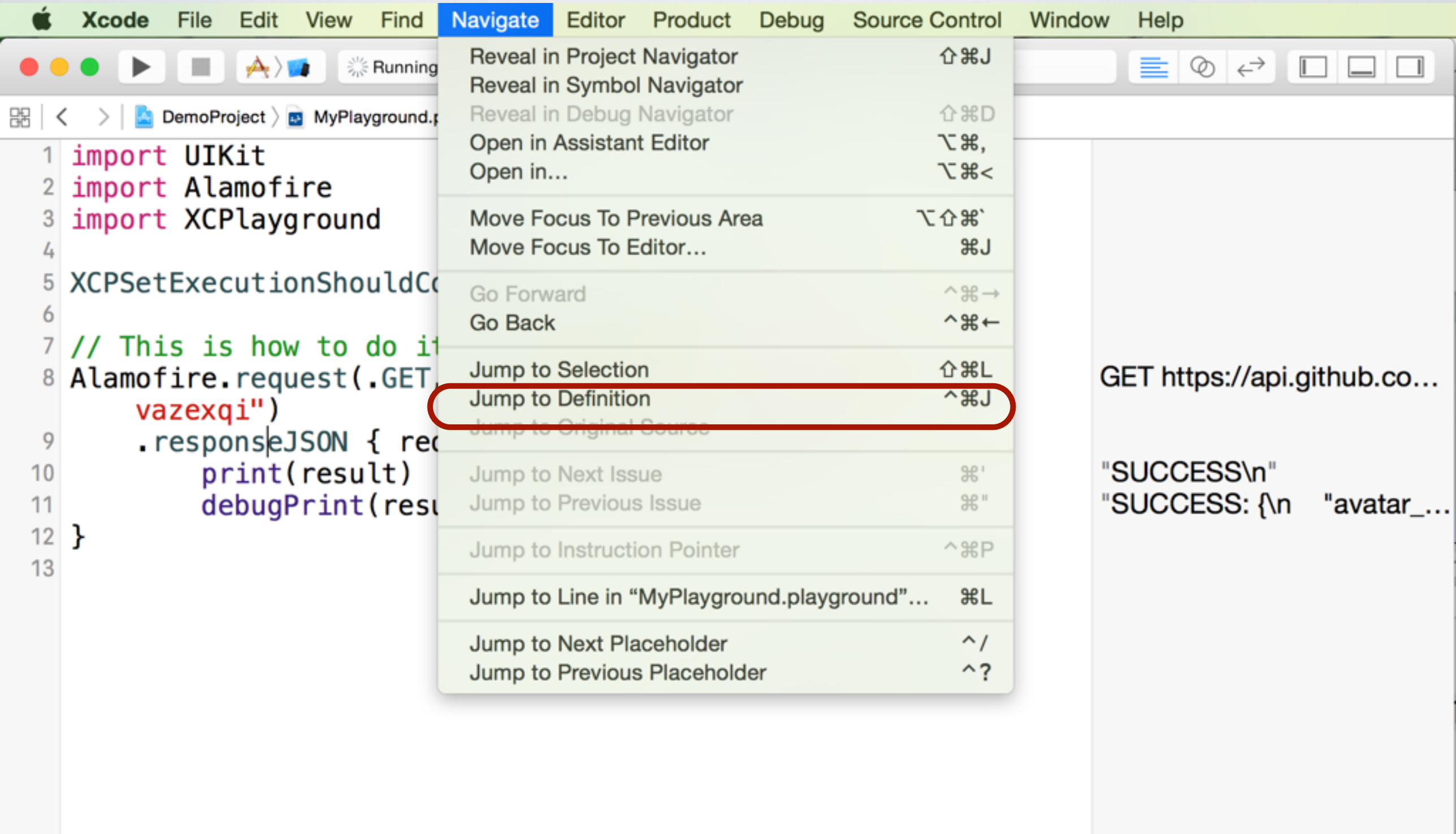
XCODE


Indexing
DemoProject
Playgrounds
MyPlayground.playground
DemoProject
Playgrounds
MyPlayground.playground
Sources
Resources
DemoProjectTests
Products
Pods
Frameworks
Pods
Podfile
Frameworks
Pods
Products
Targets Support Files
30 sec

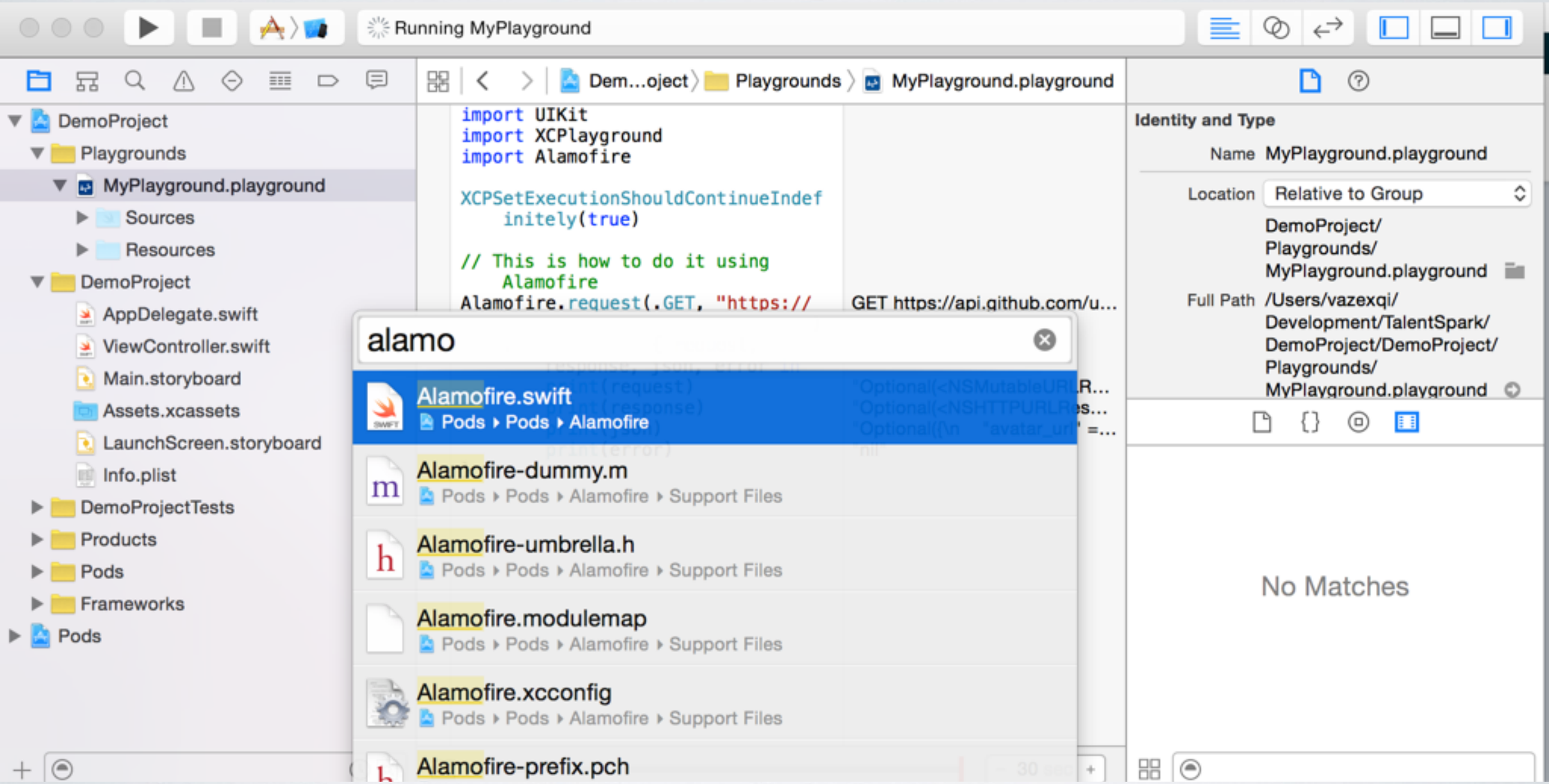
XCODE


Running MyPlayground
DemoProject
MyPlayground.playground
import UIKit
import Alamofire
import XCPlayground
XCPSetExecutionShouldContinueIndefinitely(true)
// This is how to do it with Alamofire
Alamofire.request(.GET, "https://api.github.com/users/vazexqi")
.responseJSON { request, response, result in
print(result)
debugPrint(result)
}
GET https://api.github.co...
"SUCCESS\n"
"SUCCESS: {\n "avatar_...

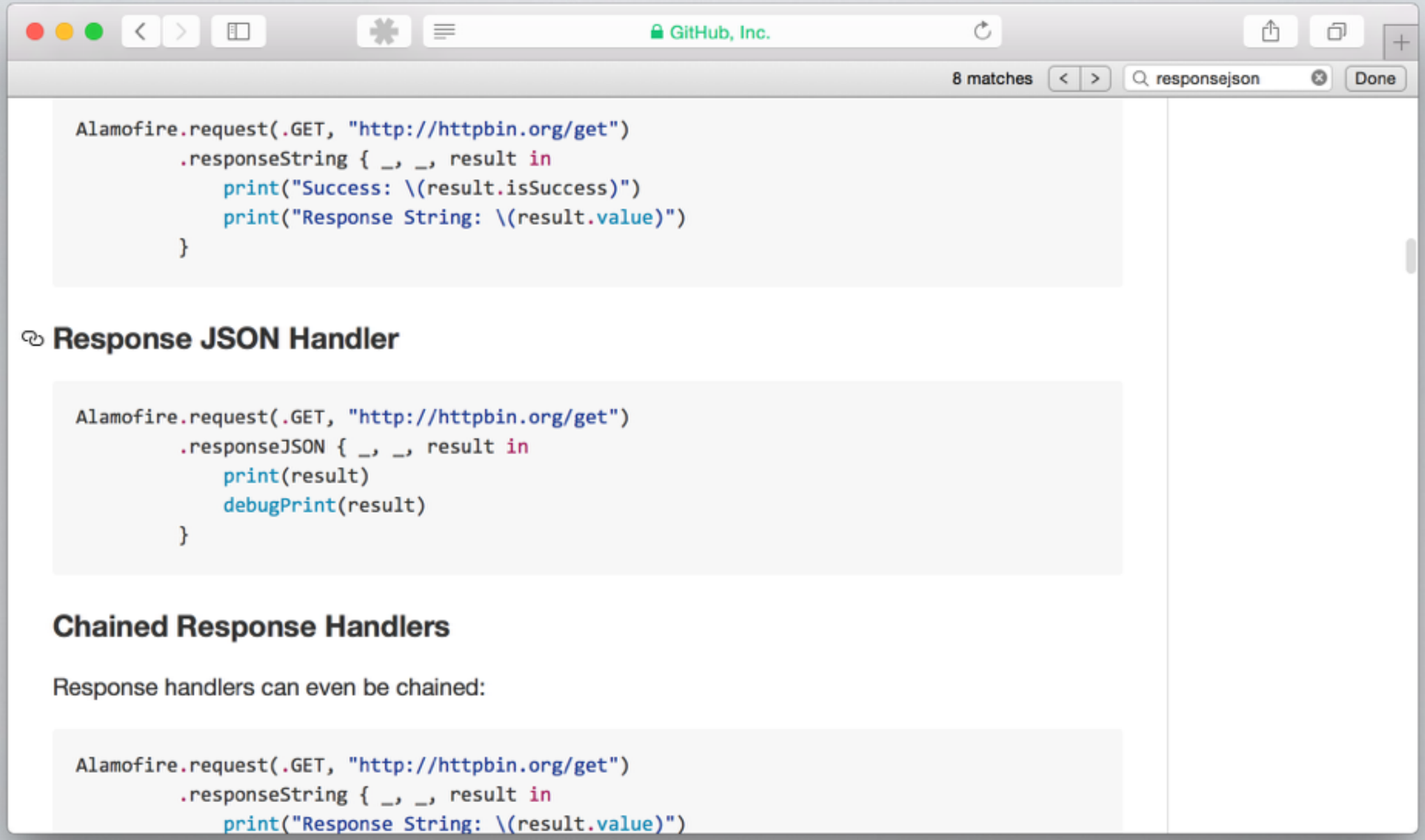
XCODE


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Running
DemoProject
MyPlayground.p
1 import UIKit
2 import Alamofire
3 import XCPlayground
4
5 XCPSetExecutionShouldC
6
7 // This is how to do it
8 Alamofire.request(.GET,
vazexqi")
9 .responseJSON { rec
10 print(result)
11 debugPrint(resu
12 }
13
Reveal in Project Navigator ⇧⌘J
Reveal in Symbol Navigator
Reveal in Debug Navigator ⇧⌘D
Open in Assistant Editor ⌥⌘,
Open in... ⌥⌘<
Move Focus To Previous Area ⌥⇧⌘`
Move Focus To Editor... ⌘J
Go Forward ^⌘→
Go Back ^⌘←
Jump to Selection ⇧⌘L
Jump to Definition ^⌘J
Jump to Next Issue ⌘'
Jump to Previous Issue ⌘"
Jump to Instruction Pointer ^⌘P
Jump to Line in “MyPlayground.playground”... ⌘L
Jump to Next Placeholder ^/
Jump to Previous Placeholder ^?
GET https://api.github.co...
"SUCCESS\n"
"SUCCESS: {\n "avatar_...

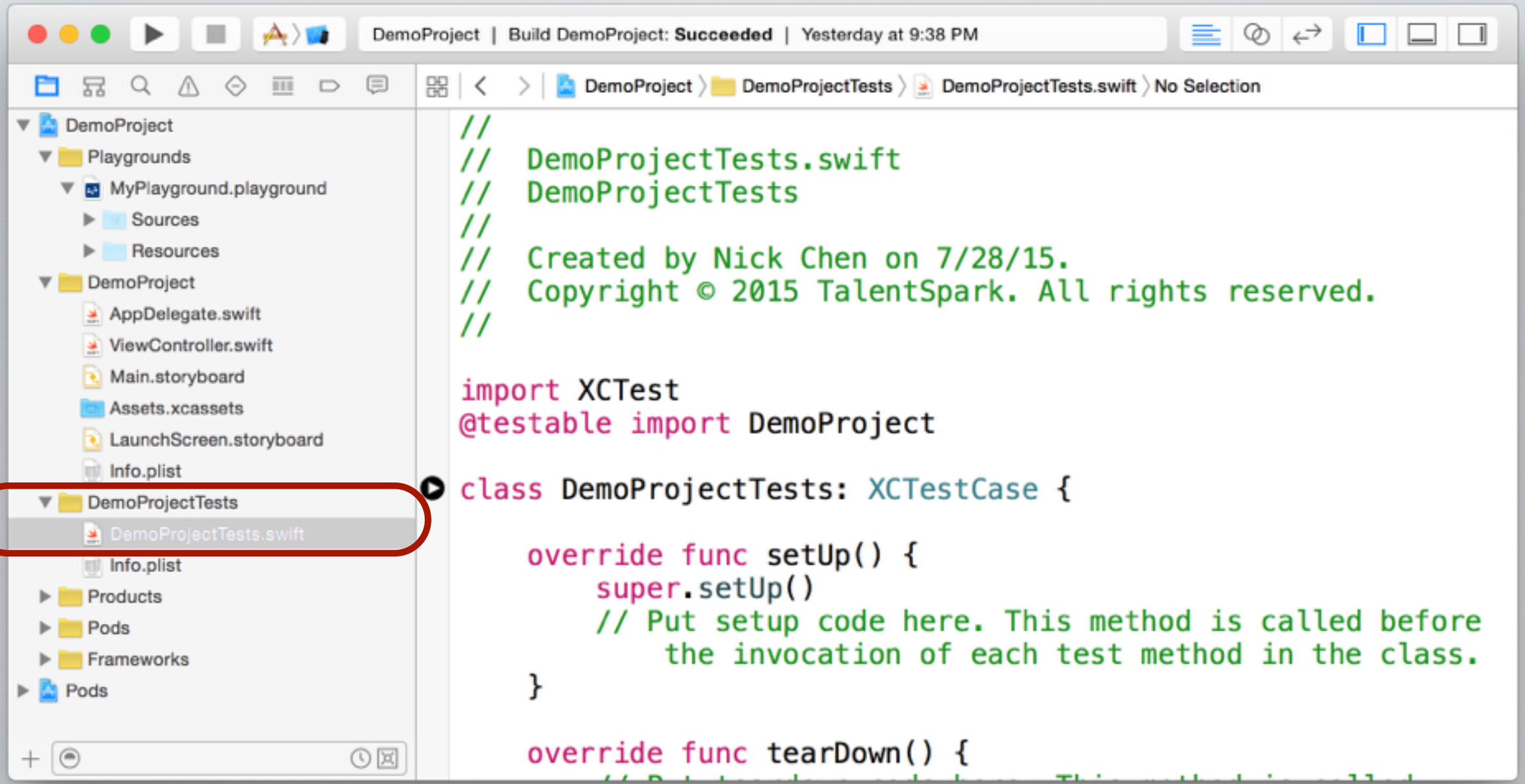
XCODE


Running MyPlayground
Dem...oject
Playgrounds
MyPlayground.playground
DemoProject
Playgrounds
MyPlayground.playground
Sources
Resources
DemoProject
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
DemoProjectTests
Products
Pods
Frameworks
Pods
import UIKit
import XCPlayground
import Alamofire
XCPSetExecutionShouldContinueIndef
initely(true)
// This is how to do it using
Alamofire
Alamofire.request(.GET, "https://
GET https://api.github.com/u...
alamo
Alamofire.swift
Pods ▸ Pods ▸ Alamofire
Alamofire-dummy.m
Pods ▸ Pods ▸ Alamofire ▸ Support Files
Alamofire-umbrella.h
Pods ▸ Pods ▸ Alamofire ▸ Support Files
Alamofire.modulemap
Pods ▸ Pods ▸ Alamofire ▸ Support Files
Alamofire.xcconfig
Pods ▸ Pods ▸ Alamofire ▸ Support Files
Alamofire-prefix.pch
Identity and Type
Name MyPlayground.playground
Location Relative to Group
DemoProject/
Playgrounds/
MyPlayground.playground
Full Path /Users/vazexqi/
Development/TalentSpark/
DemoProject/DemoProject/
Playgrounds/
MyPlayground.playground
No Matches

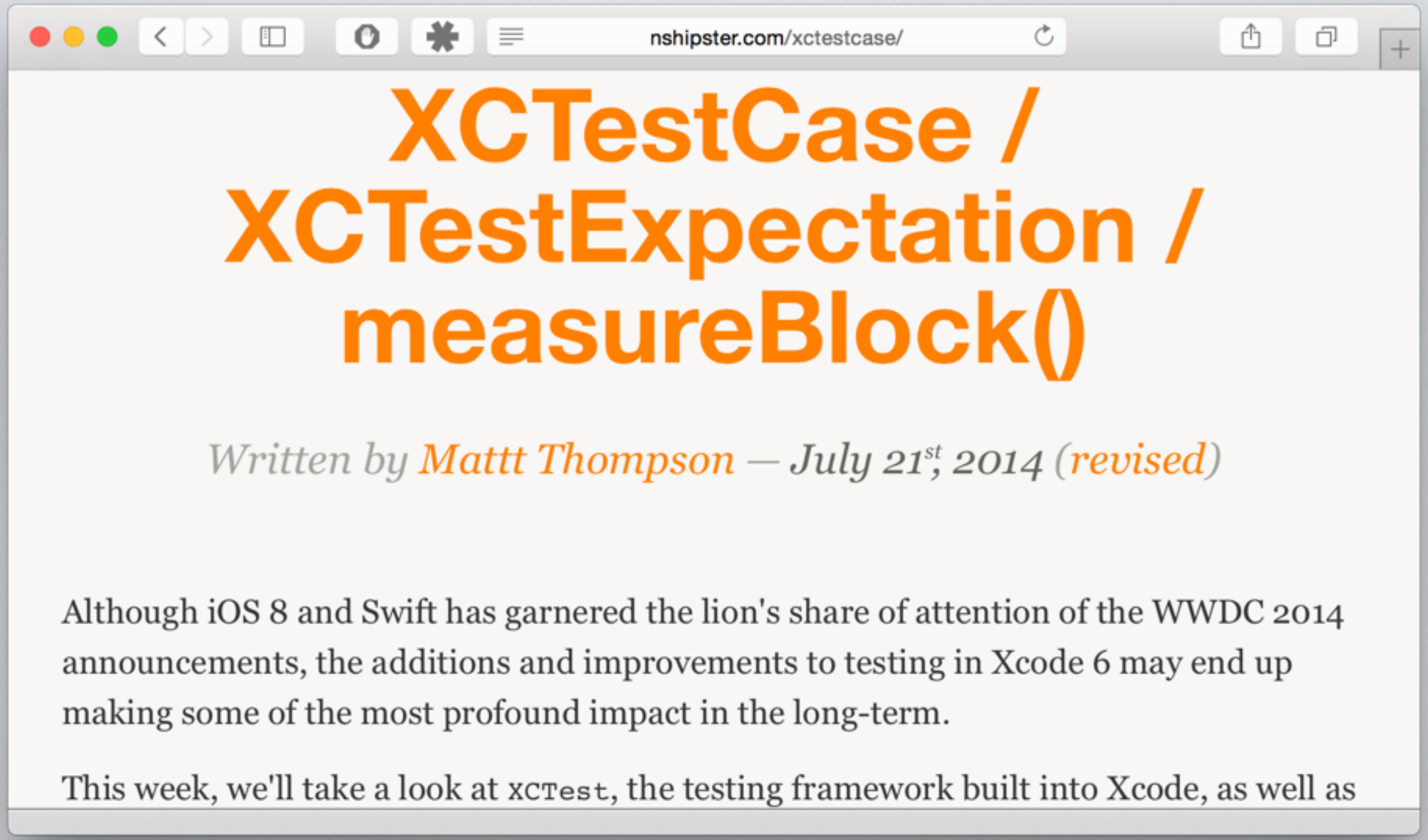
ALAMOFIRE


GitHub, Inc.
8 matches
responsejson
Done
Alamofire.request(.GET, "http://httpbin.org/get")
.responseString { _, _, result in
print("Success: \(result.isSuccess)")
print("Response String: \(result.value)")
}
Response JSON Handler
Alamofire.request(.GET, "http://httpbin.org/get")
.responseJSON { _, _, result in
print(result)
debugPrint(result)
}
Chained Response Handlers
Response handlers can even be chained:
Alamofire.request(.GET, "http://httpbin.org/get")
.responseString { _, _, result in
print("Response String: \(result.value)")

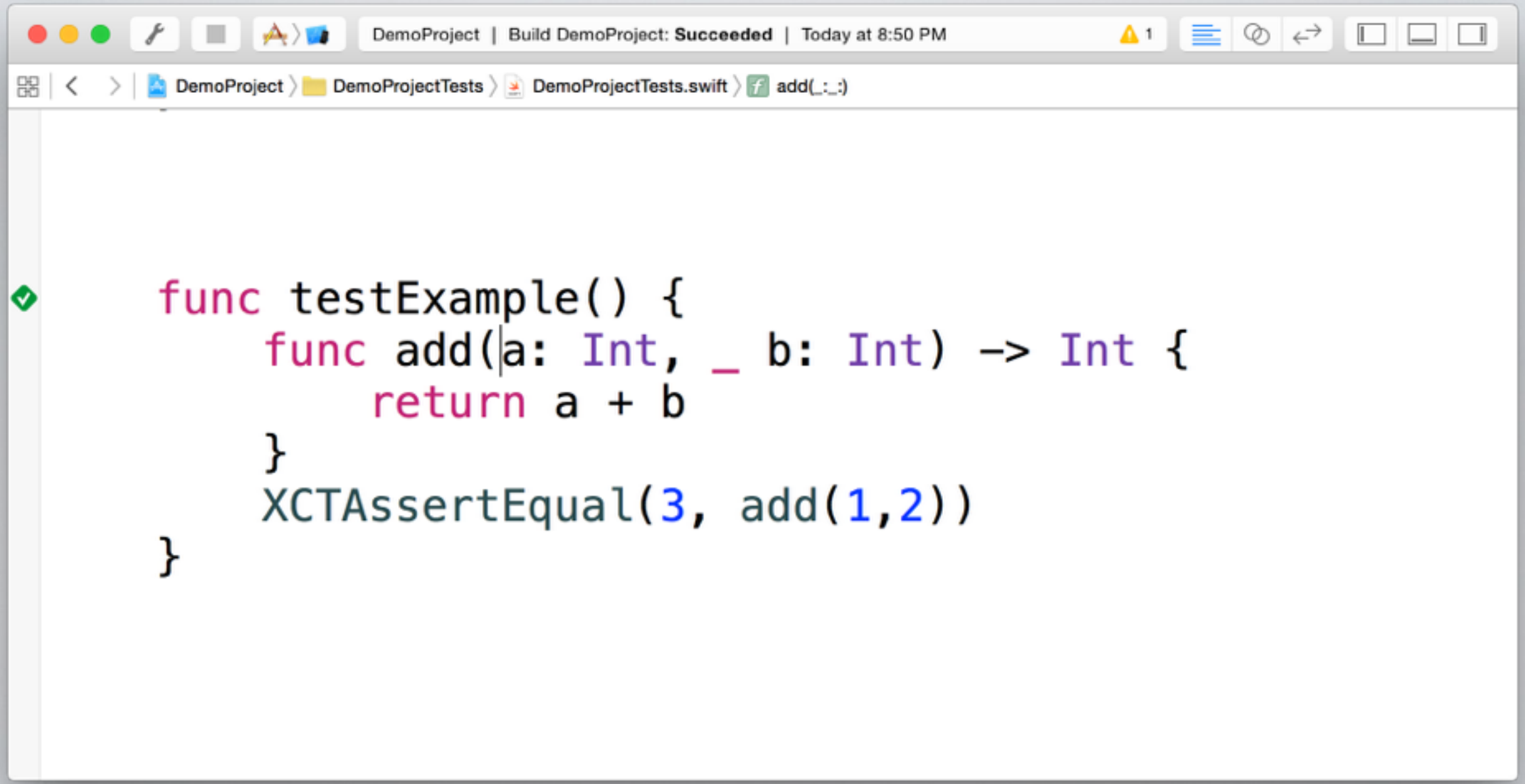
UNIT TESTS


DemoProject | Build DemoProject: Succeeded | Yesterday at 9:38 PM
DemoProject
DemoProjectTests
DemoProjectTests.swift
No Selection
DemoProject
Playgrounds
MyPlayground.playground
Sources
Resources
DemoProject
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
DemoProjectTests
DemoProjectTests.swift
Info.plist
Products
Pods
Frameworks
Pods
//
// DemoProjectTests.swift
// DemoProjectTests
//
// Created by Nick Chen on 7/28/15.
// Copyright © 2015 TalentSpark. All rights reserved.
//
import XCTest
@testable import DemoProject
class DemoProjectTests: XCTestCase {
override func setUp() {
super.setUp()
// Put setup code here. This method is called before
the invocation of each test method in the class.
}
override func tearDown() {

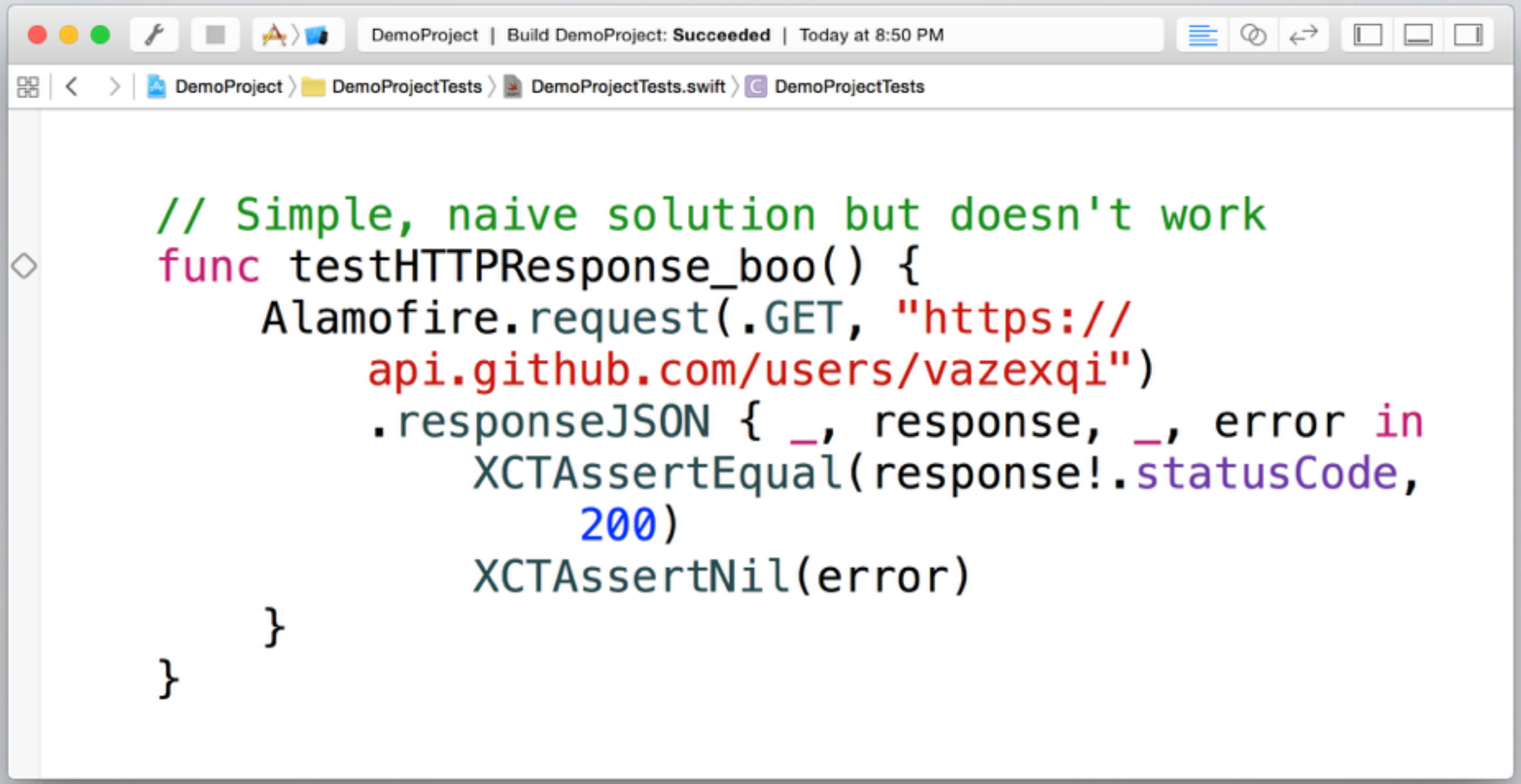
UNIT TESTS


nshipster.com/xctestcase/
XCTestCase / XCTestExpectation / measureBlock()
Written by Mattt Thompson — July 21st, 2014 (revised)
Although iOS 8 and Swift has garnered the lion's share of attention of the WWDC 2014 announcements, the additions and improvements to testing in Xcode 6 may end up making some of the most profound impact in the long-term.
This week, we'll take a look at xctest, the testing framework built into Xcode, as well as

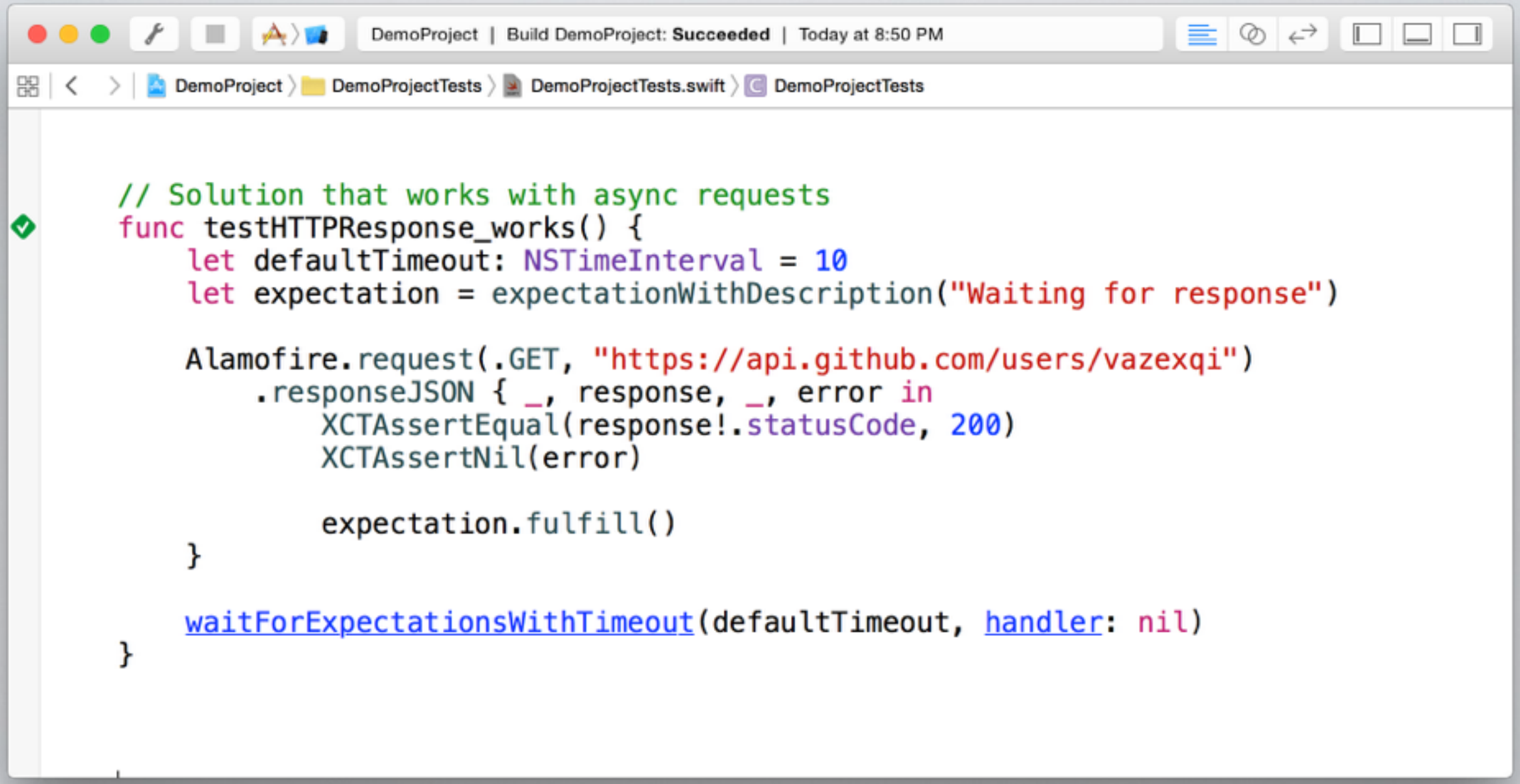
UNIT TESTS


DemoProject | Build DemoProject: Succeeded | Today at 8:50 PM
1
DemoProject
DemoProjectTests
DemoProjectTests.swift
add(_:_:)
func testExample() {
func add(a: Int, _ b: Int) -> Int {
return a + b
}
XCTAssertEqual(3, add(1,2))
}

UNIT TESTS


DemoProject | Build DemoProject: Succeeded | Today at 8:50 PM
DemoProject
DemoProjectTests
DemoProjectTests.swift
DemoProjectTests
// Simple, naive solution but doesn't work
func testHTTPResponse_boo() {
Alamofire.request(.GET, "https://
api.github.com/users/vazexqi")
.responseJSON { _, response, _, error in
XCTAssertEqual(response!.statusCode,
200)
XCTAssertNil(error)
}
}

UNIT TESTS


DemoProject | Build DemoProject: Succeeded | Today at 8:50 PM
DemoProject
DemoProjectTests
DemoProjectTests.swift
DemoProjectTests
// Solution that works with async requests
func testHTTPResponse_works() {
let defaultTimeout: NSTimeInterval = 10
let expectation = expectationWithDescription("Waiting for response")
Alamofire.request(.GET, "https://api.github.com/users/vazexqi")
.responseJSON { _, response, _, error in
XCTAssertEqual(response!.statusCode, 200)
XCTAssertNil(error)
expectation.fulfill()
}
waitForExpectationsWithTimeout(defaultTimeout, handler: nil)
}

PROJECT 1

- Out today
- Due one week later on Sept 11, 2015
- Push to your GitHub account under Project 1 and write a note to Piazza.