

CODECRAFTERS

**Build Better
Apps Faster**

What we'll cover

- Stateful ViewControllers. 
- Size classes & Autolayout. 
- Networking and Promises. 
 - Forms 
 - CoreData 
 - Soft Skills 

And of course...

Tests

We'll build:

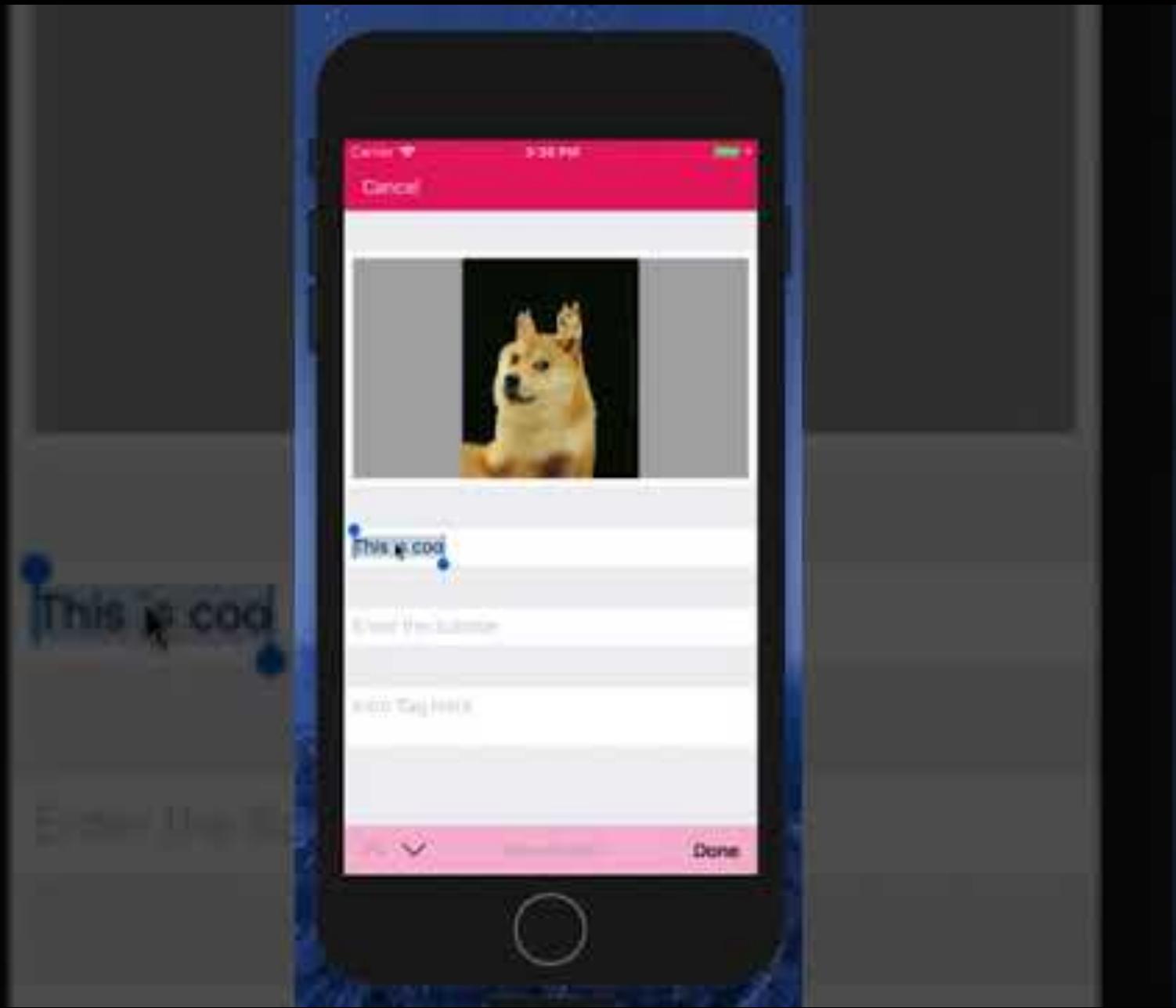
A Wallet

A Crypto Wallet?

A Crypto Wallet

A GIF Wallet

Gif Wallet



- Store GIFs in a handy way to share
- Integration with Giphy to find new GIFs
- iPad support

Schedule:

Friday

Stateful VC

Autolayout

Size Classes

Saturday

Networking

Promises

Forms

Sunday

CoreData

UITests

Soft Skills

...But first

A close-up photograph of a person's hands holding a dark brown leather belt. The belt features several silver-toned metal buckles of different styles. The hands are positioned as if presenting or adjusting the belt.

Bootstrapping

Goals of Bootstrapping:

- Separation of concerns
- Dependency Management
- Schemes for Development and AppStore
 - Tests setup
 - Continuous Integration

GitHub, Inc.

Search GitHub Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner Repository name
 piercifani /

Great repository names are short and memorable. Need inspiration? How about [silver-fortnight](#).

Description (optional)

 Public
Anyone can see this repository. You choose who can commit.

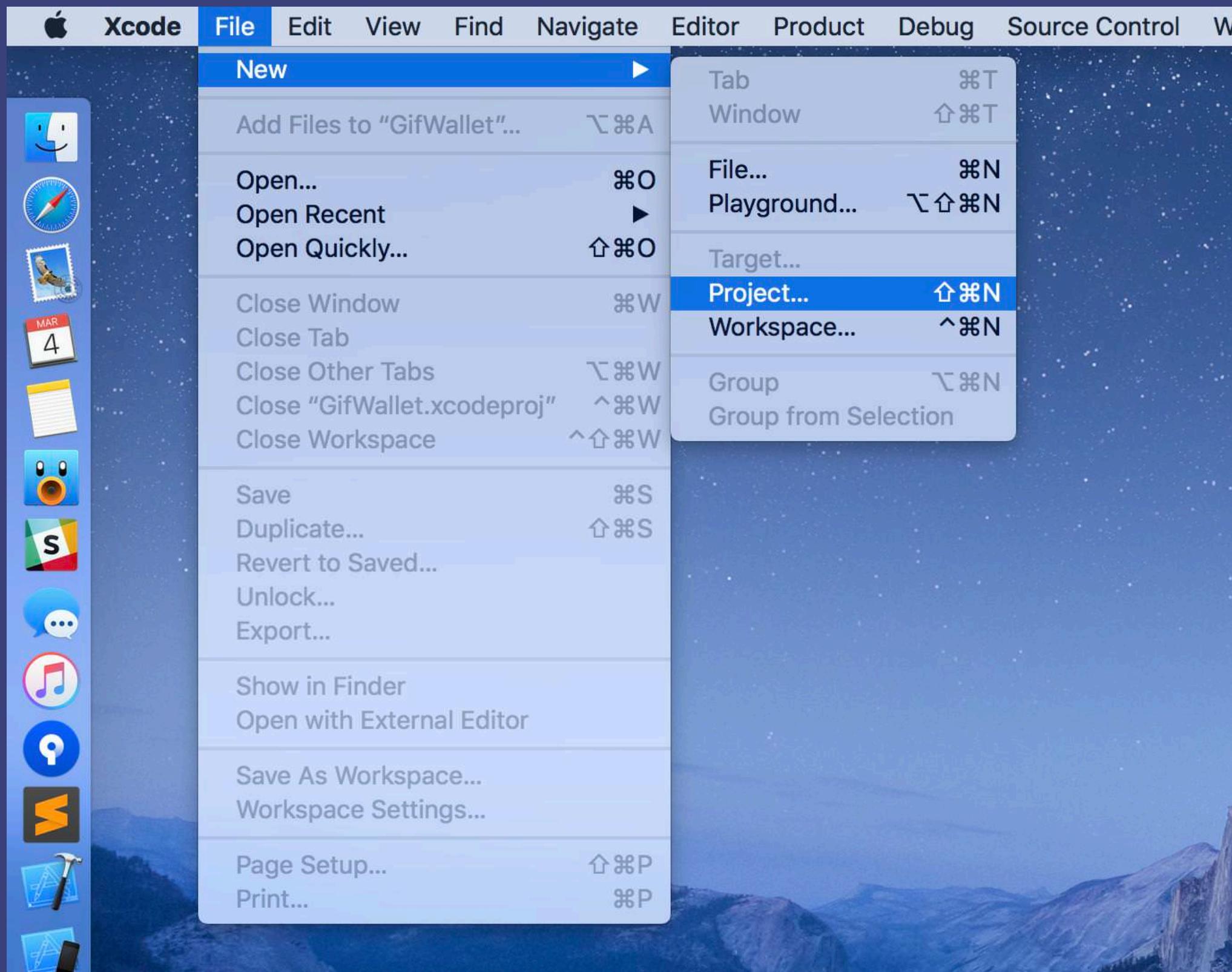
 Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: [None](#) | Add a license: [None](#) 

[Create repository](#)

© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)



Separation of concerns

UI and Model layers should be separated into modules.

- Easier testing
- Faster compilation
- Reusability

UIKit → UI Module
Foundation → Foundation Module

What's a module?

A Module¹:

- Encapsulates code and data.
- Has an interface (API)
- Easily deployed

¹ Modular programming from [Wikipedia](#).

UIKit → UI Module → App
Foundation → Foundation Module → Framework

However...

https://itunes.apple.com/es/developer/santander-uk-plc/id482

Mac iPad iPhone Watch TV Music Soporte Q

iTunes Preview

Santander UK plc [View in iTunes](#)

Open iTunes to buy and download apps.

iPhone Apps

1.  Santander Personal Bank... [View in iTunes](#)
2.  Spendlytics [View in iTunes](#)
3.  Business Banking [View in iTunes](#)

iPad Apps

1.  Santander Personal Bank... [View in iTunes](#)
2.  Business Banking [View in iTunes](#)

App Store en Facebook y Twitter

 Descubre y comparte nuevas apps. Síguenos en [@AppStore](#).

iTunes en Facebook

 Descubre y comparte música, películas, libros y mucho más.

iTunes en Twitter

 Síguenos en [@iTunes](#) y descubre la música que más nos gusta.

itunes.apple.com

Mac iPad iPhone Watch TV Music Soporte

iTunes Preview

Banco Santander [View in iTunes](#)

Open iTunes to buy and download apps.

iPhone Apps

1. Santander View in iTunes	2. Santander Wallet View in iTunes	3. Santander Money Plan View in iTunes	4. Santander Empresas View in iTunes	5. Criptocalculadora View in iTunes
6. Santander Shareholders ... View in iTunes	7. 1 2 3 Mini View in iTunes	8. Santander Broker View in iTunes	9. Santander Watch View in iTunes	10. Santander Agro View in iTunes
11. Santander Grants View in iTunes	12. Renting Santander Bansal... View in iTunes	13. Santander Mi Comercio View in iTunes	14. Confirming Santander View in iTunes	15. Santander Justicia View in iTunes
16. FACTORING digital	17. FACTORING digital			

In this cases:

UIKit → UI Module → Framework

Foundation → Foundation Module → Framework

PROJECT



GifWallet

TARGETS



GifWallet



GifWalletTests



GifWalletKit



GifWalletKitTests

App Target

- Cells and Views
- ViewController
- Interactors

Framework Target

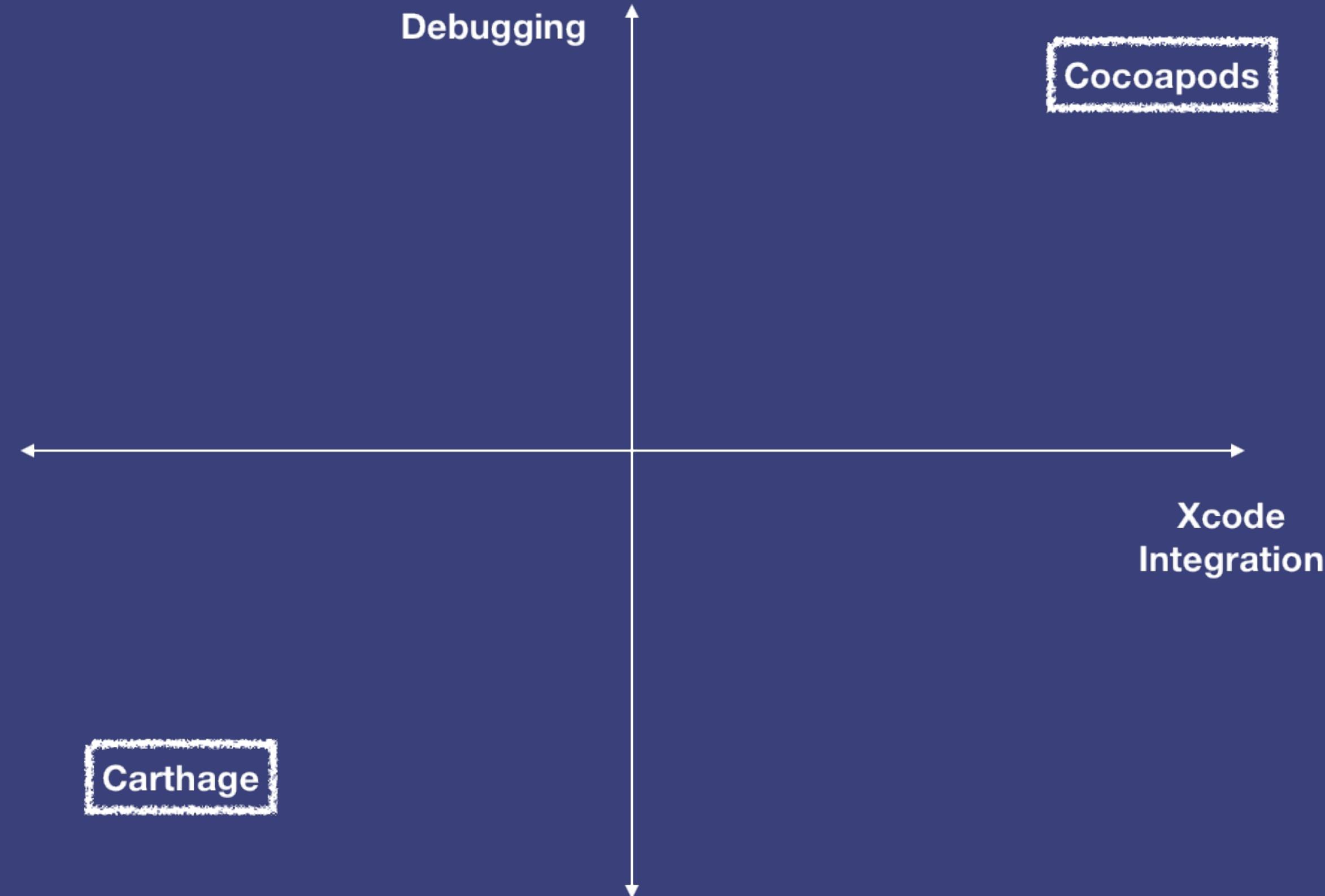
- Form validation
- Data store
- APIClient

Dependency Management

Dependency Management

The de-facto standard is Cocoapods

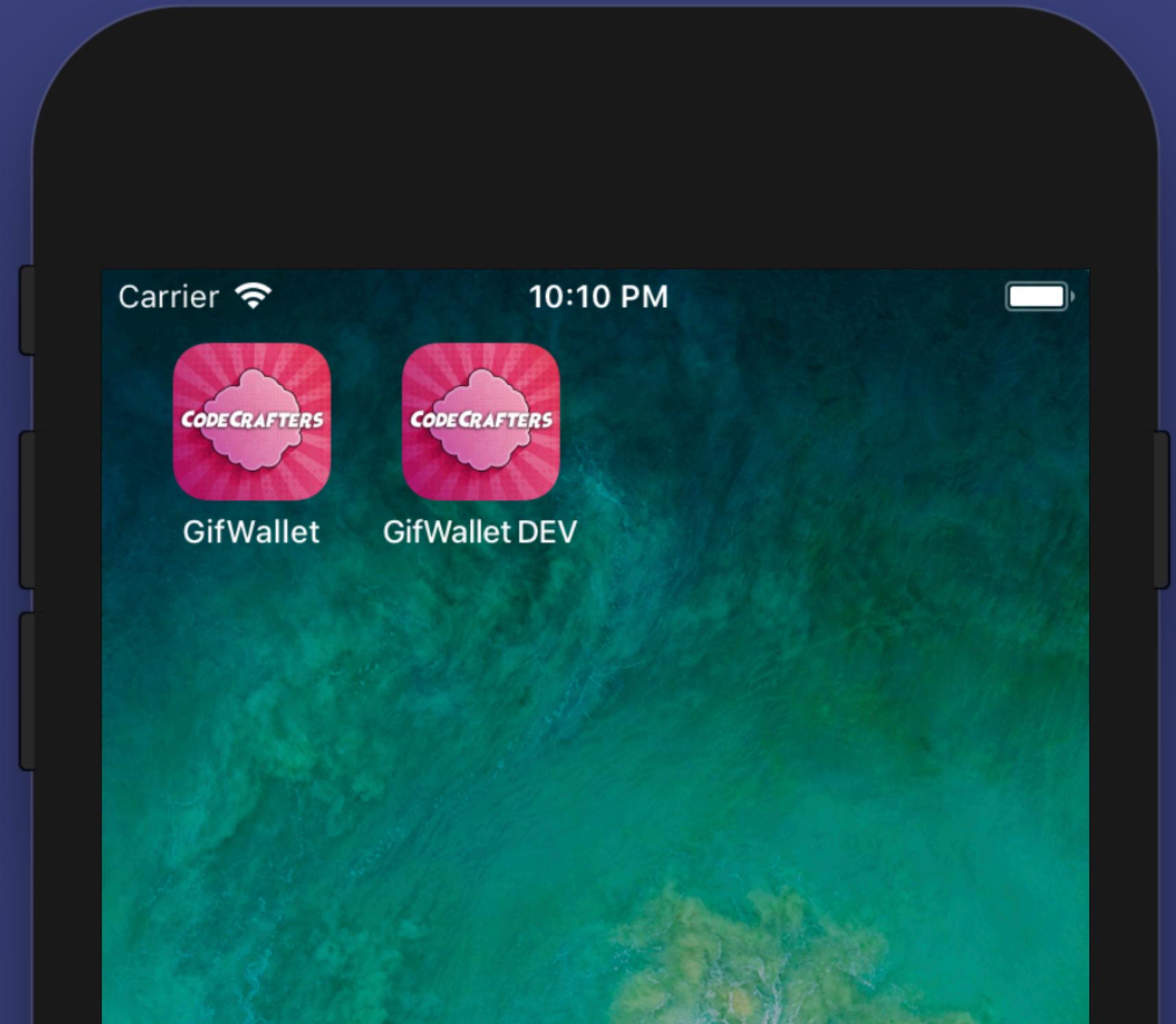
However, Carthage is also a challenger.



Tradeoffs:

- Add /Pods to repo.
- Xcode integration.
- Too many frameworks.

Schemes for Development and AppStore



We shouldn't use the two target approach

- It 2x xcodeproj file's size.
- It leads to errors when adding new files.
 - It feels dirty.

Instead, we should use schemes

- With schemes we can get the same result.
- We can parametrize what we need using xcconfig
 - It's predictable.

XCConfig



- Allows us to parametrize:
Bundle ID, Entitlements,
Compilation Flags...
- Can be set per scheme.
- Integrates with Cocoapods.

```
#include "Pods/Target Support Files/Pods- GifWallet/Pods-  
GifWallet.release.xcconfig" #include "Pods/Target Support Files/Pods-  
GifWallet/Pods-  
GifWallet.debug.xcconfig"  
  
// App Name  
APP_DISPLAY_NAME = GifWallet // App Name  
APP_DISPLAY_NAME = GifWallet DEV  
  
// Code Signing  
PRODUCT_BUNDLE_IDENTIFIER = com.codecrafters.gifwallet // Code Signing  
PRODUCT_BUNDLE_IDENTIFIER = com.codecrafters.gifwallet.dev
```

▼ Configurations

Name	Based on Configuration File
Debug	2 Configurations Set
GifWallet	None ▾
GifWallet	Development ▾
GifWalletTests	None ▾
GifWalletKit	Pods-GifWalletKit.deb... ▾
GifWalletKitTests	None ▾
Release	2 Configurations Set
GifWallet	None ▾
GifWallet	AppStore ▾
GifWalletTests	None ▾
GifWalletKit	Pods-GifWalletKit.rele... ▾
GifWalletKitTests	None ▾

Tests setup

Tests setup

- Tests should not launch your app's logic.
 - Integrate with Snapshot testing.
- Separate App Tests from Model tests

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate, UISplitViewControllerDelegate {

    var window: UIWindow?

    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions options: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

        self.window = UIWindow(frame: UIScreen.main.bounds)
        defer { self.window?.makeKeyAndVisible() }

        guard NSClassFromString("XCTest") == nil else {
            self.window?.rootViewController = UIViewController()
            return true
        }

        self.window?.rootViewController = ...
        return true
    }
}
```

Continuous
integration

Continuous integration

Several options:

- Buddybuild
- Bitrise
- CircleCI
- Travis

Why Bitrise?

Why Bitrise?

- Multiplatform
- Easy setup
- Github integration for PRs
- Apple Developer integration
- Price scales linearly

bitrise.io

Refer your friends for more build time

Dashboard / piercifani / GifWallet bitrise

Builds Workflow Team Code Settings Start/Schedule a Build

Click here to setup deployment to your devices! > HIDE

Try build number or commit message... All triggers All branches All workflows

You have no scheduled builds. To schedule a build, [click here](#) or select the **Start a build** button above.

Today

Preparing ⚡ develop primary
Triggered @ 16:42 ⏱ N/A # 154 5904a73 Add bitrise badge

Success ⚡ develop primary
Triggered @ 15:28 ⏱ 6m 23s # 153 9ca0b17 Merge pull request #21 from CodeCraftersIO/ui-tests UI T...

Success 21 CodeCraftersIO:ui-tests ➔ develop primary

The screenshot shows the Bitrise web interface on a Mac OS X desktop. The top navigation bar includes the Bitrise logo, a 'Refer your friends for more build time' button, and standard browser controls. The main content area has a light gray header with the current project path: 'Dashboard / piercifani / GifWallet'. Below this is a navigation bar with five tabs: 'Builds' (selected), 'Workflow', 'Team', 'Code', and 'Settings', followed by a large green 'Start/Schedule a Build' button. A blue banner with the text 'Click here to setup deployment to your devices!' and a 'HIDE' link is centered above a search bar and filter dropdowns for triggers, branches, and workflows. A message box below the search bar says 'You have no scheduled builds. To schedule a build, [click here](#) or select the **Start a build** button above.' The main body contains three build log cards. The first card is purple and labeled 'Preparing' with a status of 'develop' and 'primary'. It was triggered at 16:42 and has a build ID of # 154. The second card is green and labeled 'Success' with 'develop' and 'primary'. It was triggered at 15:28 and has a build ID of # 153, with a note about merging pull request #21. The third card is also green and labeled 'Success', showing a build ID of 21 and a commit hash of 9ca0b17. A purple circular icon with a white smiley face is located in the bottom right corner of the main content area.

GitHub, Inc.

This repository Search Pull requests Issues Marketplace Explore

CodeCraftersIO / GifWallet Watch 0 Star 1 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

UI Tests #20

Closed piercifani wants to merge 2 commits into develop from feature/ui-tests

Conversation 0 Commits 2 Files changed 8 +277 -1

piercifani commented 6 hours ago
No description provided.

daiku60 added some commits a day ago
created some ui tests of basic navigation 2e44539
Corrected tests for iPad 43c0bb0

piercifani closed this 4 hours ago

piercifani deleted the feature/ui-tests branch 4 hours ago Restore branch

Write Preview

Leave a comment

Owner +

Reviewers
No reviews

Assignees
No one—assign yourself

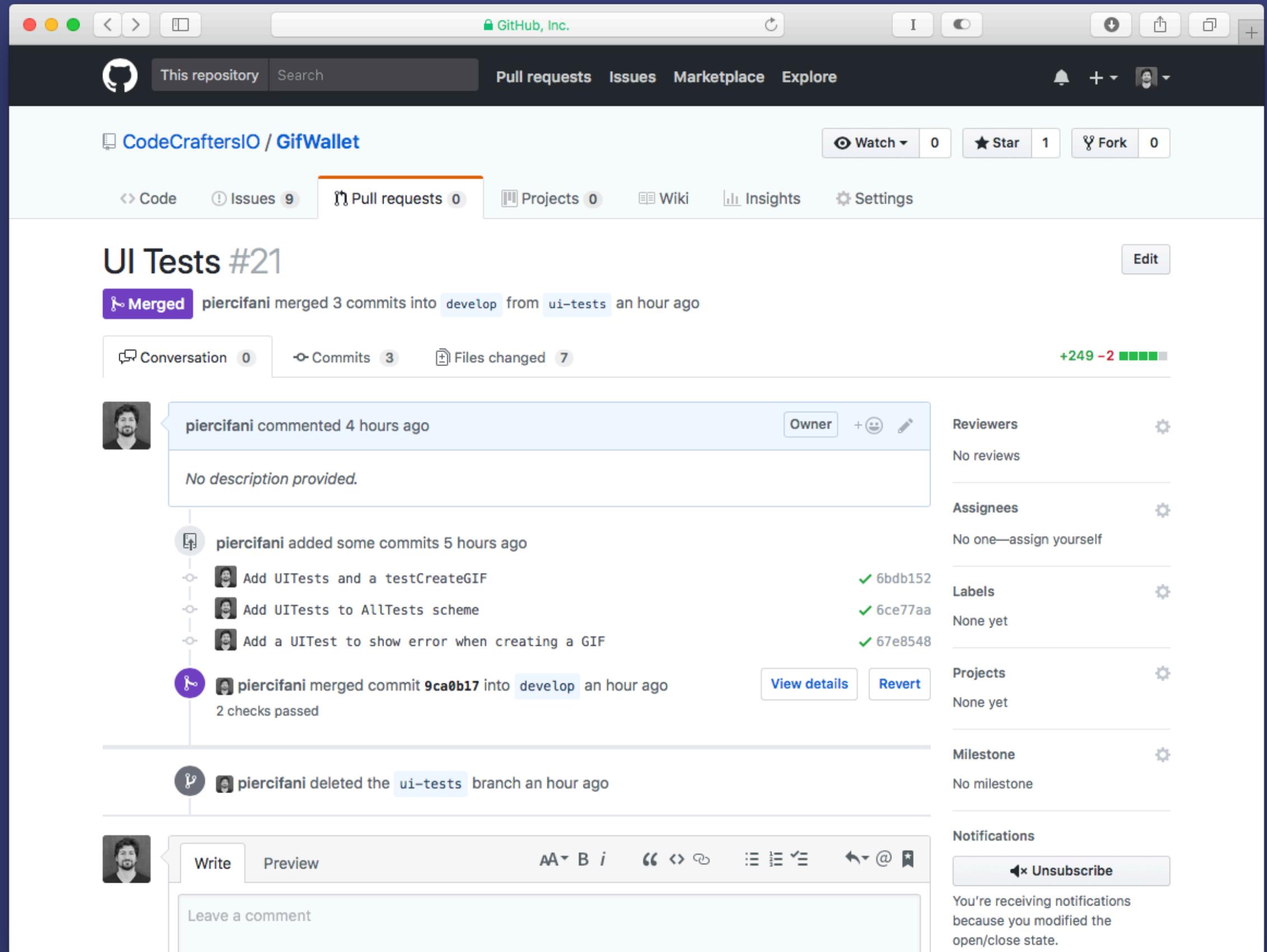
Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
 Unsubscribe

You're receiving notifications because you modified the open/close state.



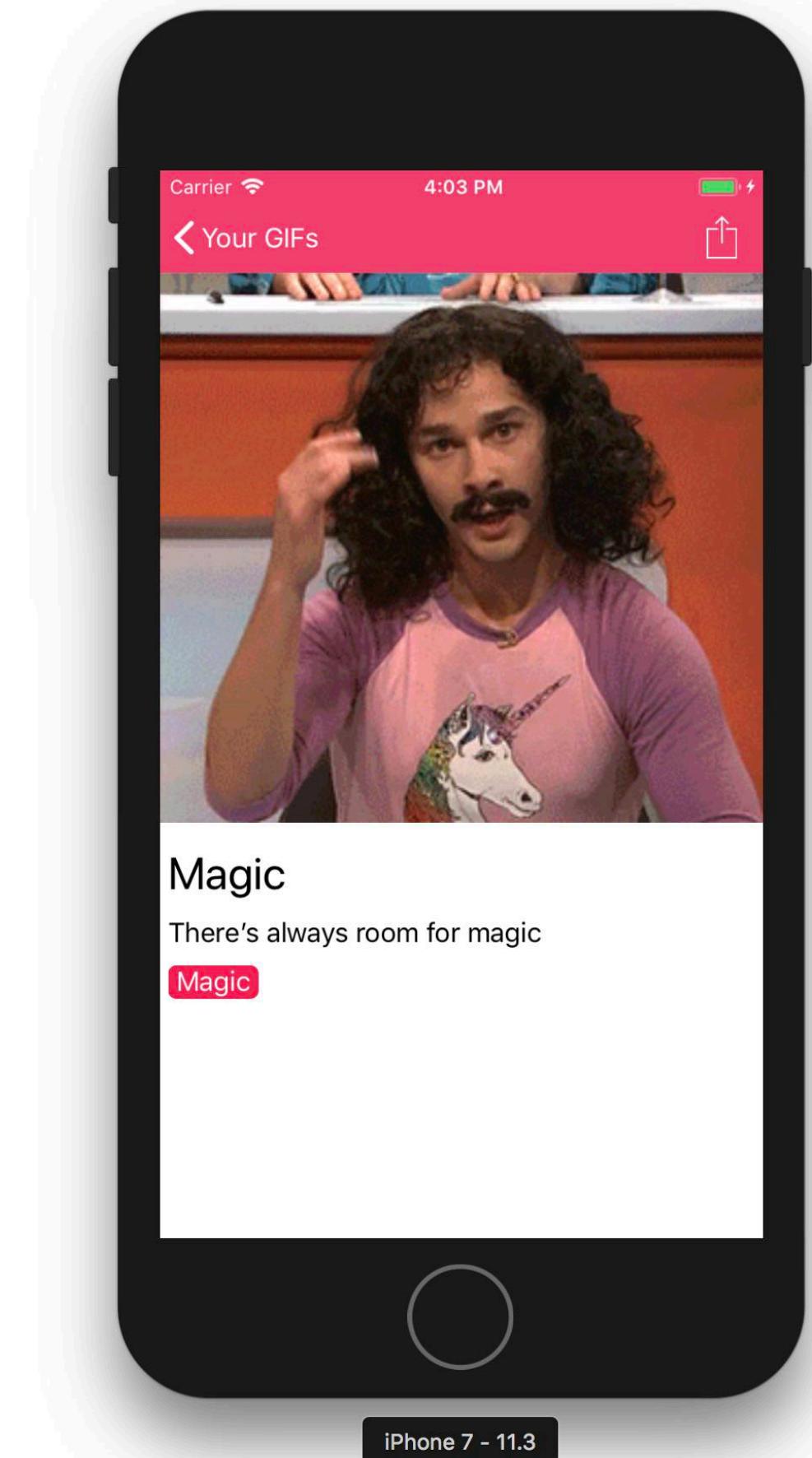
Architecture

- This is just a fancy word to explain how you manage complexity regarding:
- Presenting and dismissing `UIViewController` instances.
 - Passing data.

ViewController Data

→ MVVM allows us to quickly mock data required by a ViewController to display it's contents.

```
extension GIFDetailsViewController {  
    struct VM {  
        let gifID: String  
        let title: String  
        let subtitle: String  
        let url: URL  
        let tags: Set<String>  
    }  
}
```



Data Retrieval

- A type called Interactor will create mock/real VM and pass it to the VC.
- Interactors will be injected into the ViewControllers in the init

```
class GIFWalletViewController: UIViewController {  
  
    let interactor: GIFWalletInteractorType  
  
    init(interactor: GIFWalletInteractorType) {  
        self.interactor = interactor  
        super.init(nibName: nil, bundle: nil)  
    }  
}  
  
protocol GIFWalletInteractorType {  
    func fetchData(handler: @escaping WalletDataHandler)  
}
```

viewController Creation

- It'll also be handled by Interactor.
- However, in a complex app this would not scale correctly, and this class should be split into two: Repository and Wireframe.

Let's go!

A screenshot of a GitHub repository page for "CodeCraftersIO / GifWallet". The page shows basic repository statistics and a list of recent commits.

The repository has the following metrics:

- 101 commits
- 1 branch
- 10 releases
- 2 contributors
- MIT license

The commits listed are:

File / Action	Description	Date
.gitignore	Bootstrap is completed	a month ago
GenerateScreenshots.sh	Create a script to automate snapshot generation	5 days ago
LICENSE	Initial commit	a month ago
GifWallet.xcodeproj	Add UITests to AllTests scheme	2 days ago
GifWallet.xcworkspace	Add GIFCreateVC to add a new GIF	5 days ago
GifWallet	Add a UITest to show error when creating a GIF	2 days ago
GifWalletKit	Finish hooking up the CoreData store to every interactor	3 days ago
GifWalletKitTests	Merge branch 'promises' into core-data	4 days ago
GifWalletTests	Finish hooking up the CoreData store to every interactor	3 days ago
GifWalletUITests	Add a UITest to show error when creating a GIF	2 days ago

The "Fork" button in the top right corner is circled in red.