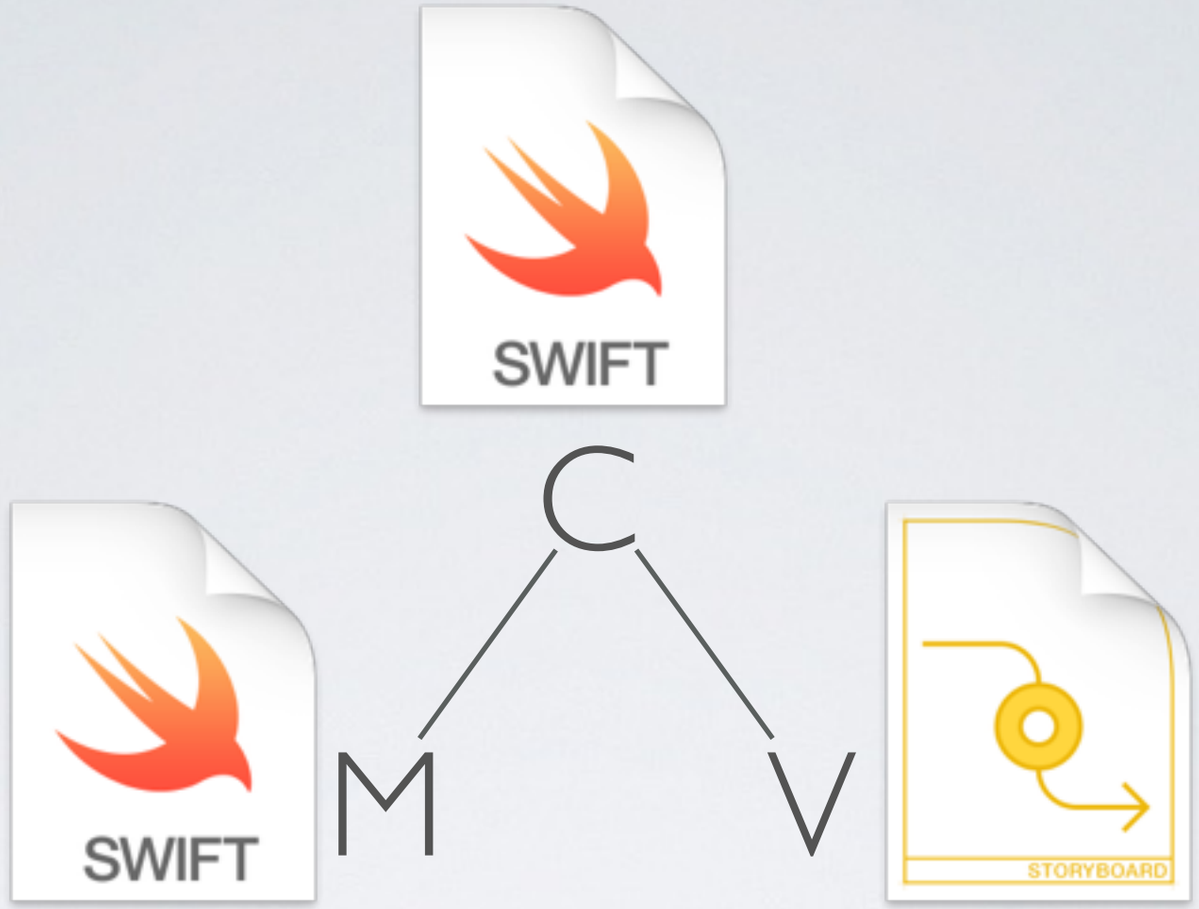
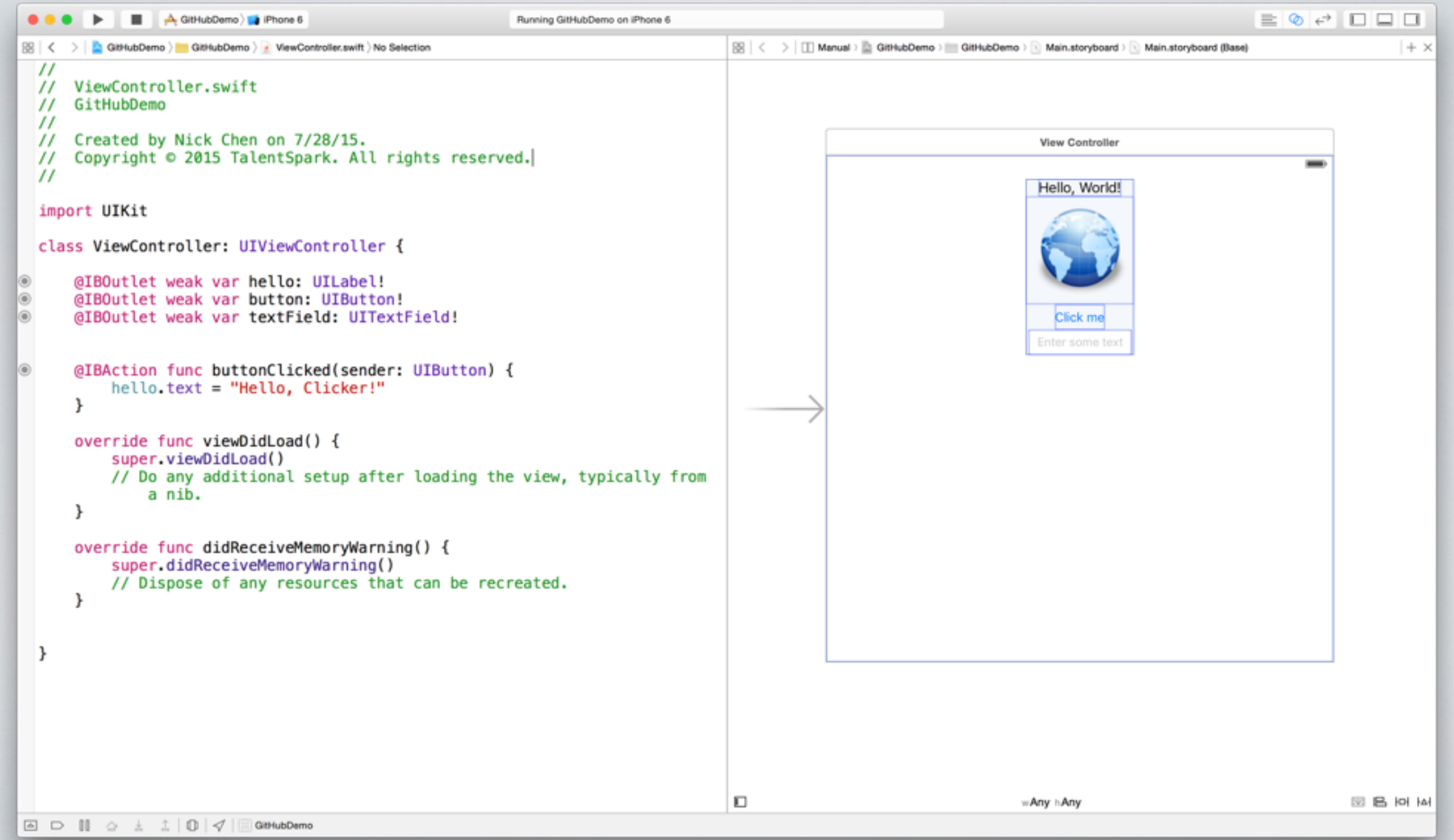


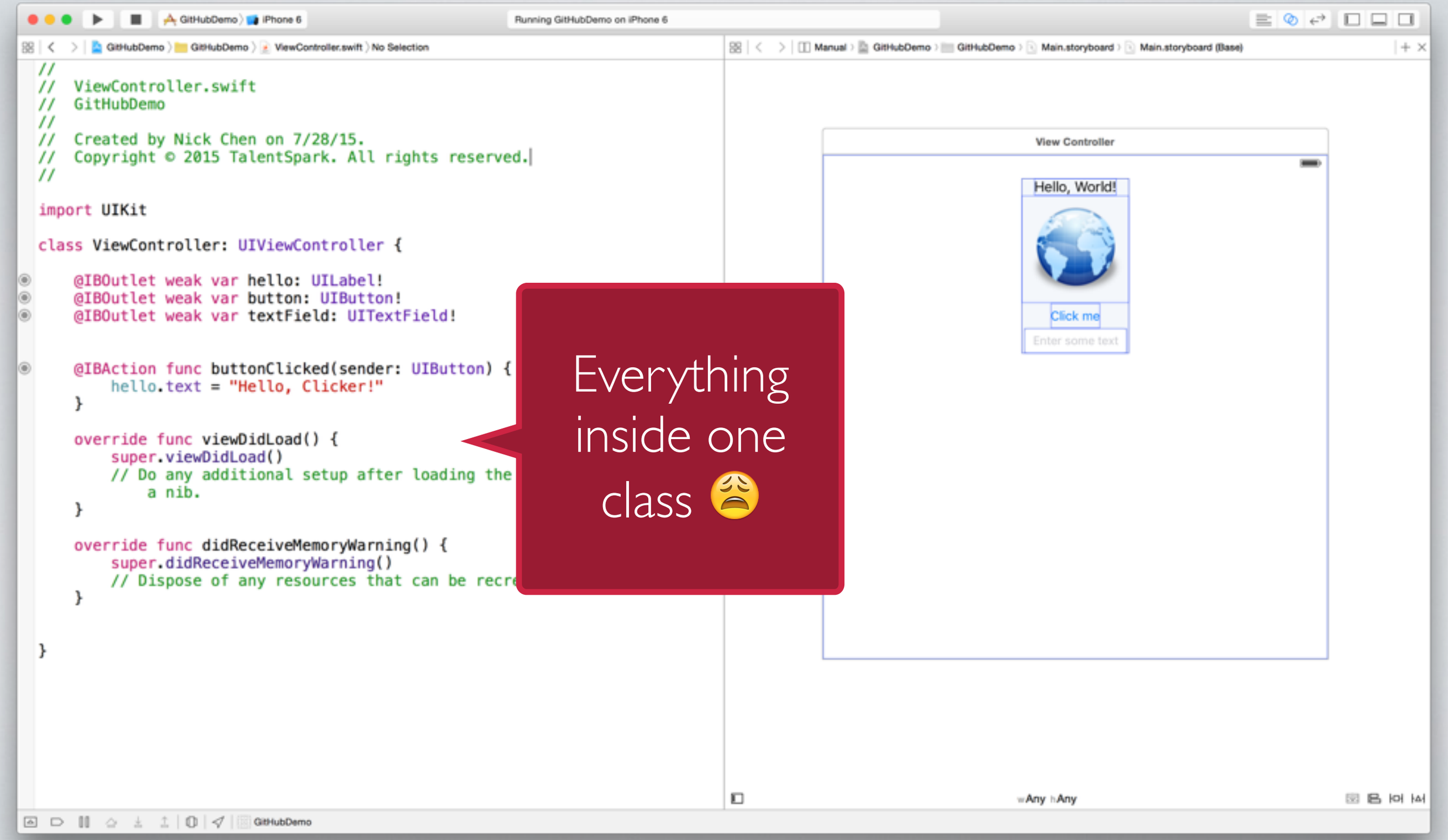
MODEL-VIEW-CONTROLLER AND TABLEVIEWS

Nick Chen
Fall 2015
talentspark.io

PREVIOUSLY


GitHubDemo
iPhone 6
Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
ViewController.swift
No Selection
Manual
GitHubDemo
GitHubDemo
Main.storyboard
Main.storyboard (Base)
//
// ViewController.swift
// GitHubDemo
//
// Created by Nick Chen on 7/28/15.
// Copyright © 2015 TalentSpark. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var hello: UILabel!
@IBOutlet weak var button: UIButton!
@IBOutlet weak var textField: UITextField!
@IBAction func buttonClicked(sender: UIButton) {
hello.text = "Hello, Clicker!"
}
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from
a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
View Controller
Hello, World!
Click me
Enter some text
wAny hAny
GitHubDemo

PREVIOUSLY


GitHubDemo 〉 iPhone 6
Running GitHubDemo on iPhone 6
GitHubDemo 〉 GitHubDemo 〉 ViewController.swift 〉 No Selection
Manual 〉 GitHubDemo 〉 GitHubDemo 〉 Main.storyboard 〉 Main.storyboard (Base)
//
// ViewController.swift
// GitHubDemo
//
// Created by Nick Chen on 7/28/15.
// Copyright © 2015 TalentSpark. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var hello: UILabel!
@IBOutlet weak var button: UIButton!
@IBOutlet weak var textField: UITextField!
@IBAction func buttonClicked(sender: UIButton) {
hello.text = "Hello, Clicker!"
}
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the
a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recre
}
}
Everything inside one class 😩
View Controller
Hello, World!
Click me
Enter some text
wAny hAny
GitHubDemo

SPAGHETTI CODE

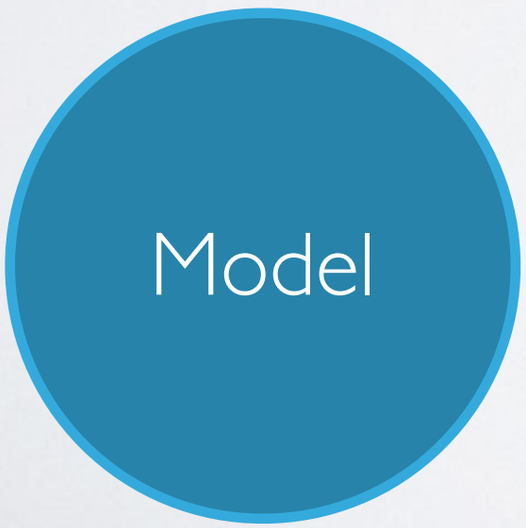
SPAGHETTI CODE

Controller

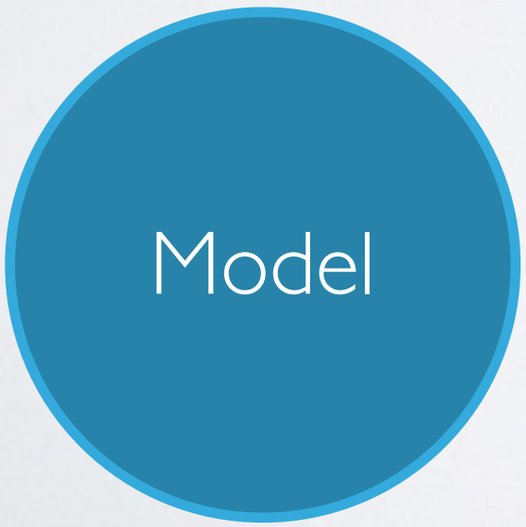
Model

View

SPAGHETTI CODE



NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

Controller

Handles the actual data and business logic

Model

View

NON-SPAGHETTI CODE

Controller

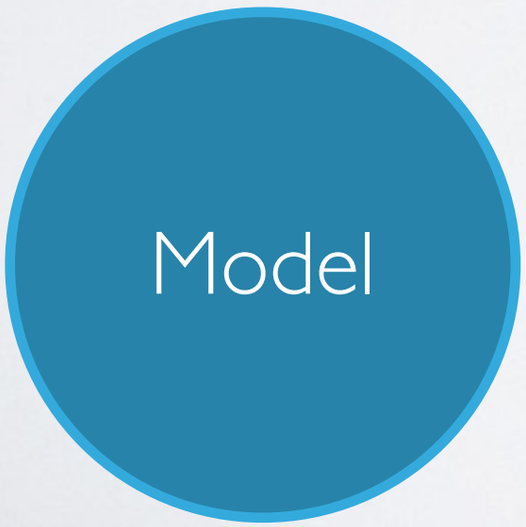
Handles the actual data and business logic

```
public class MyDomainModel {
    func myBusinessLogic() {}
}
```

Model

View

NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

Controller

Handles the presentation of data

Model

View

NON-SPAGHETTI CODE

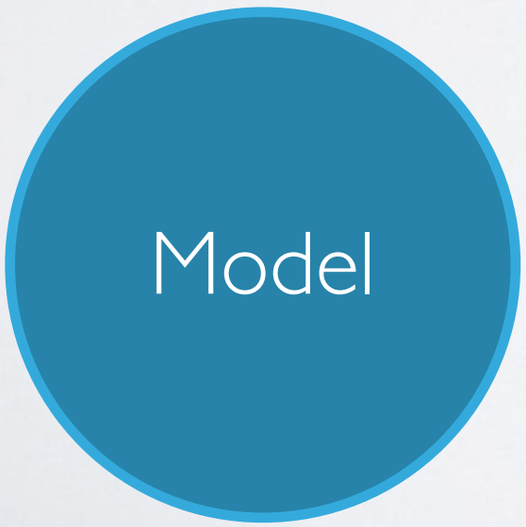
Controller

Handles the presentation of data

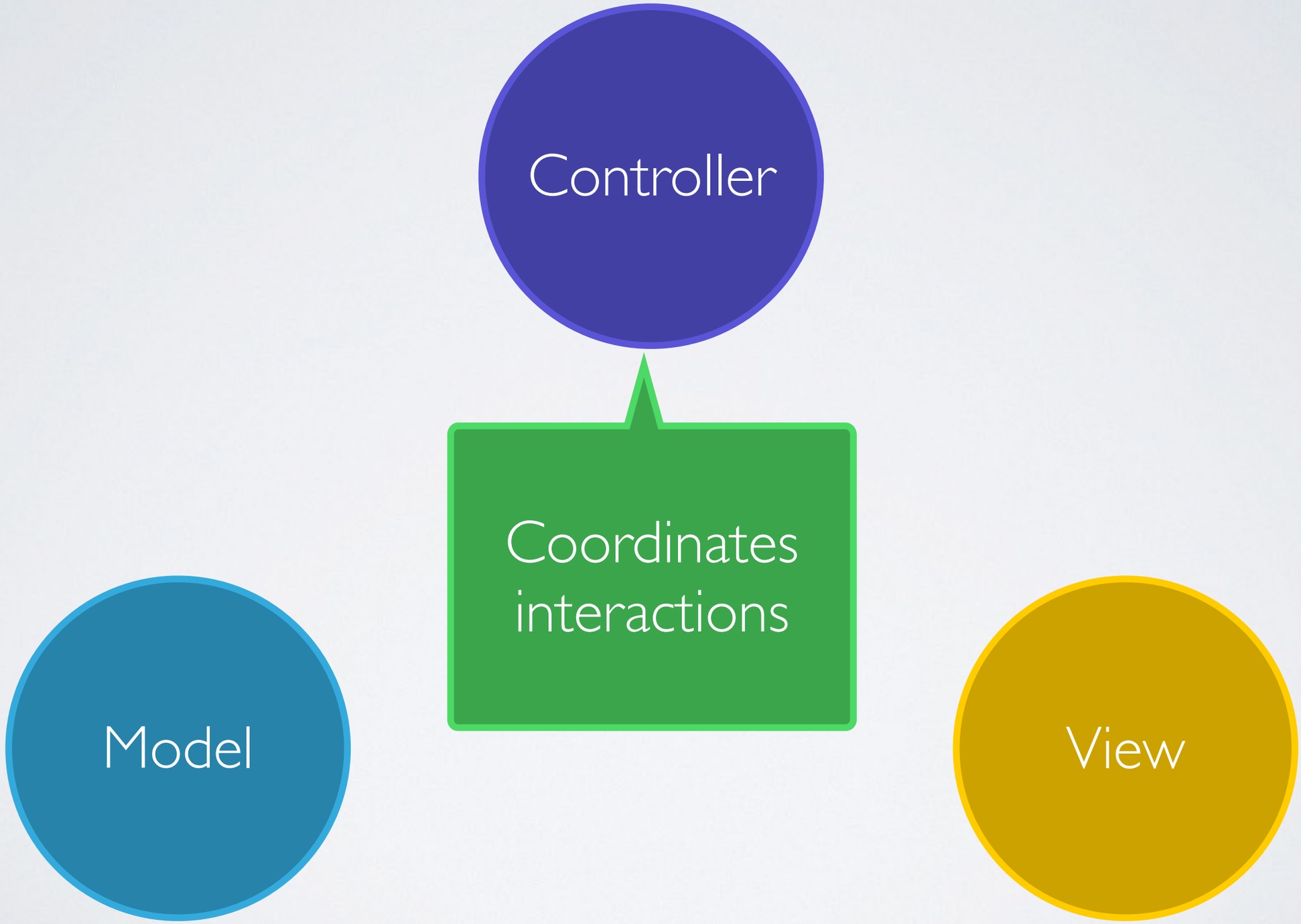
Model

View

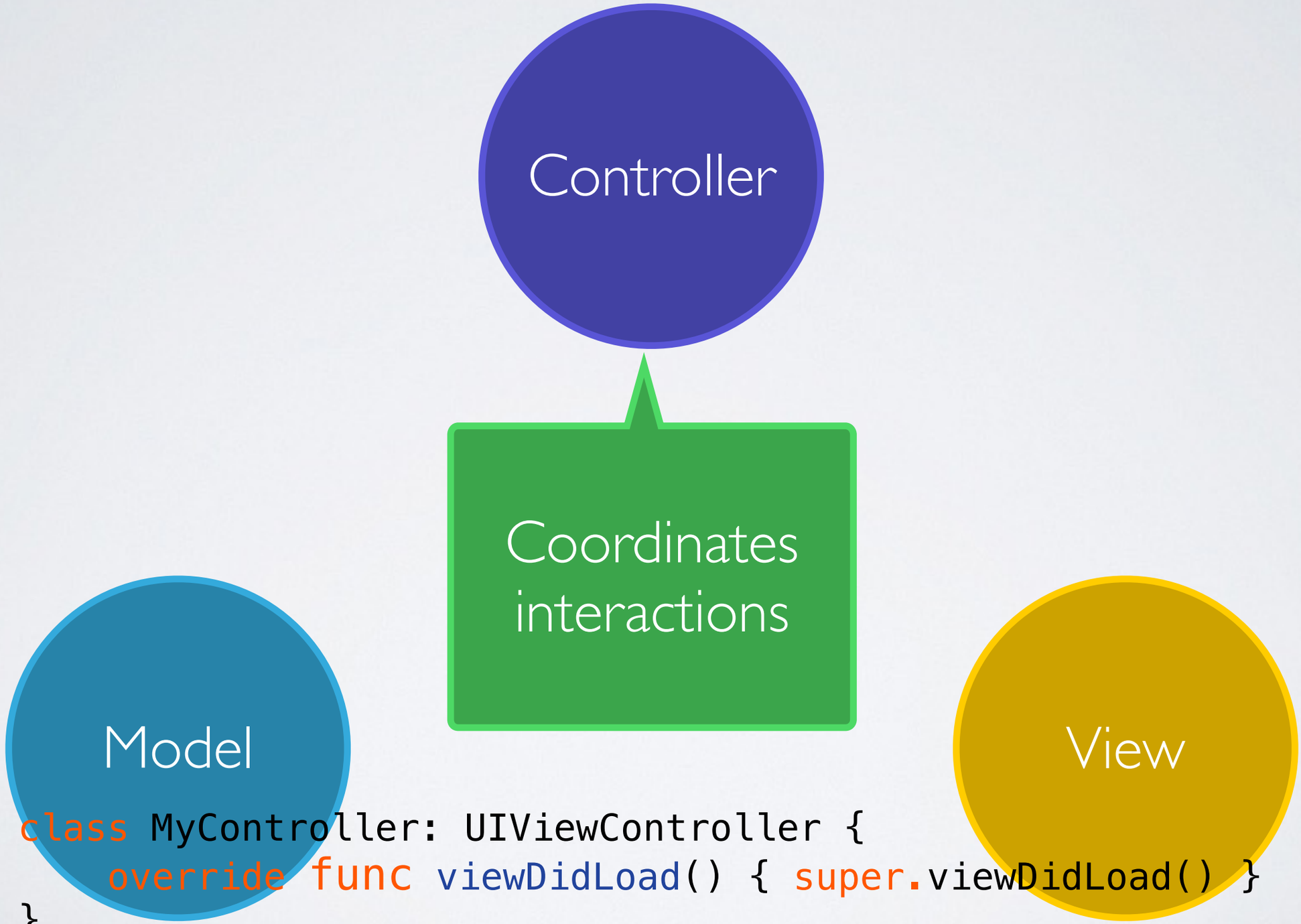
NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

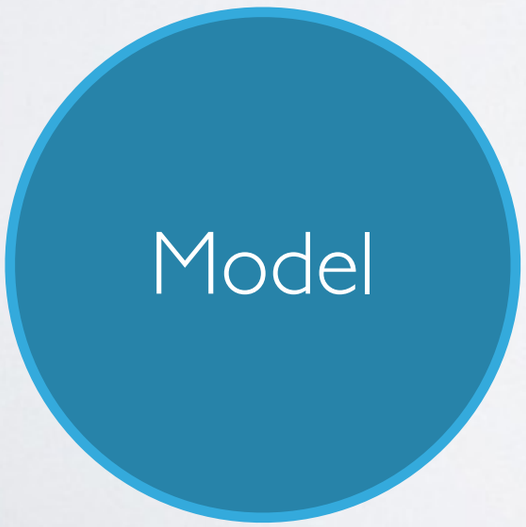


NON-SPAGHETTI CODE

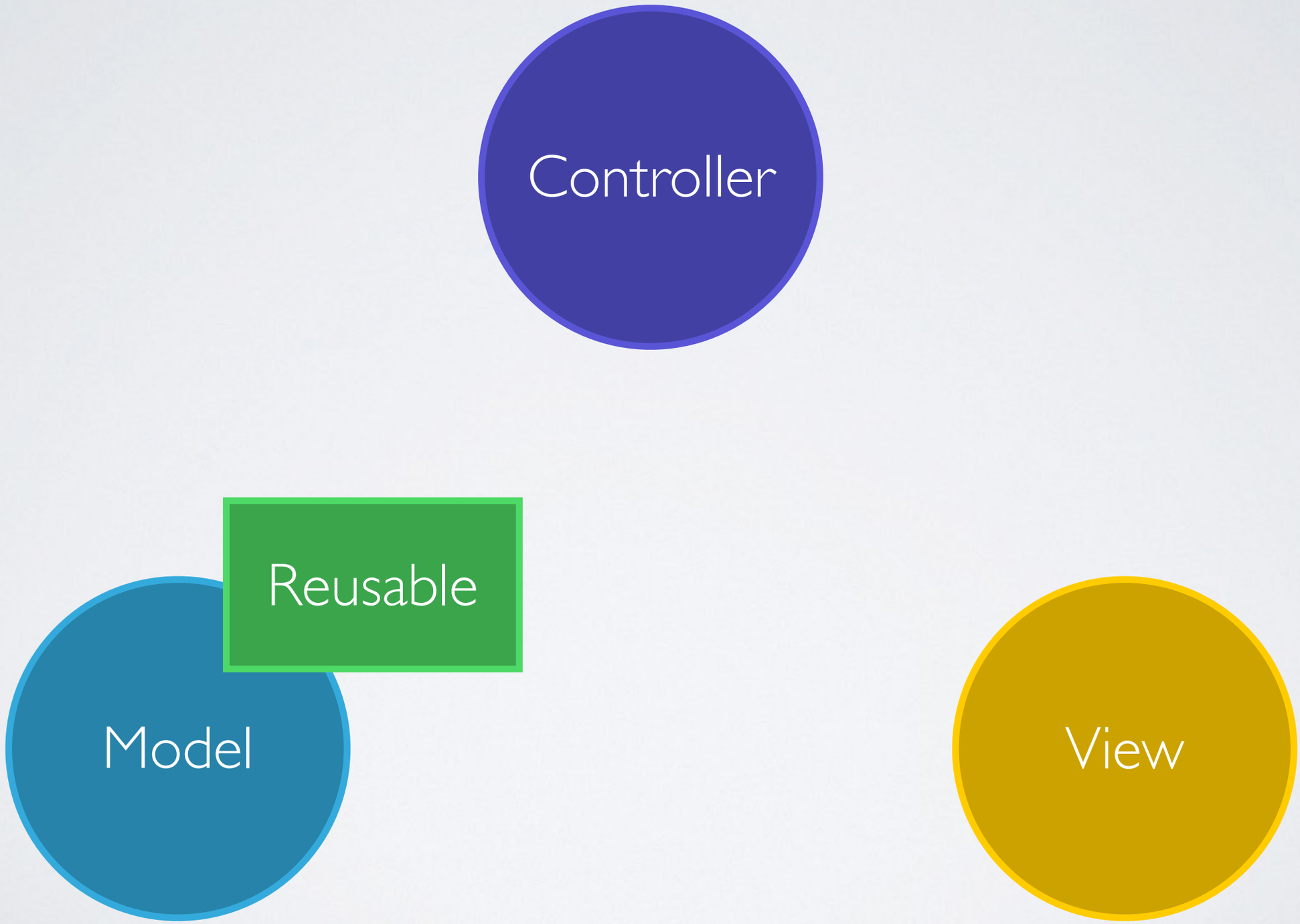



```
class MyController: UIViewController {
    override func viewDidLoad() { super.viewDidLoad() }
}
```

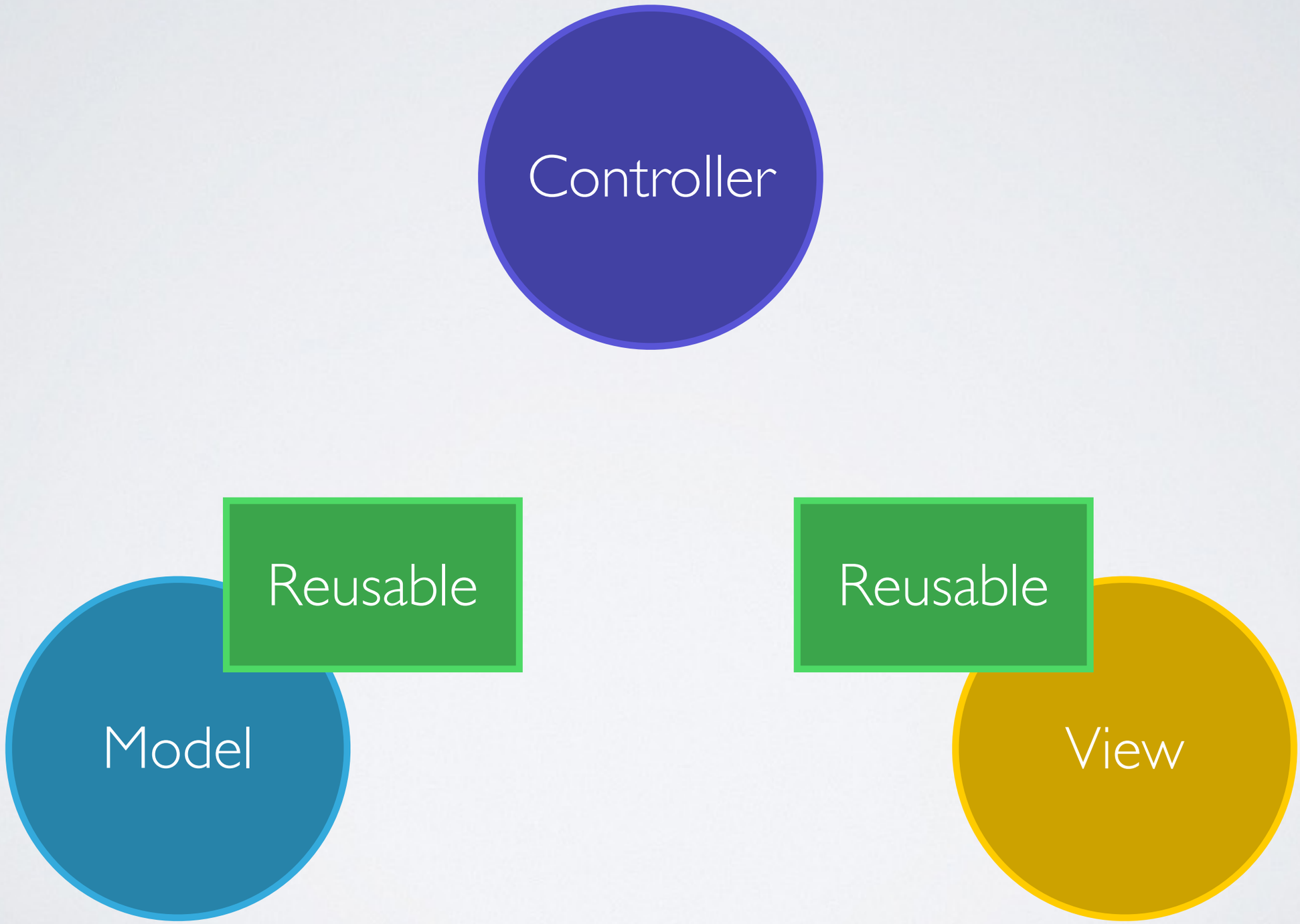
NON-SPAGHETTI CODE



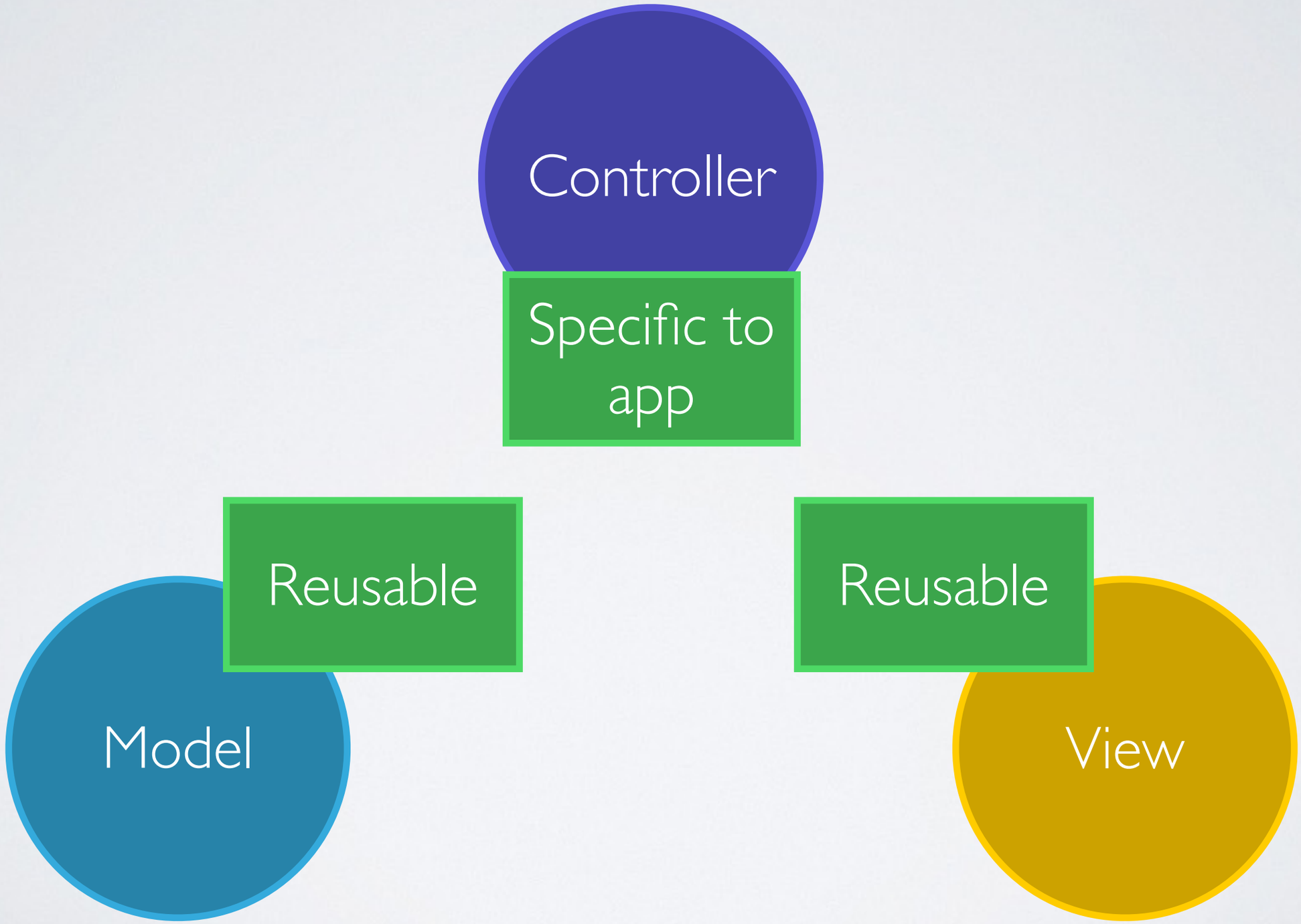
NON-SPAGHETTI CODE



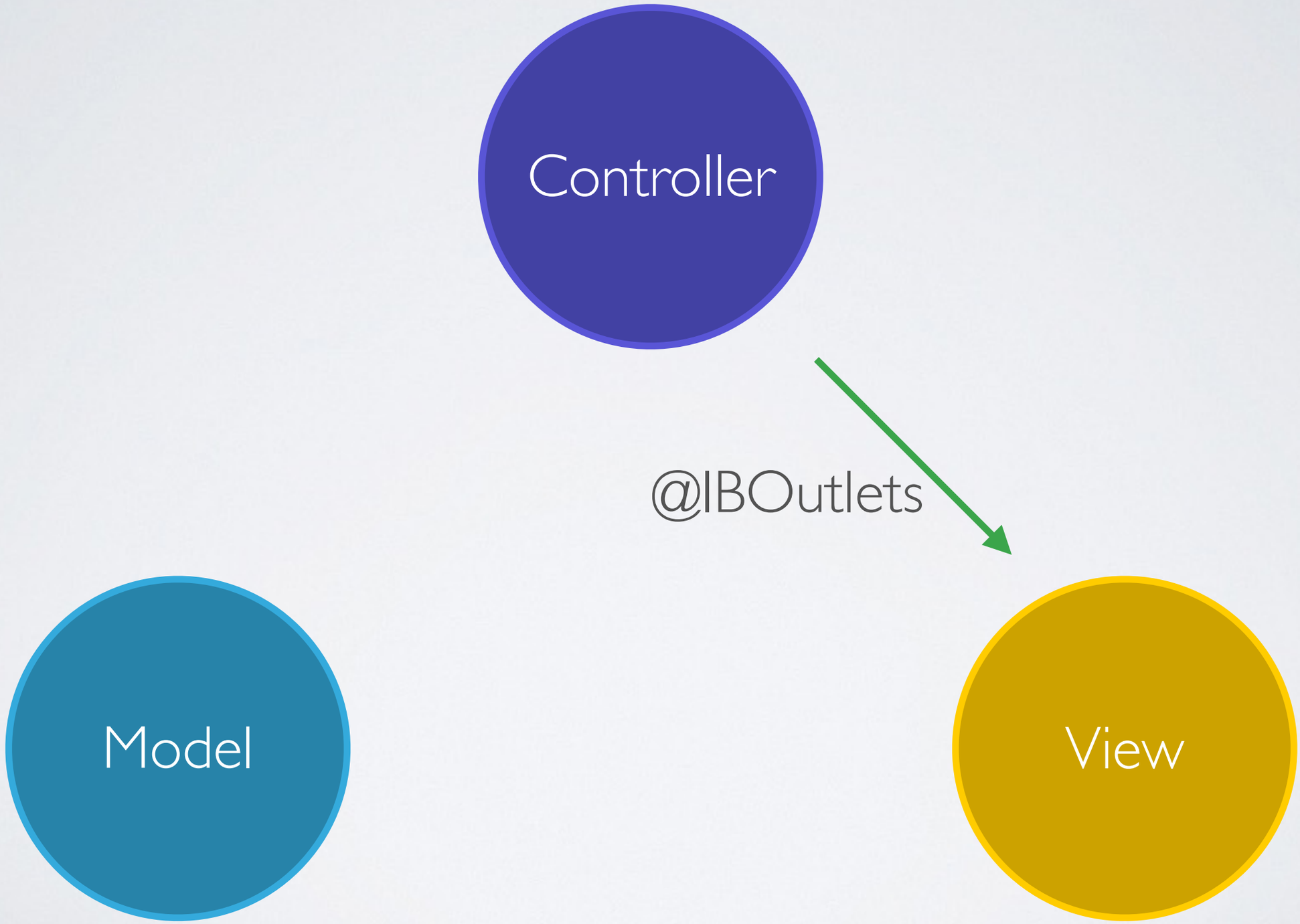
NON-SPAGHETTI CODE



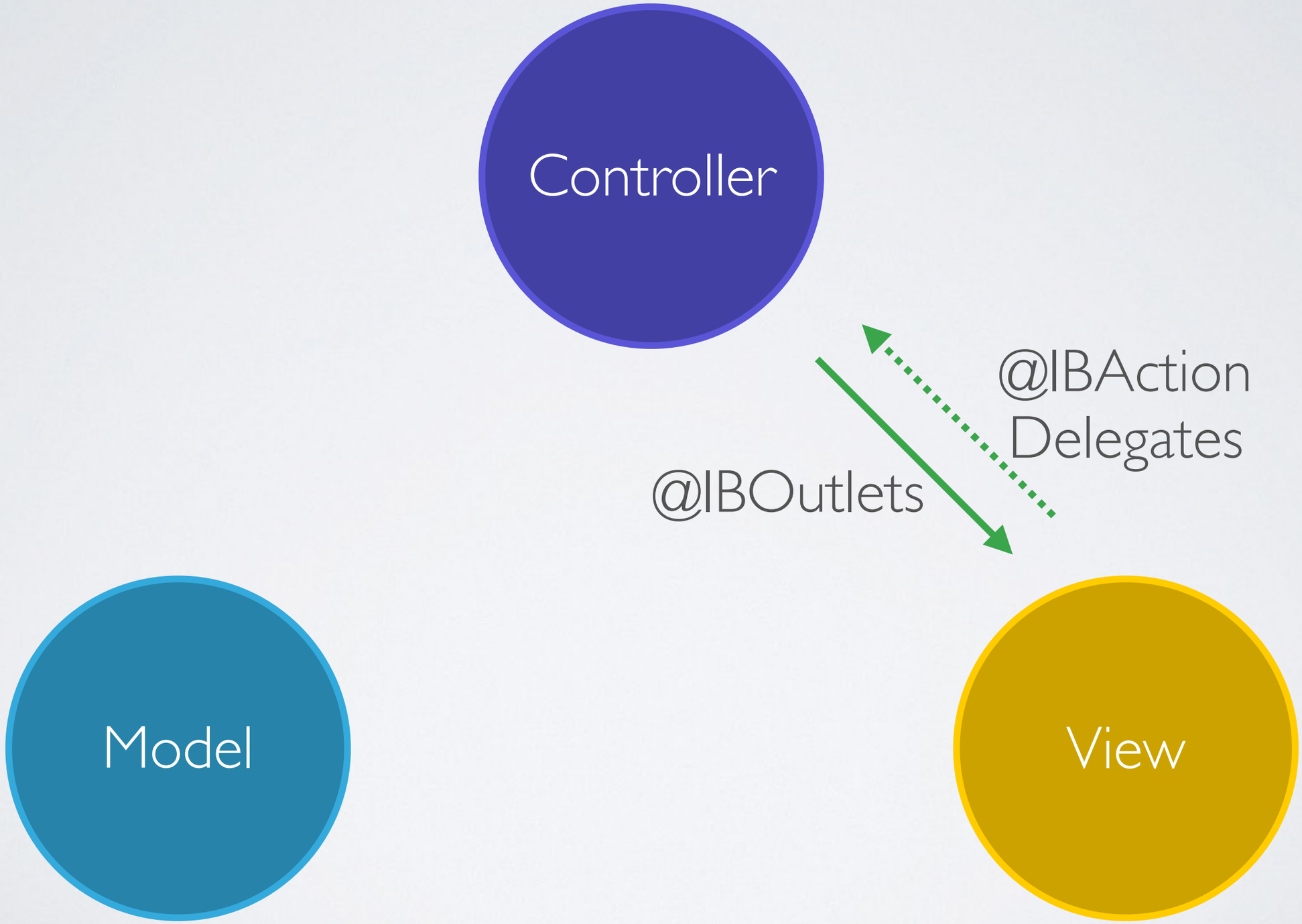
NON-SPAGHETTI CODE



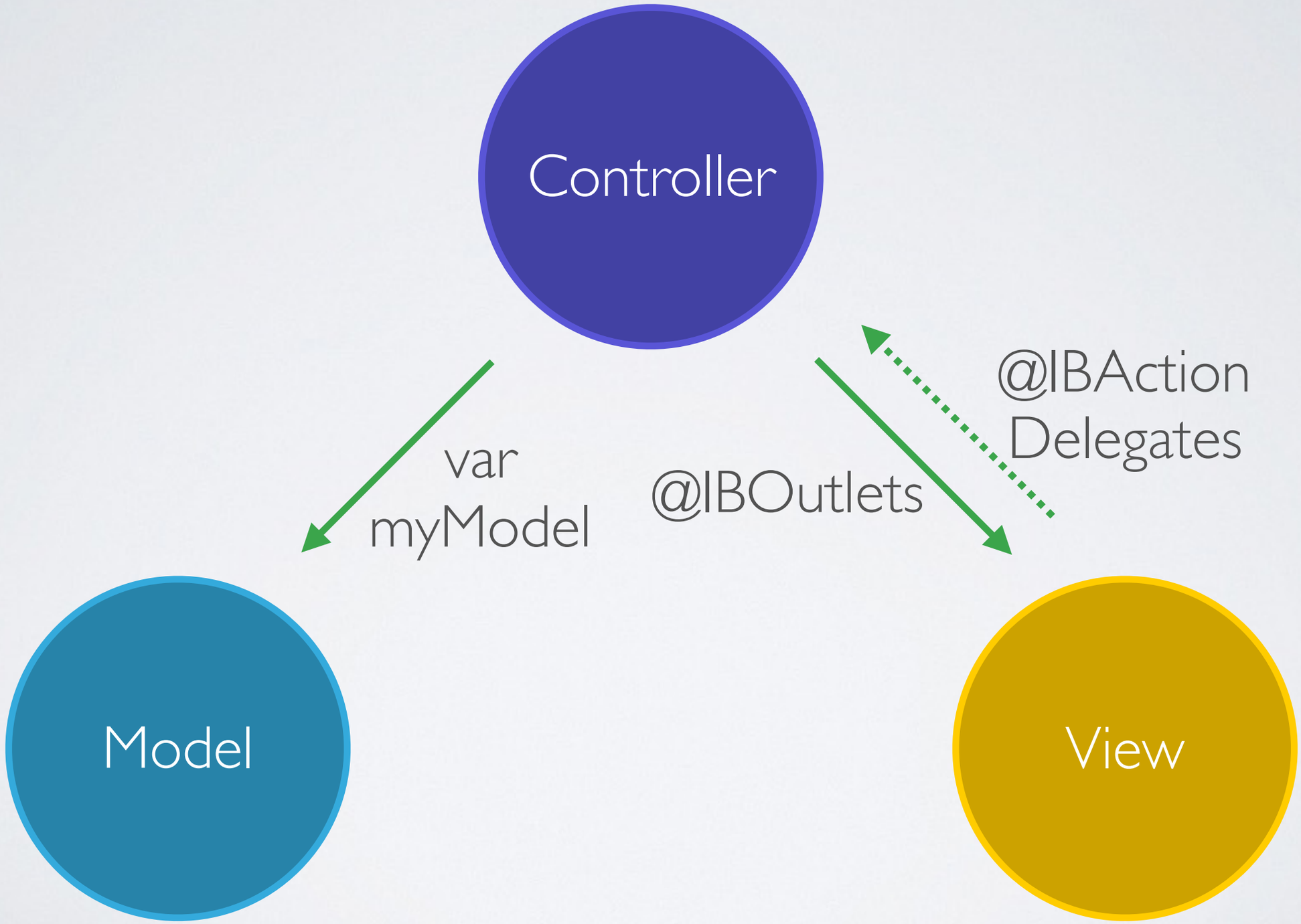
NON-SPAGHETTI CODE



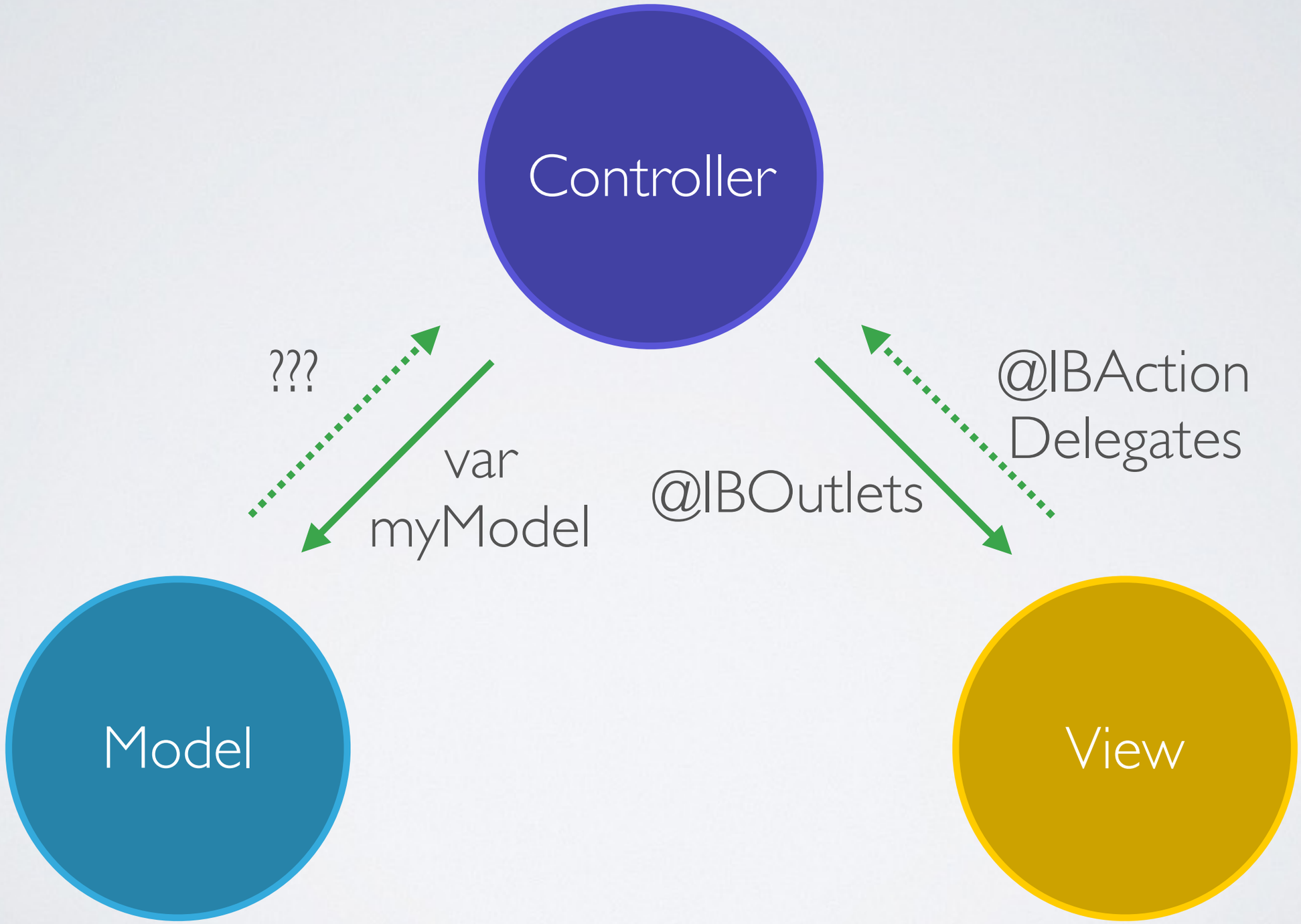
NON-SPAGHETTI CODE



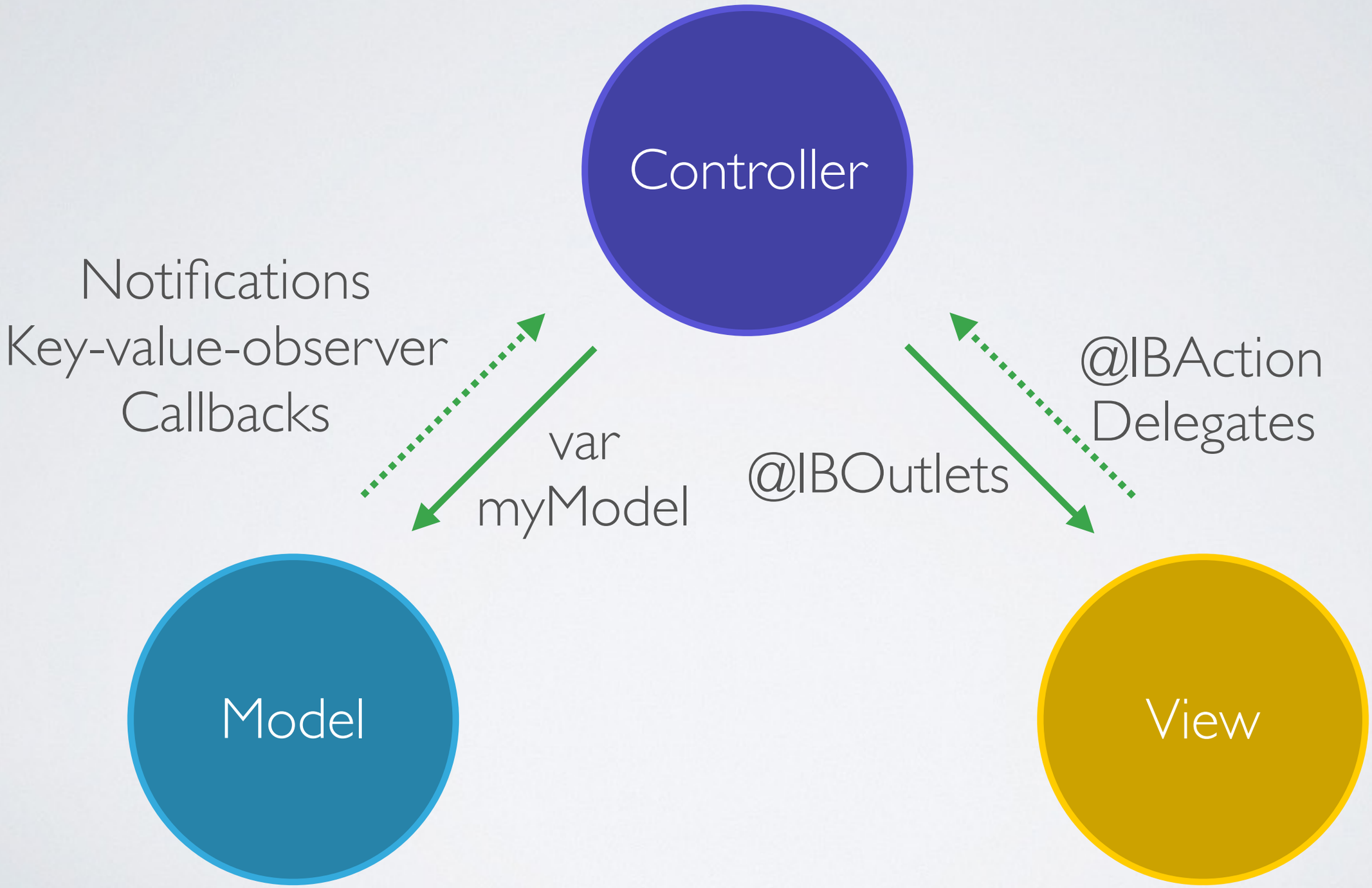
NON-SPAGHETTI CODE



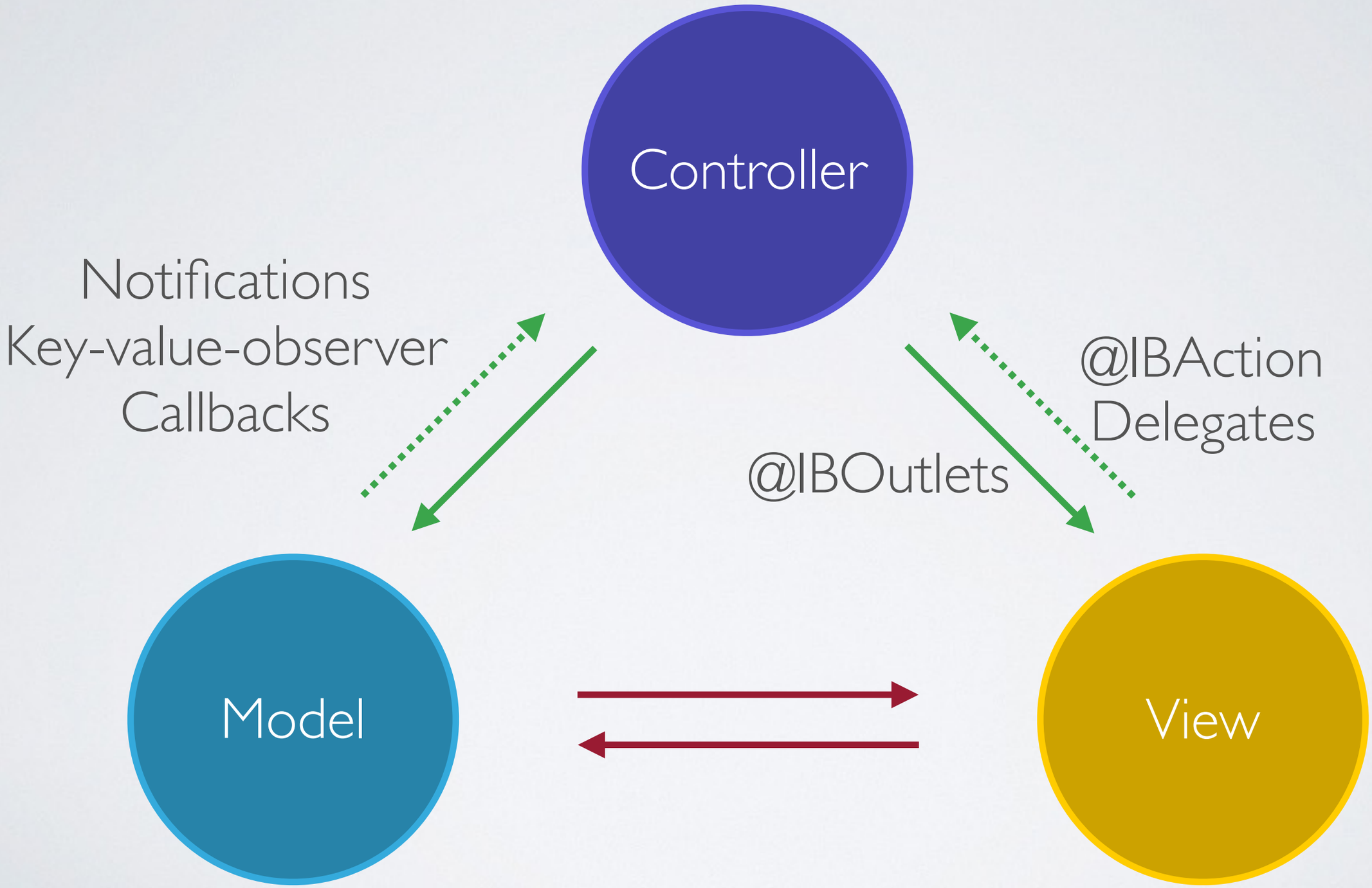
NON-SPAGHETTI CODE



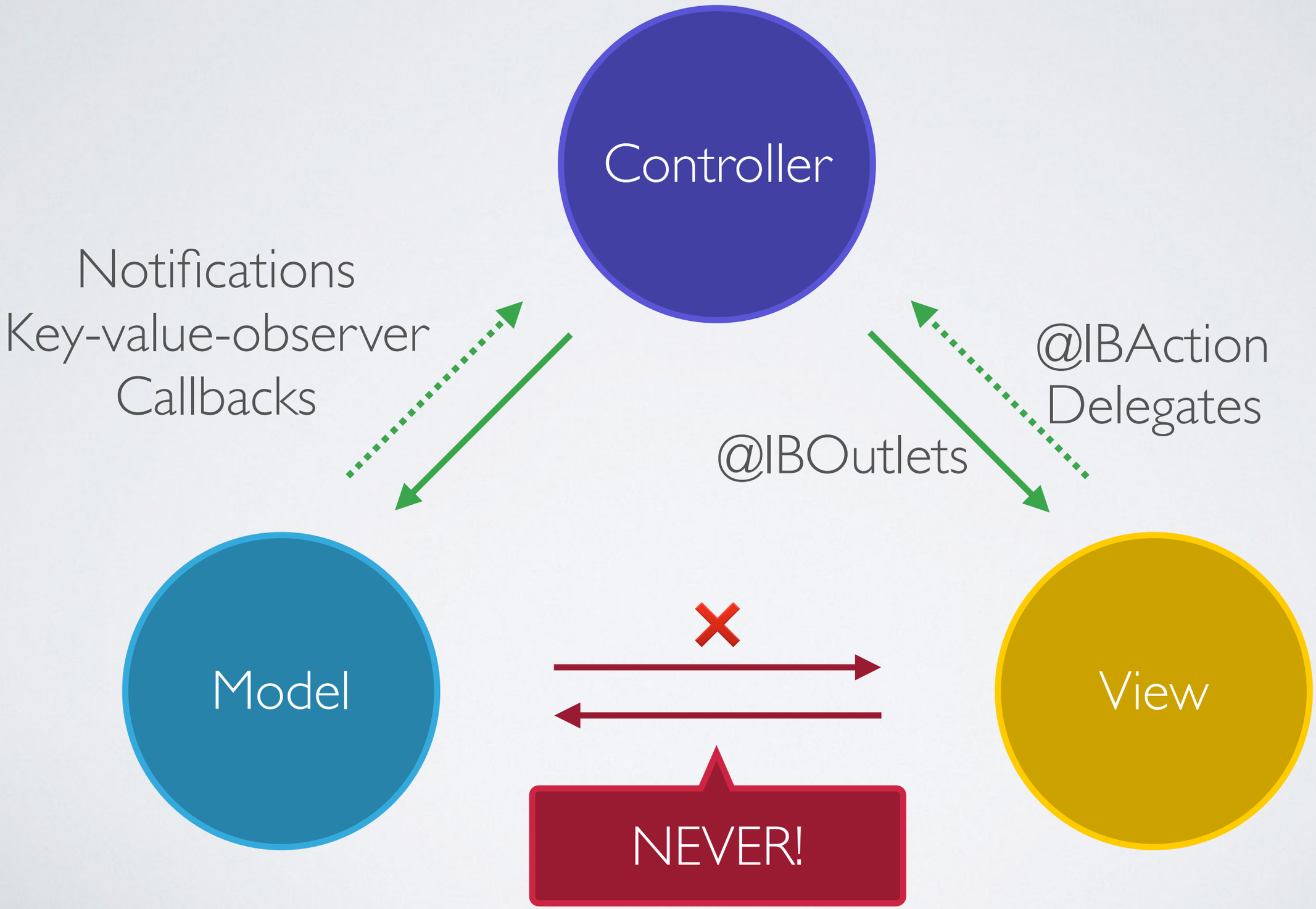
NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

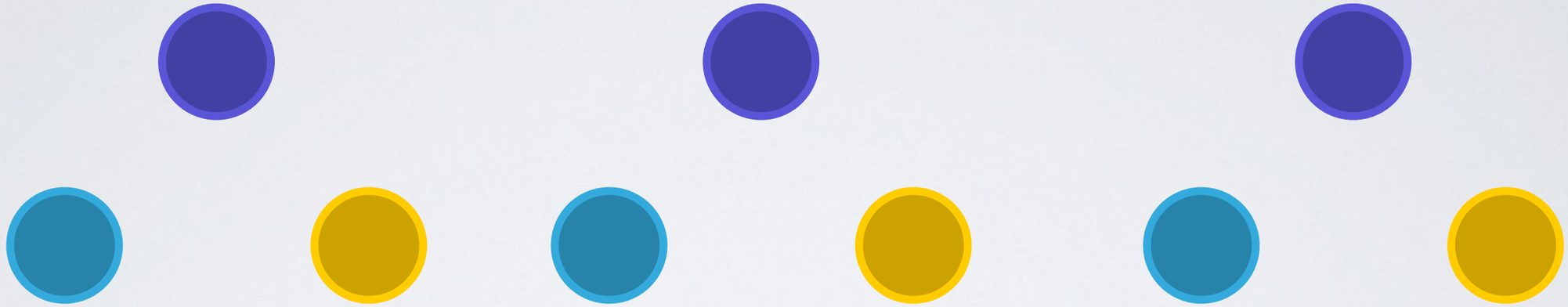


NON-SPAGHETTI CODE

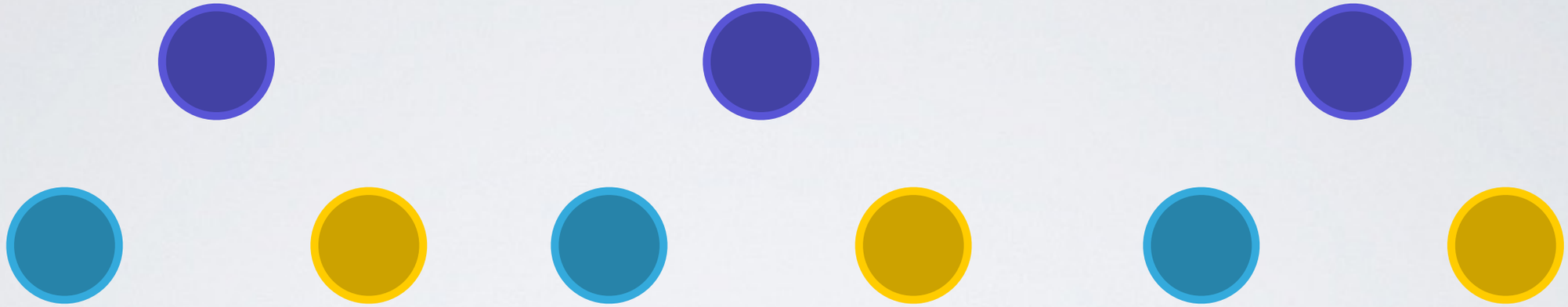


MULTIPLE MVCS

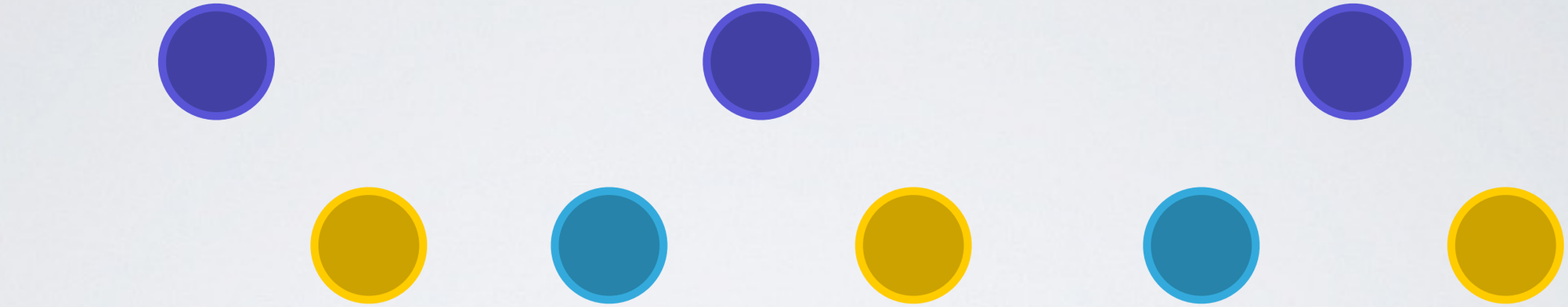
MULTIPLE MVCS



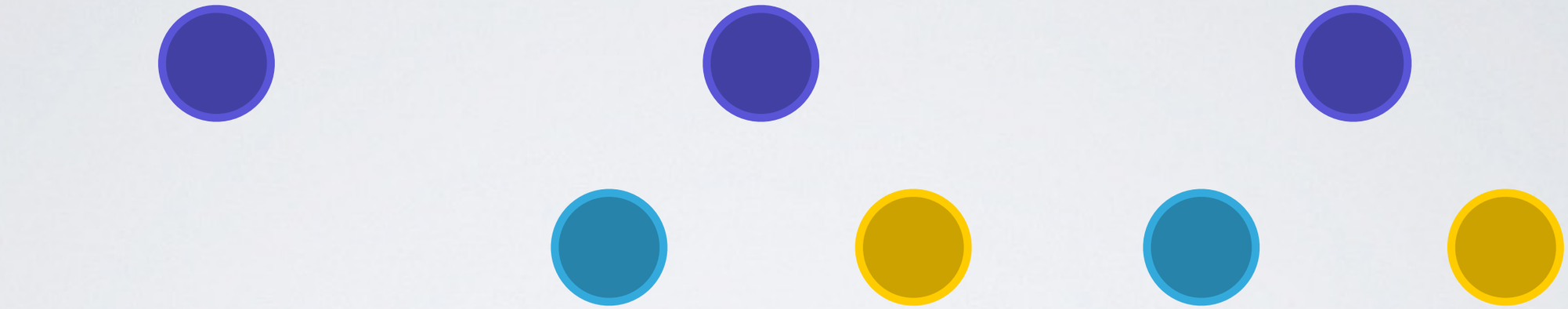
MULTIPLE MVCS



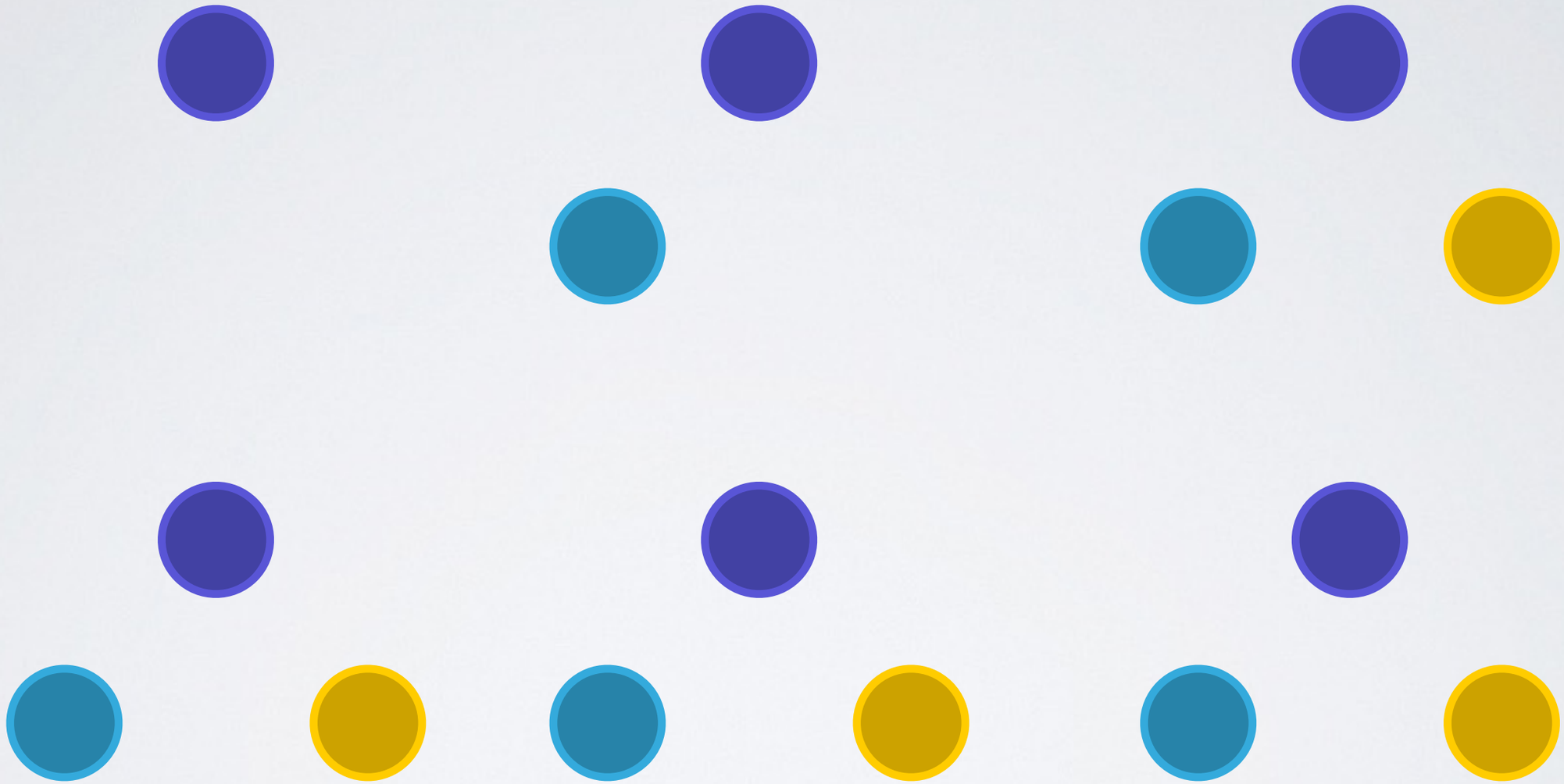
MULTIPLE MVCS



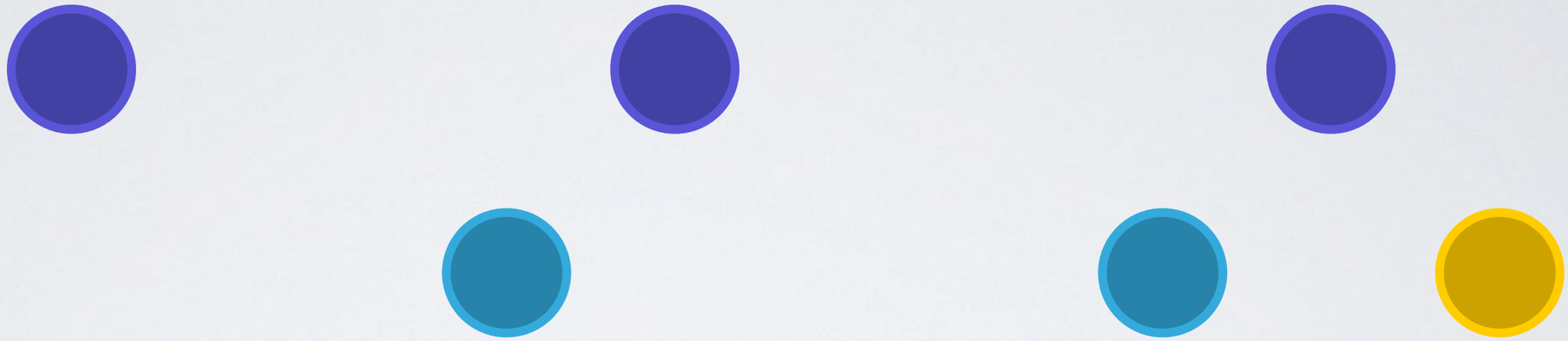
MULTIPLE MVCS



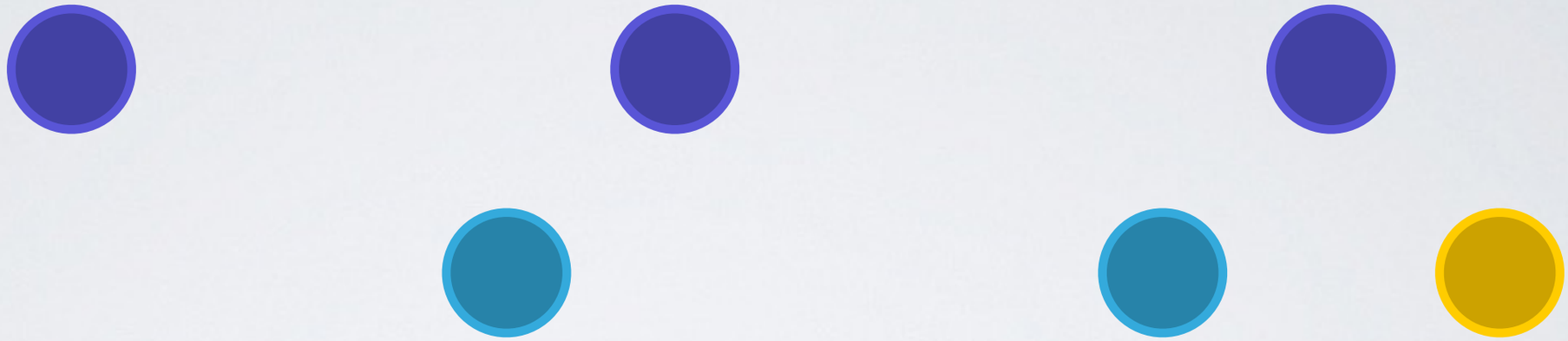
MULTIPLE MVCS



MULTIPLE MVCS

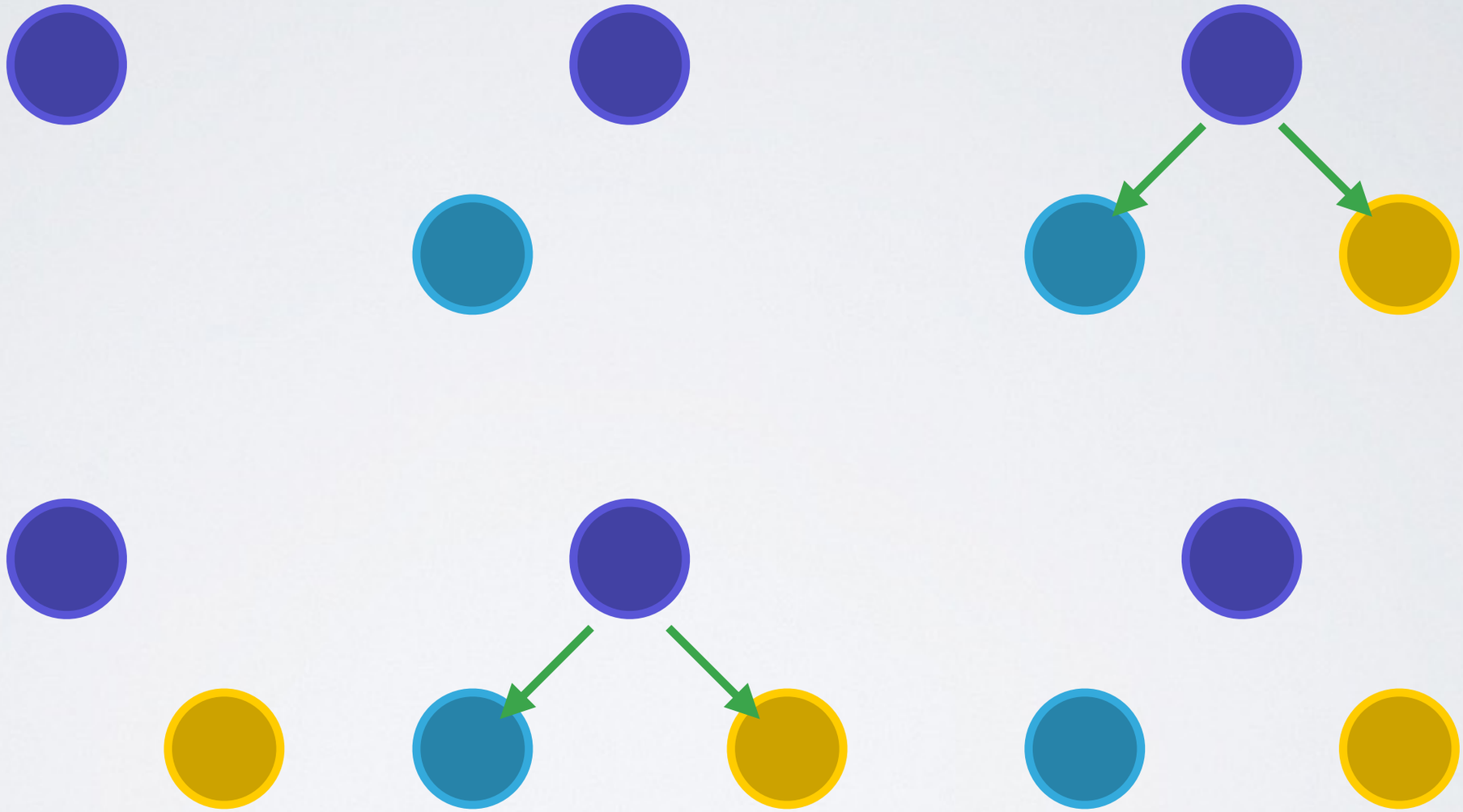


MULTIPLE MVCS

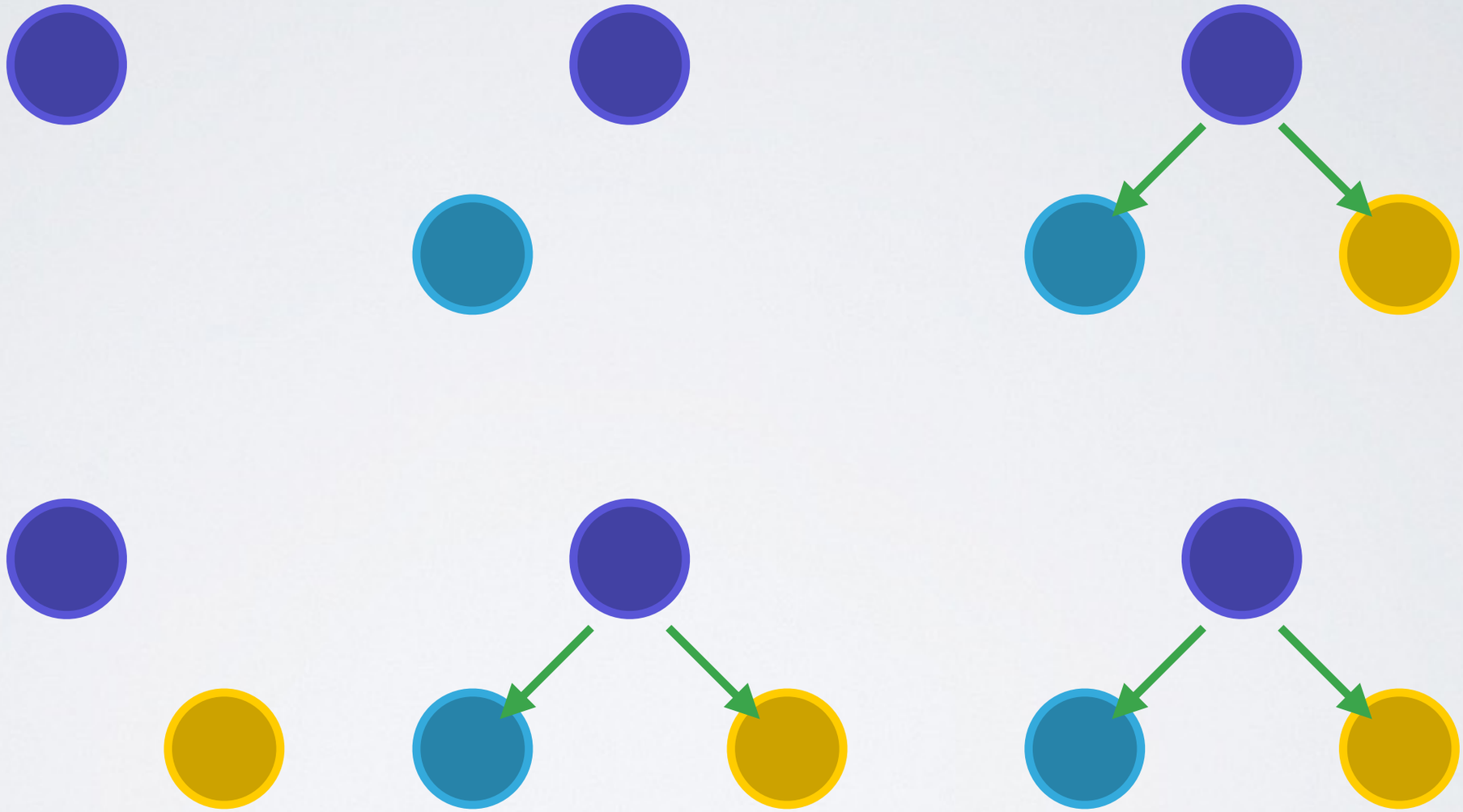


MULTIPLE MVCS

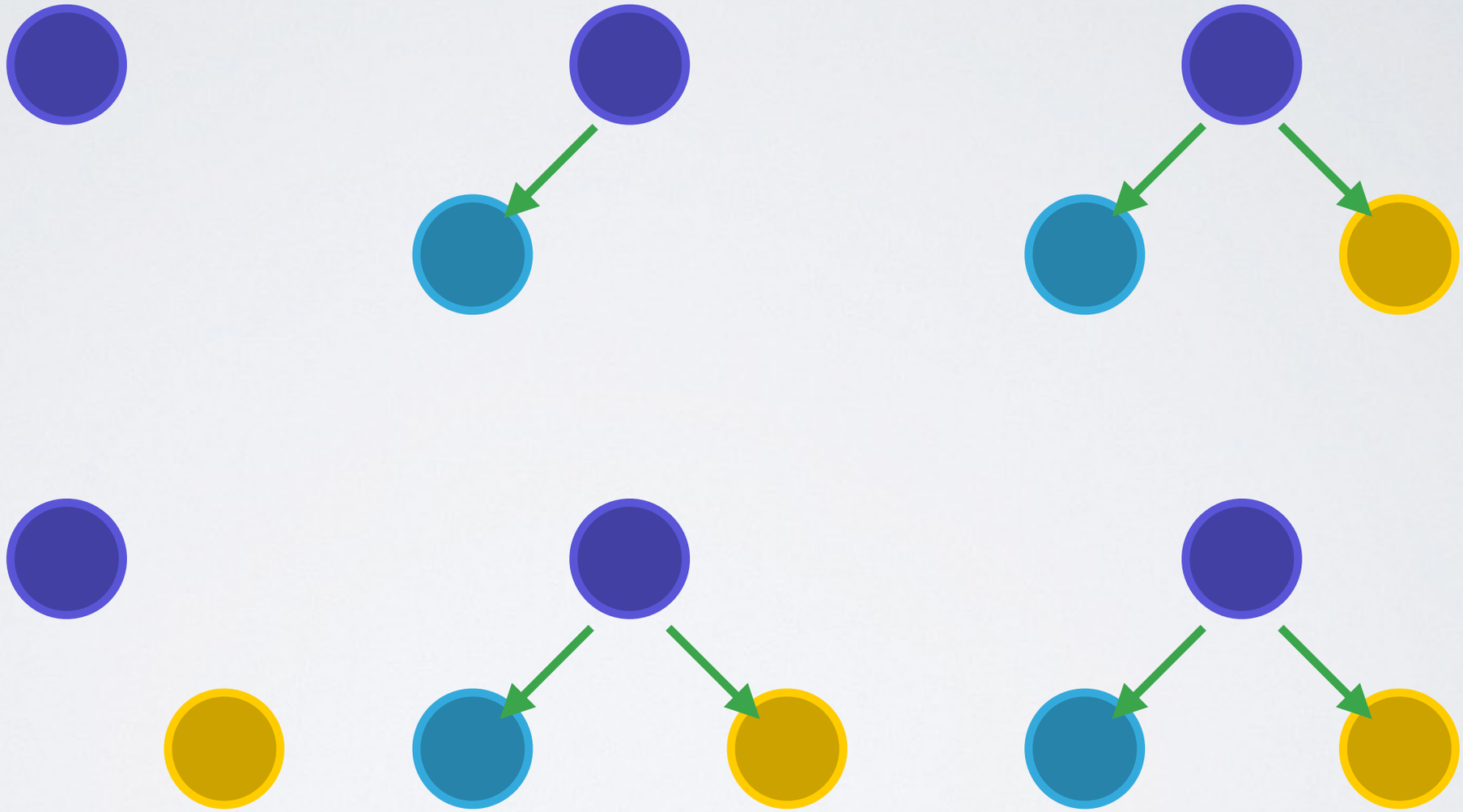
MULTIPLE MVCS



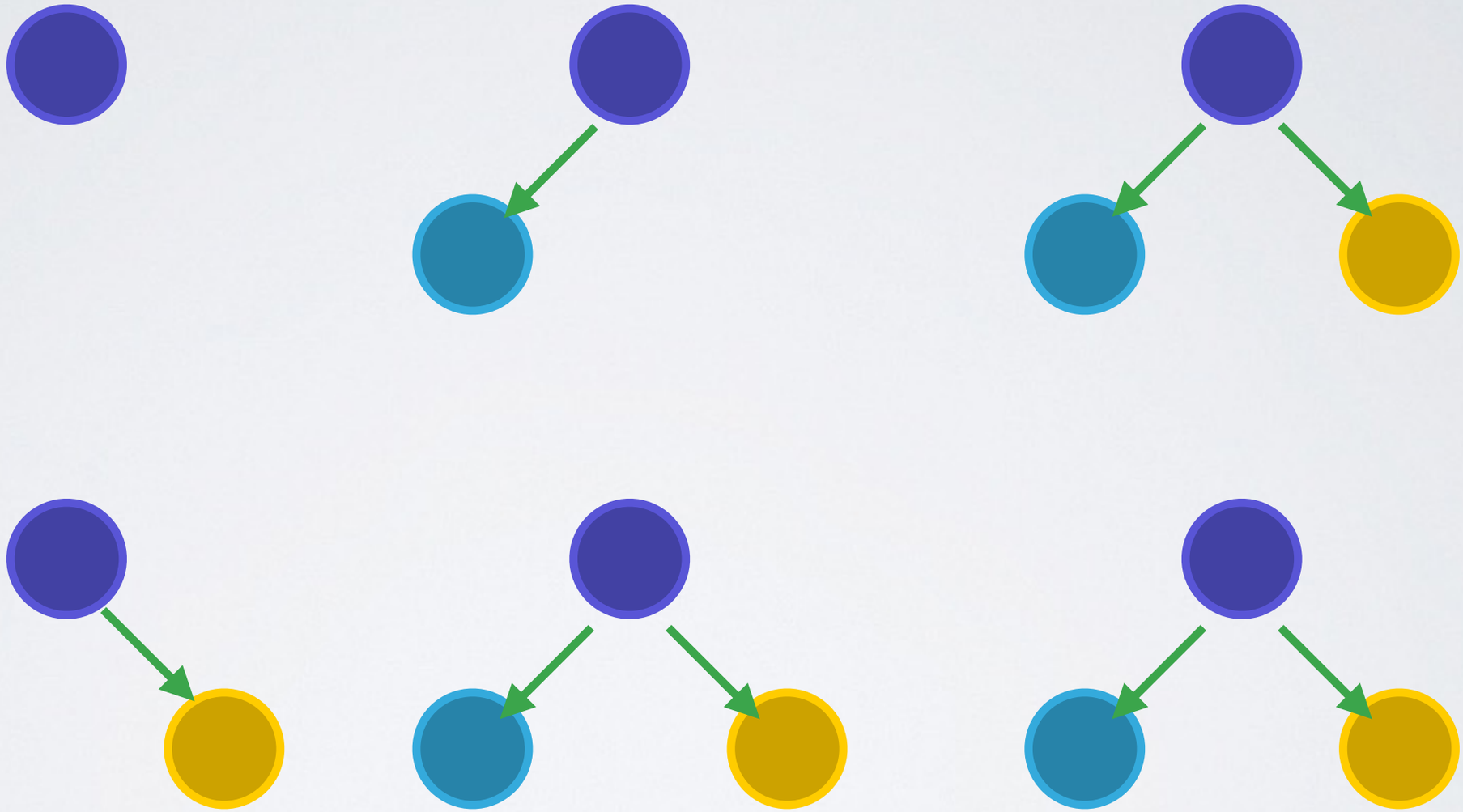
MULTIPLE MVCS



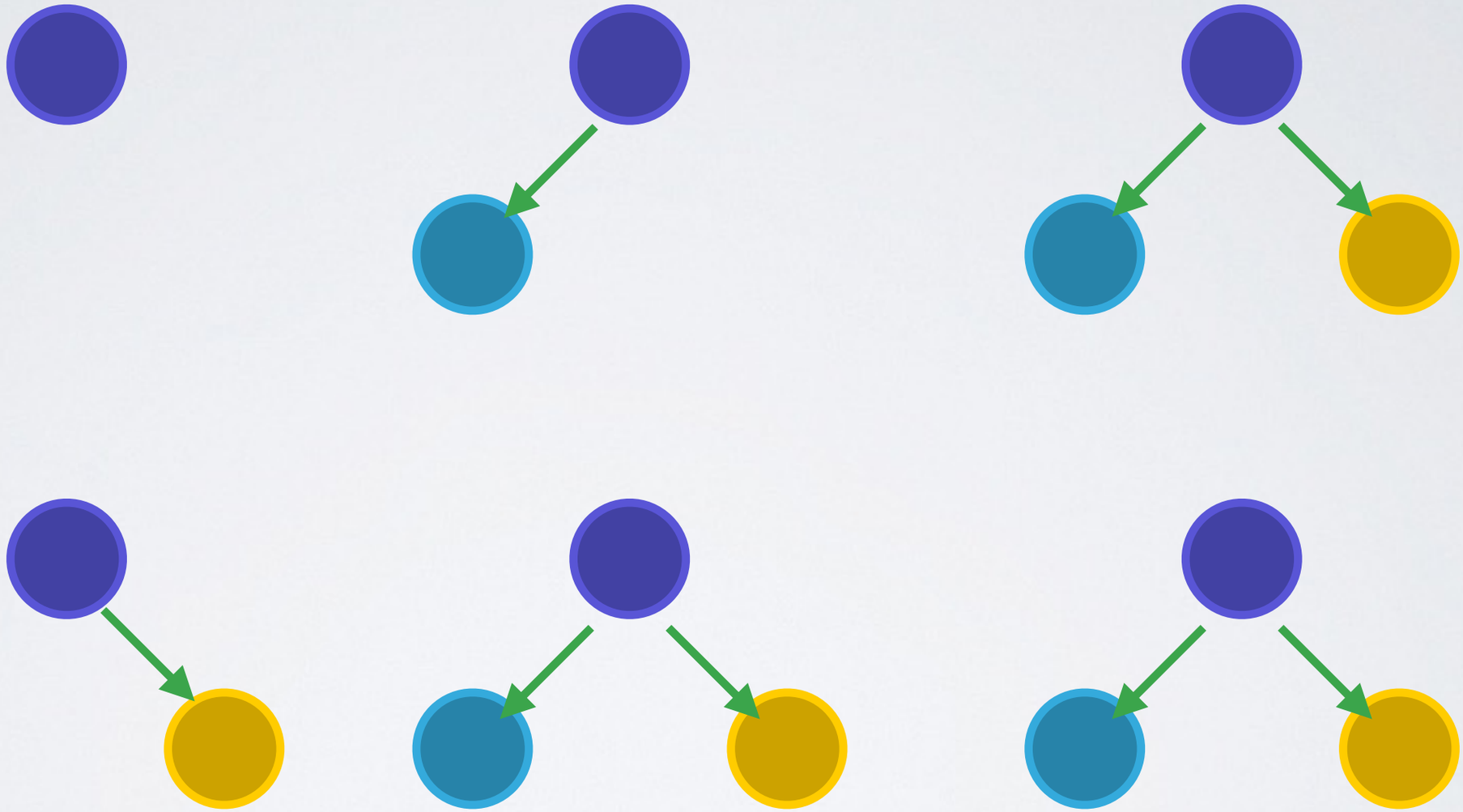
MULTIPLE MVCS



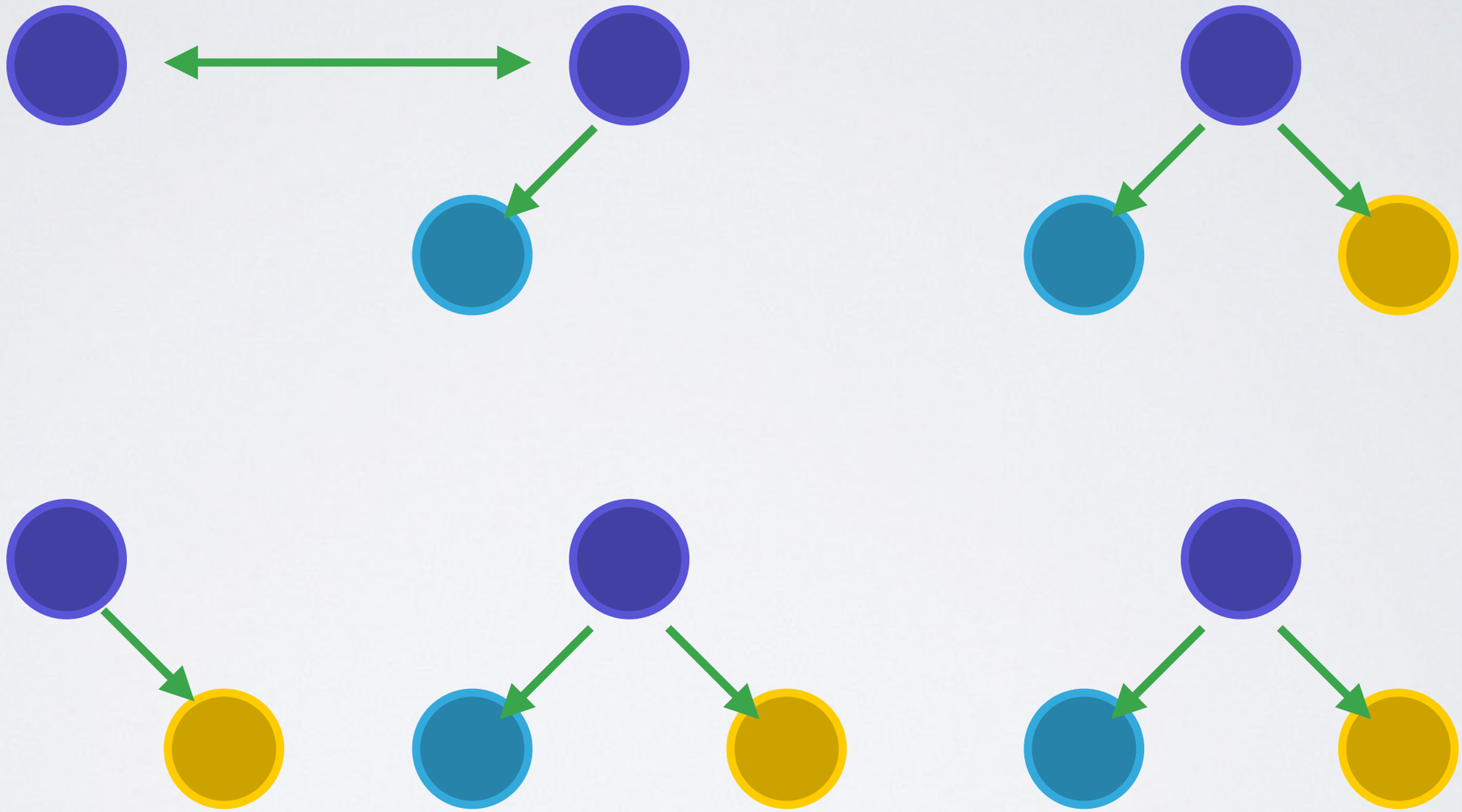
MULTIPLE MVCS



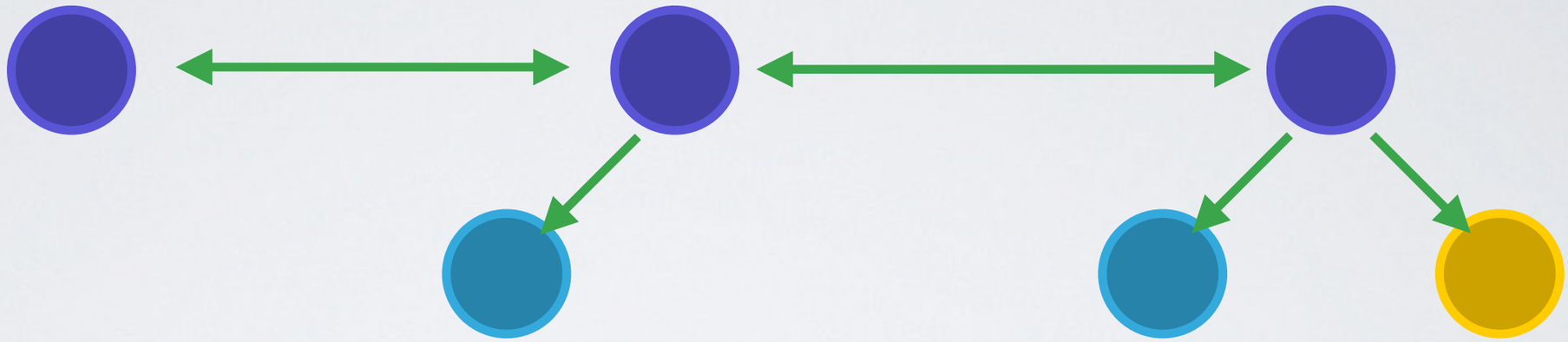
MULTIPLE MVCS



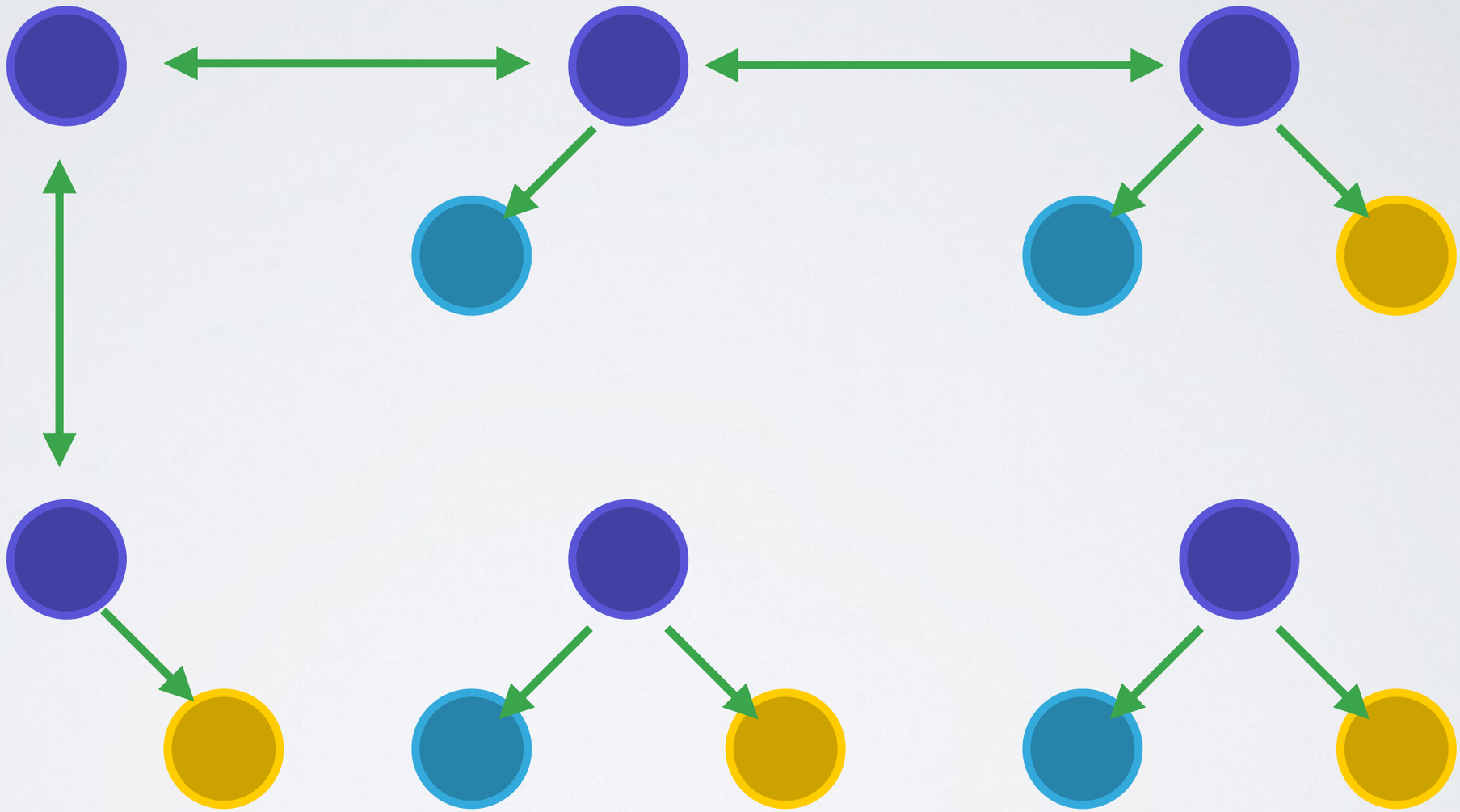
MULTIPLE MVCS



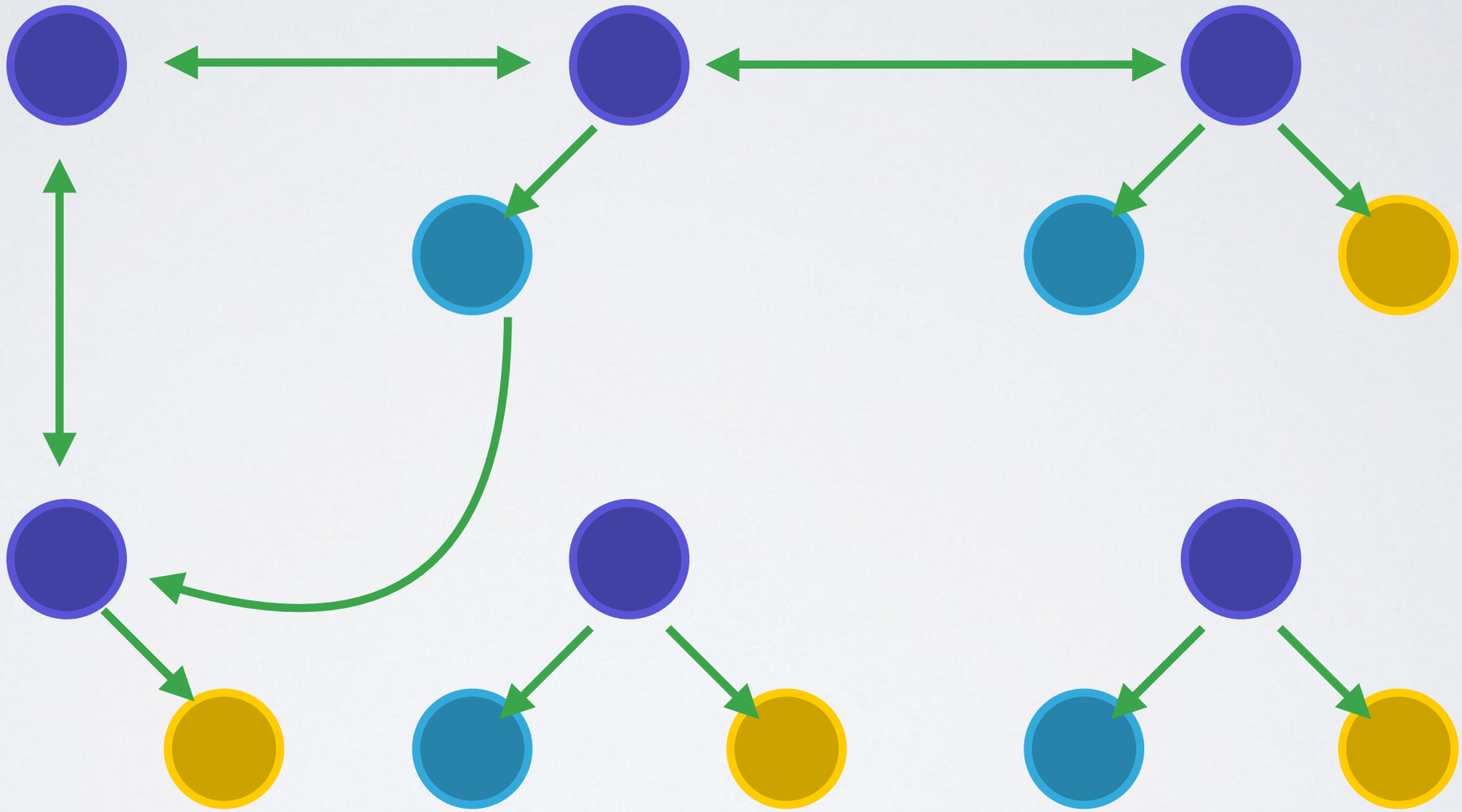
MULTIPLE MVCS



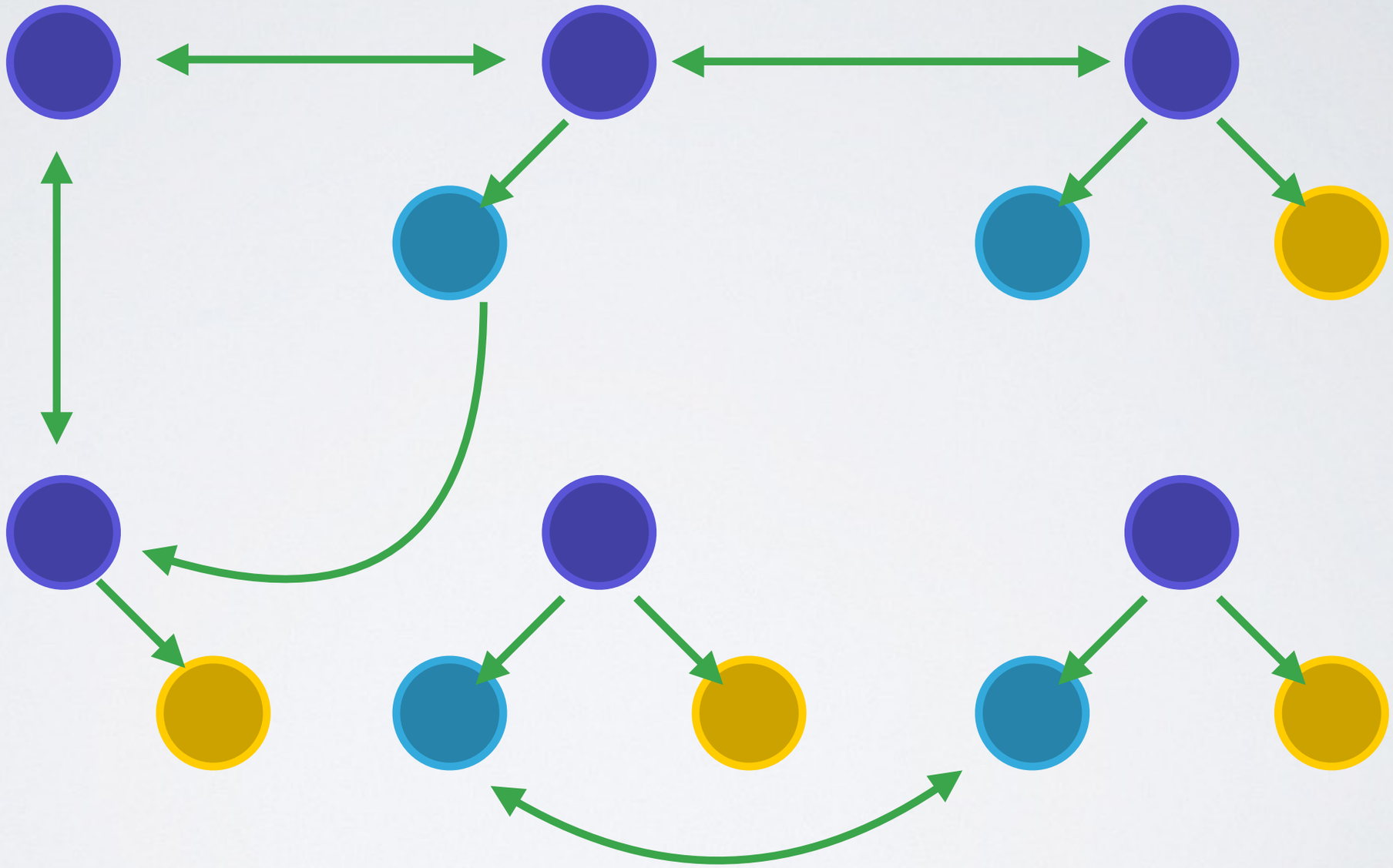
MULTIPLE MVCS



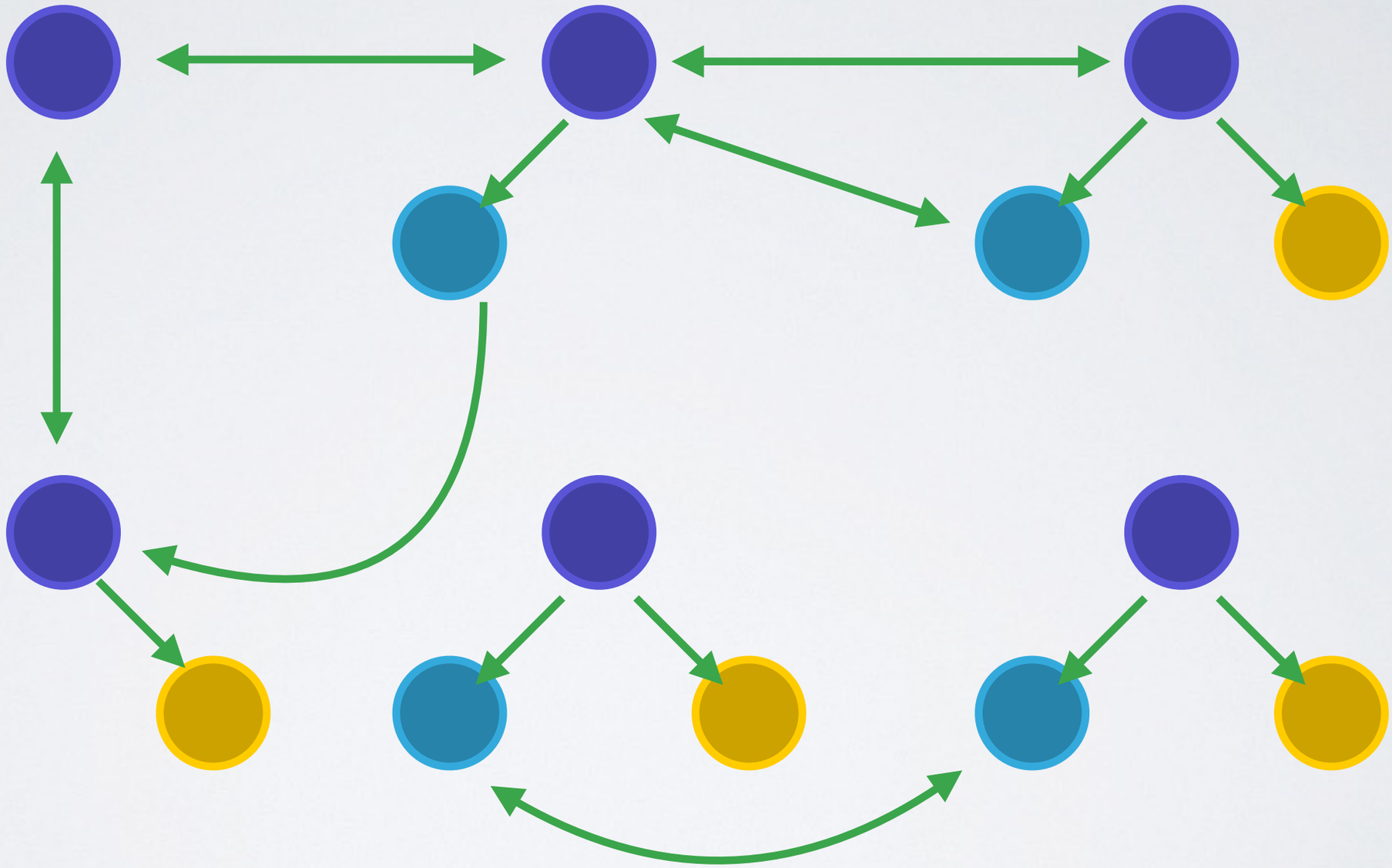
MULTIPLE MVCS



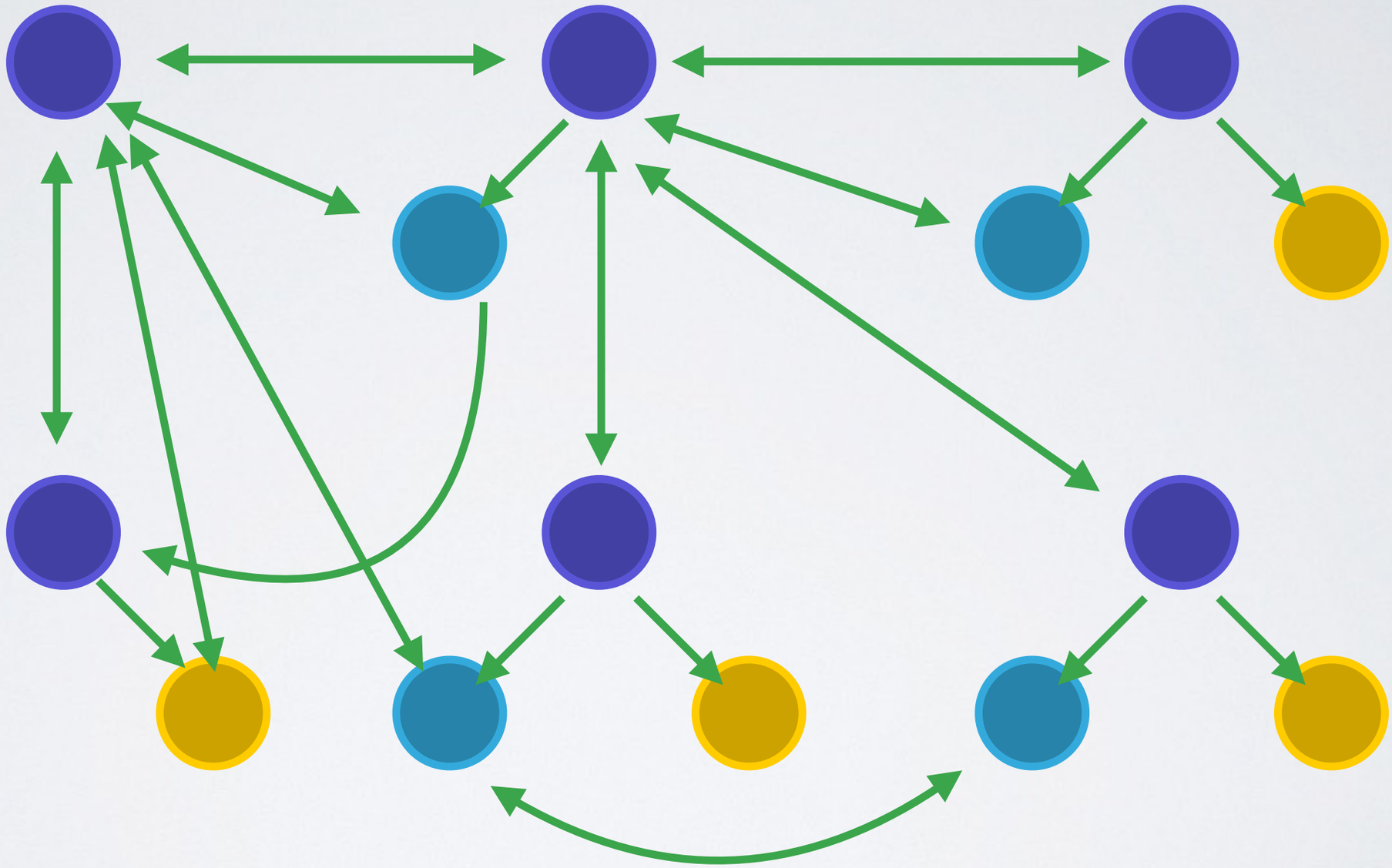
MULTIPLE MVCS



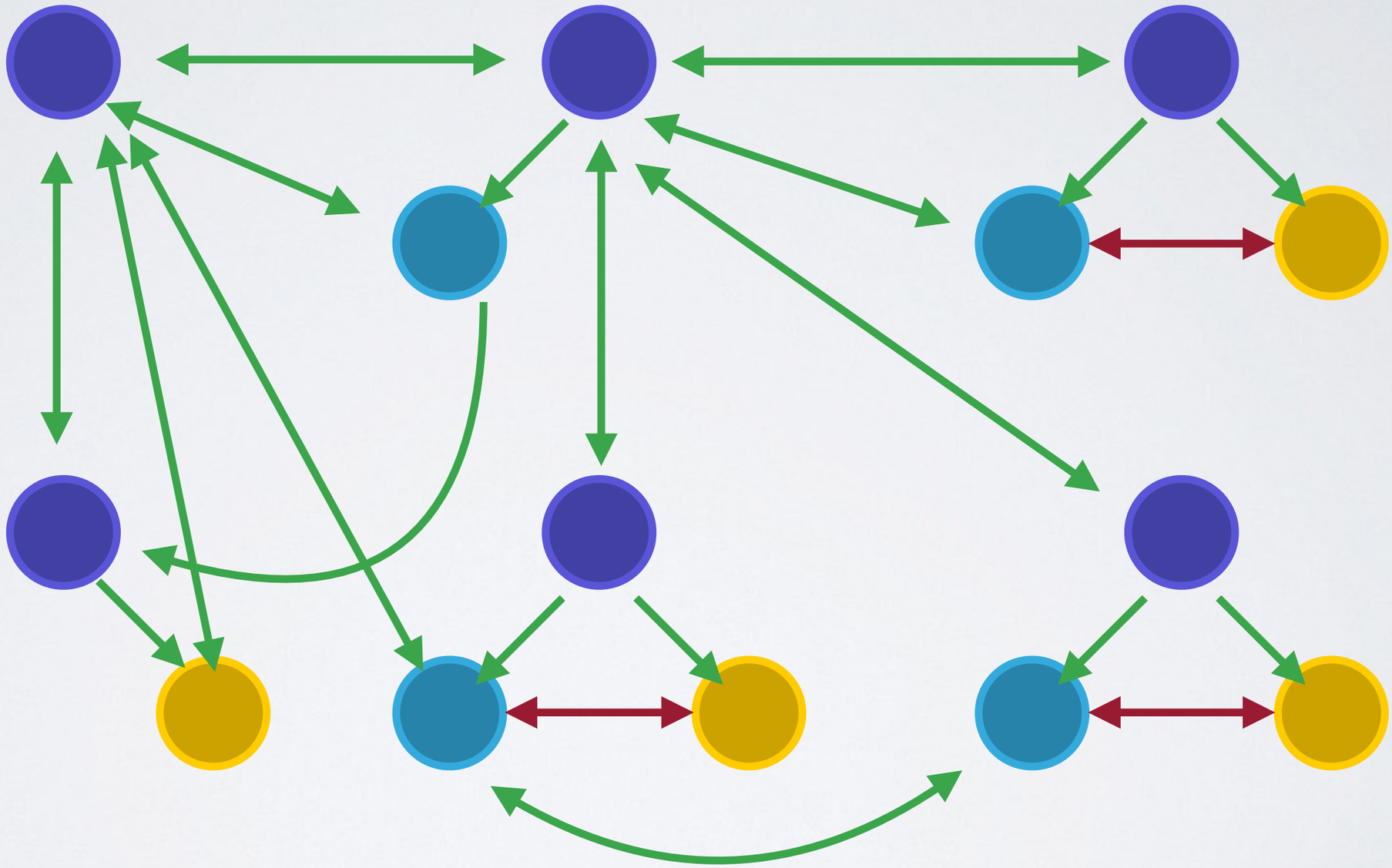
MULTIPLE MVCS



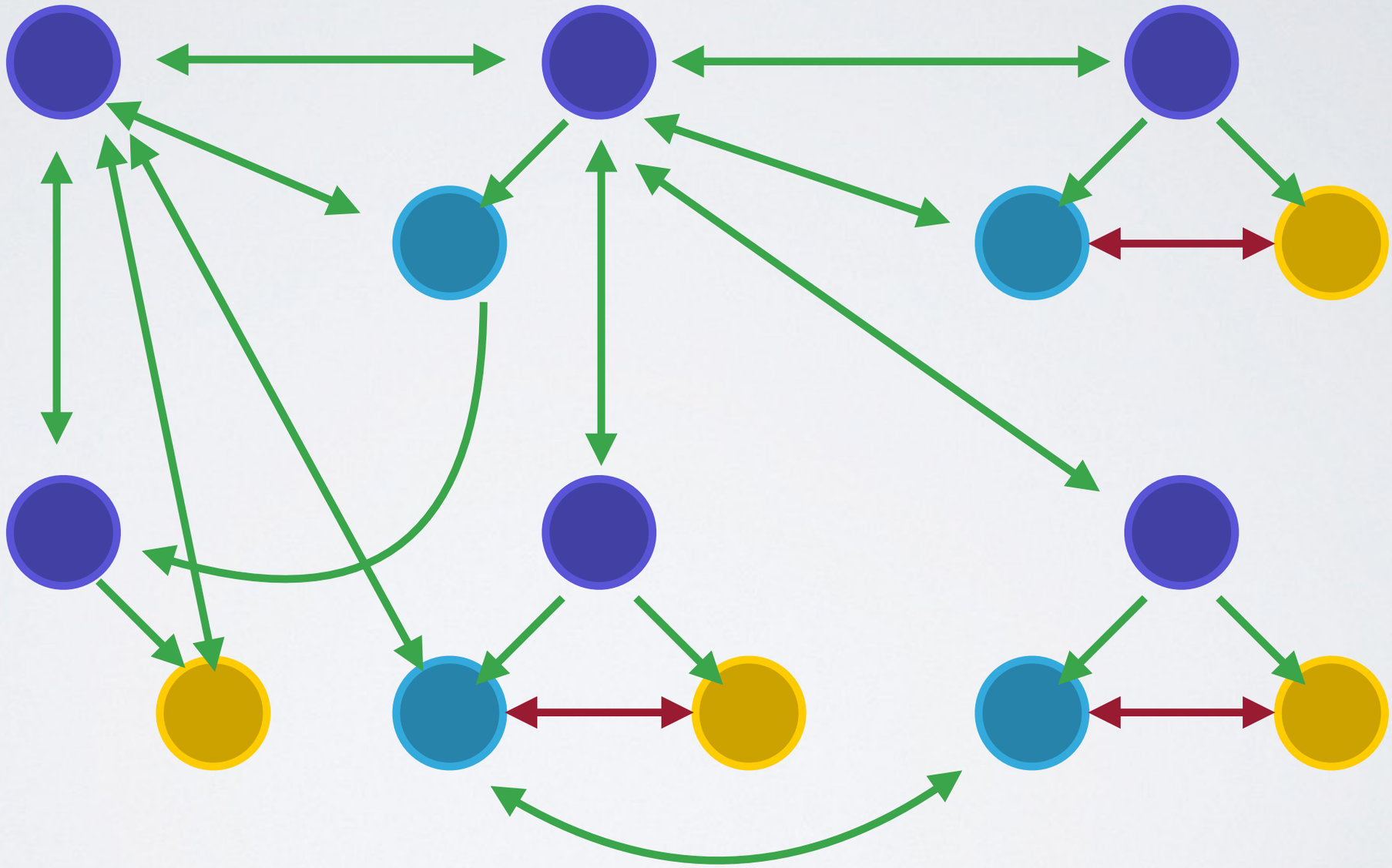
MULTIPLE MVCS



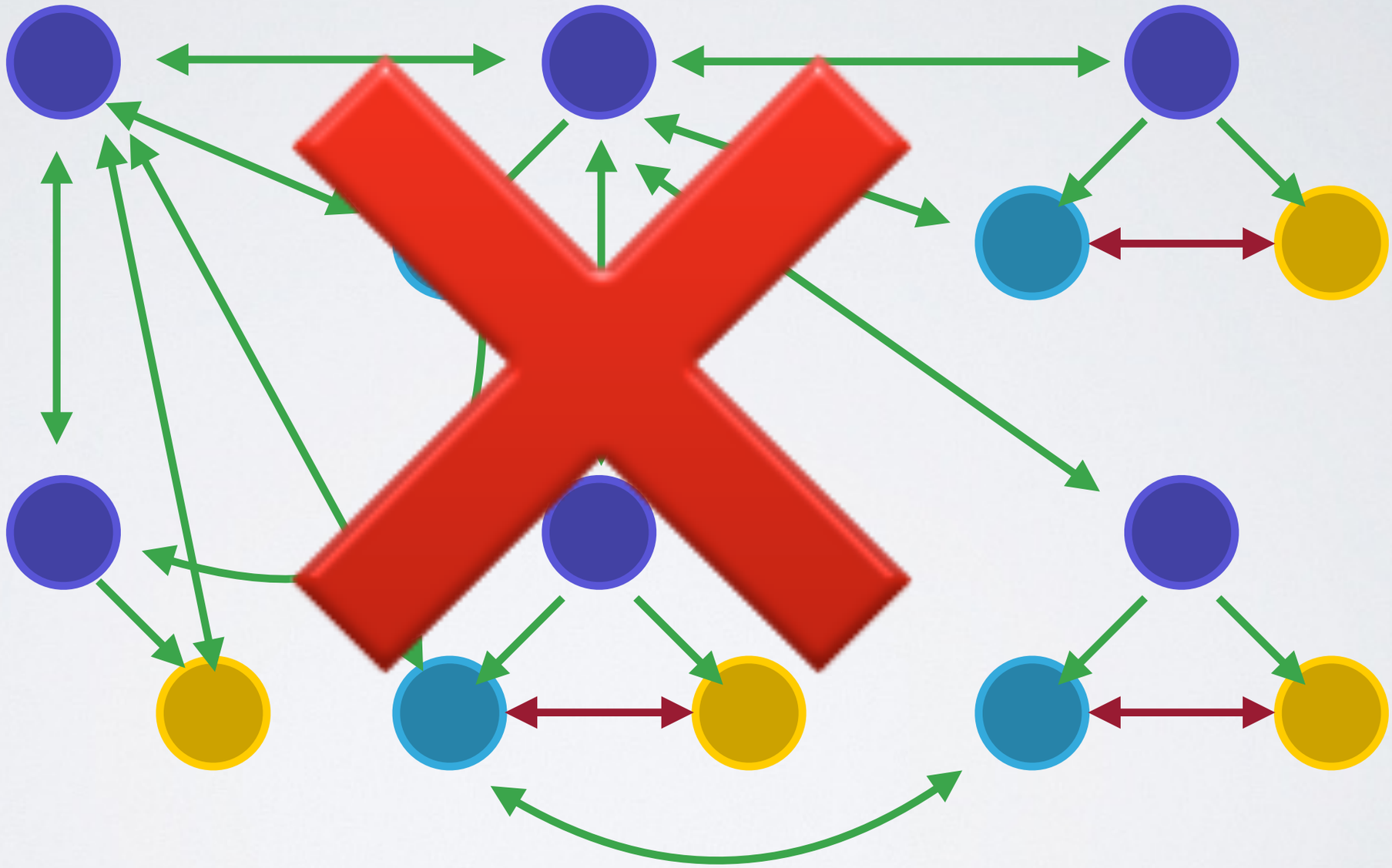
MULTIPLE MVCS



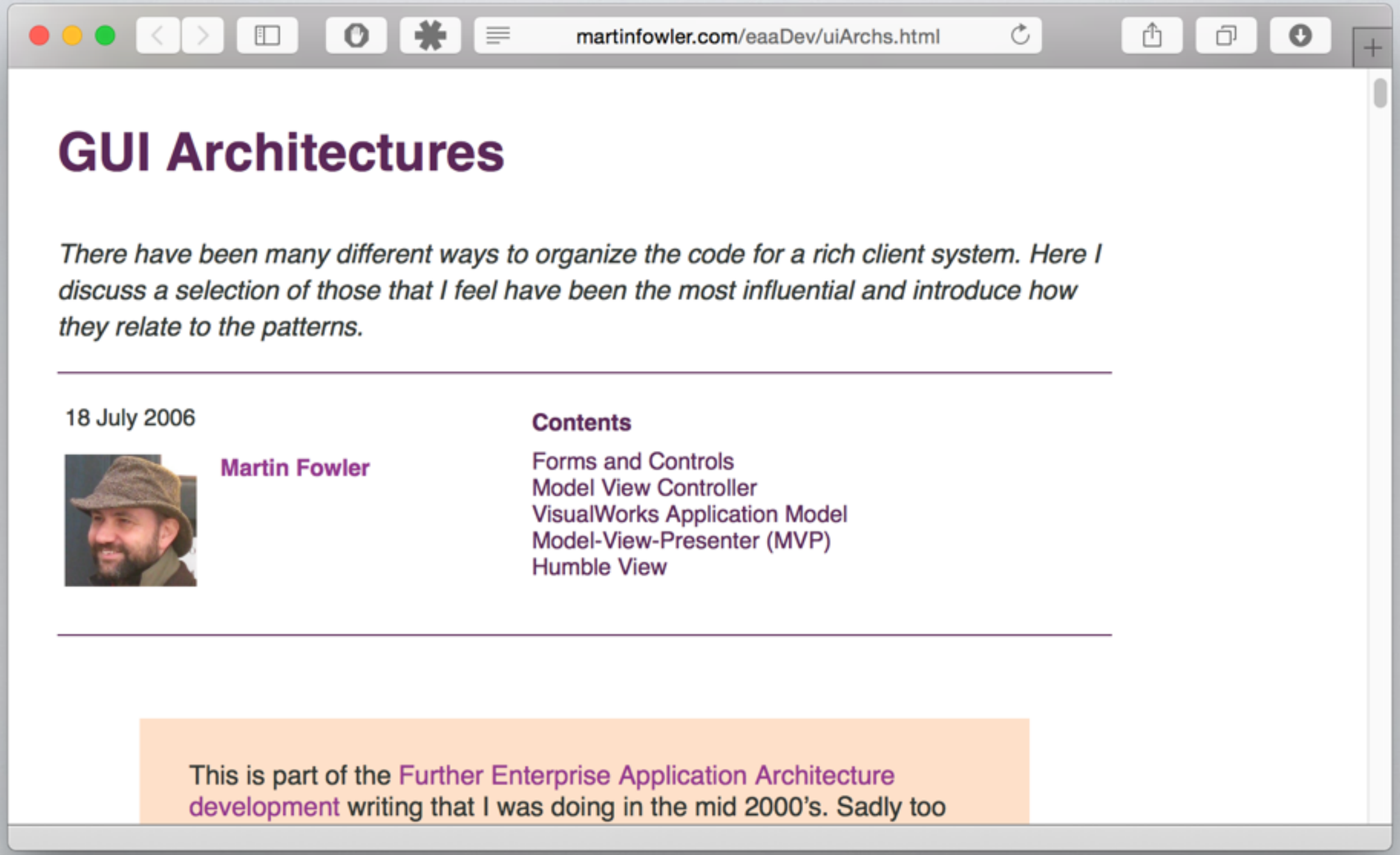
MULTIPLE MVCS FAILING



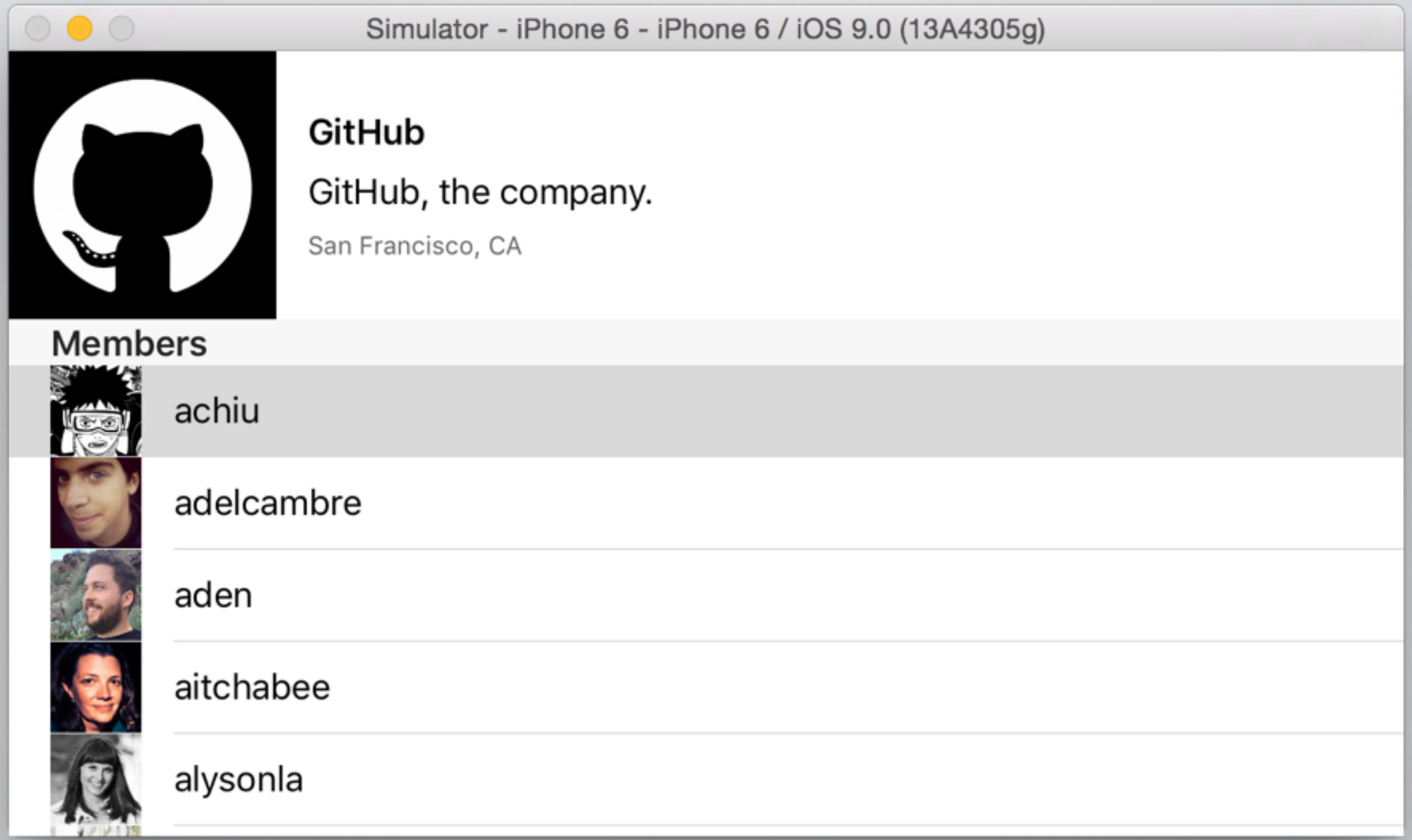
MULTIPLE MVCS FAILING



MVC (DETOUR)


martinfowler.com/eaaDev/uiArchs.html
GUI Architectures
There have been many different ways to organize the code for a rich client system. Here I discuss a selection of those that I feel have been the most influential and introduce how they relate to the patterns.
18 July 2006
Martin Fowler
Contents
Forms and Controls
Model View Controller
VisualWorks Application Model
Model-View-Presenter (MVP)
Humble View
This is part of the Further Enterprise Application Architecture development writing that I was doing in the mid 2000's. Sadly too

LET'S BUILD THIS USING MVC



VIEW

- Let's prototype the design in Storyboards first.

- Get a better feel for table views and auto-layouts

TABLE VIEW (STATIC)

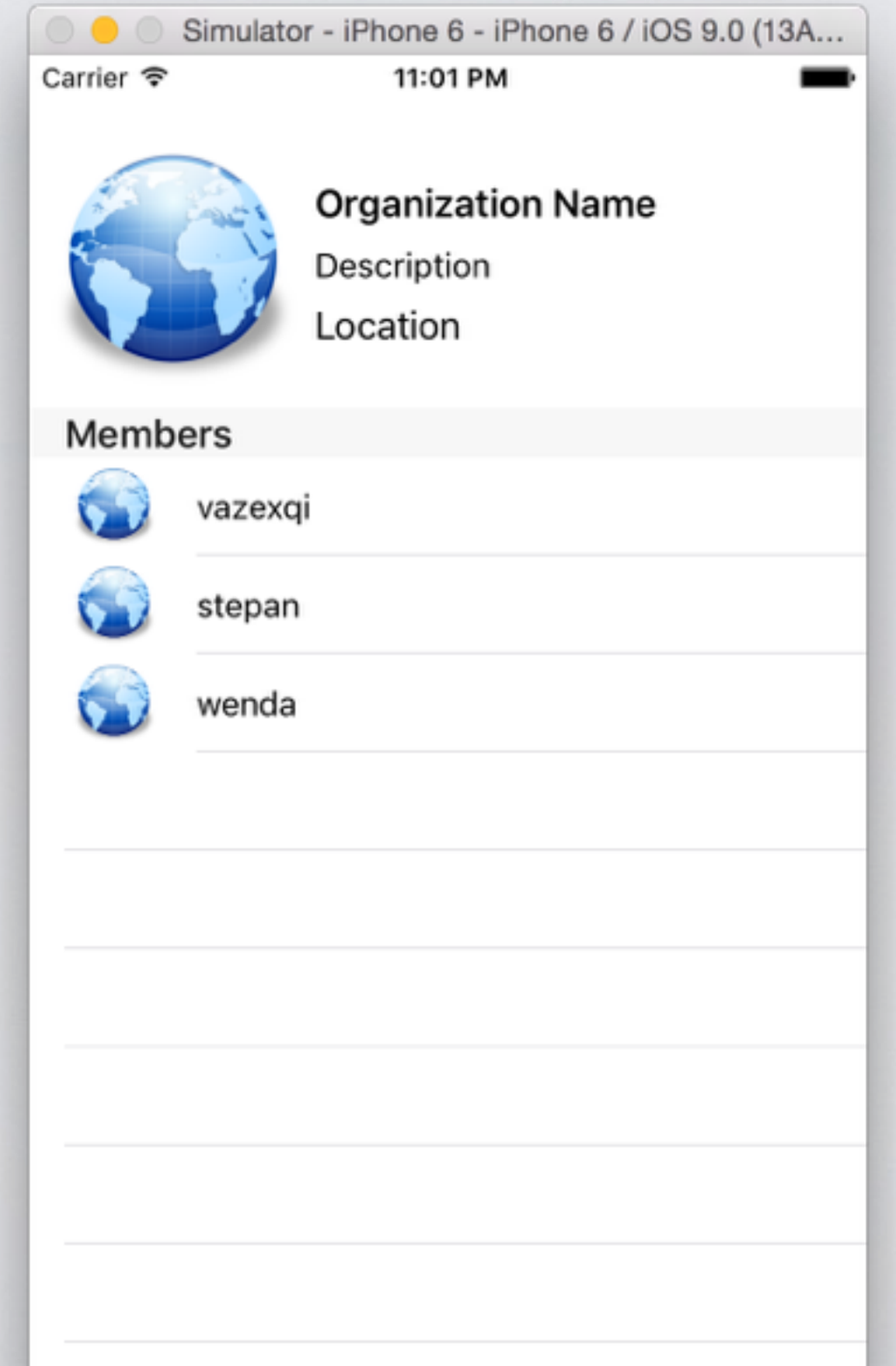


TABLE VIEW (STATIC)

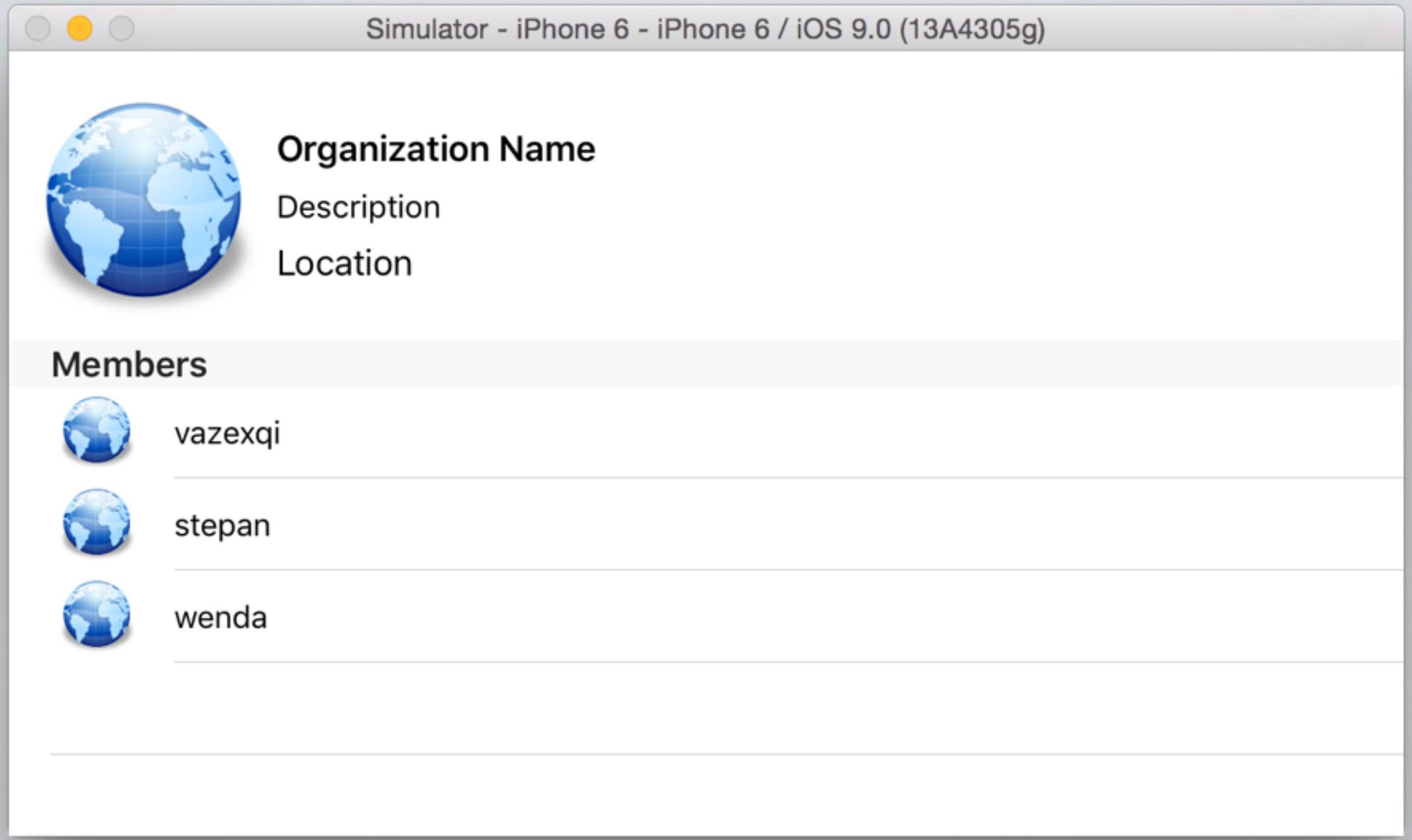


TABLE VIEW (STATIC)

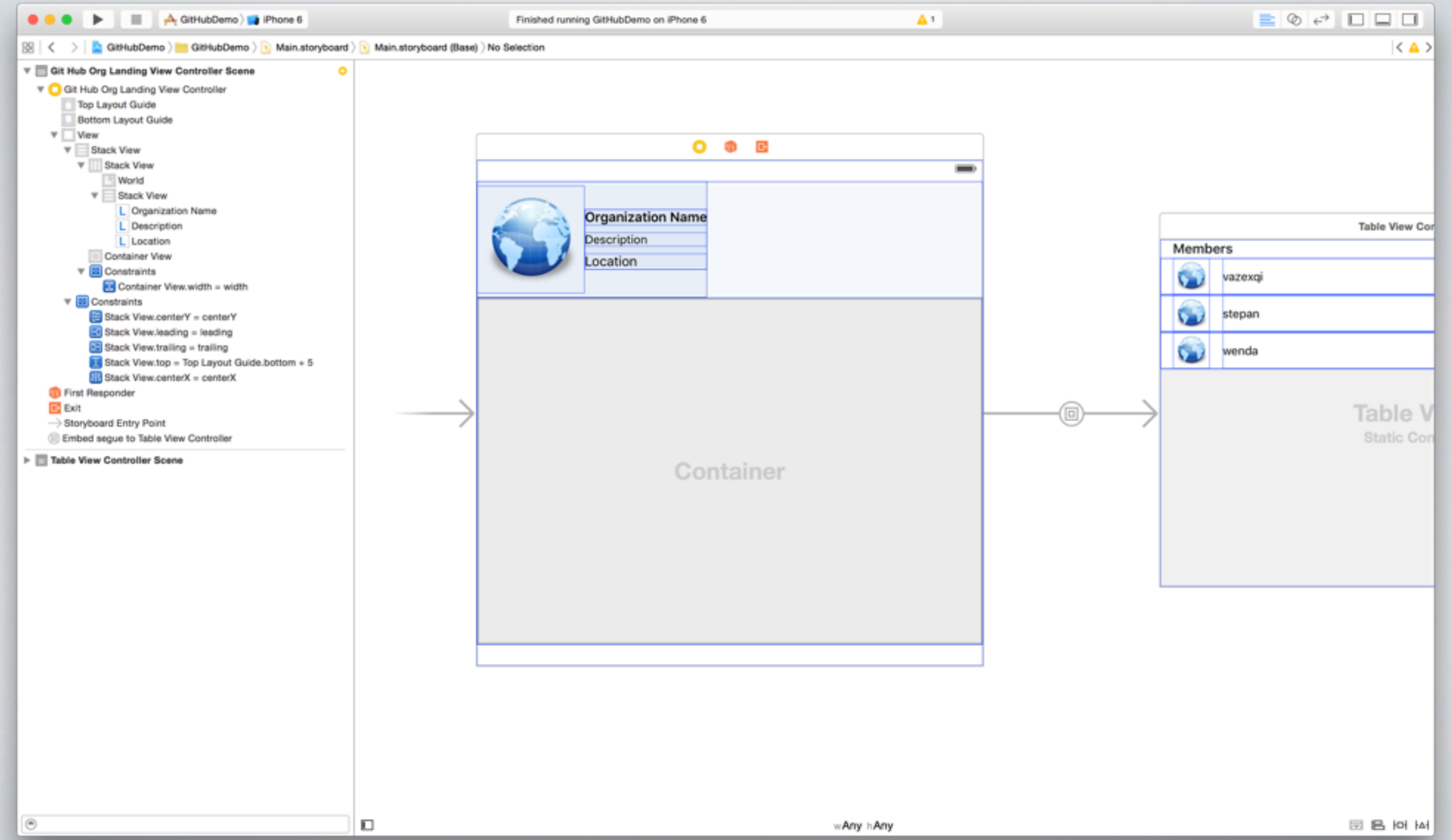


TABLE VIEW (STATIC)

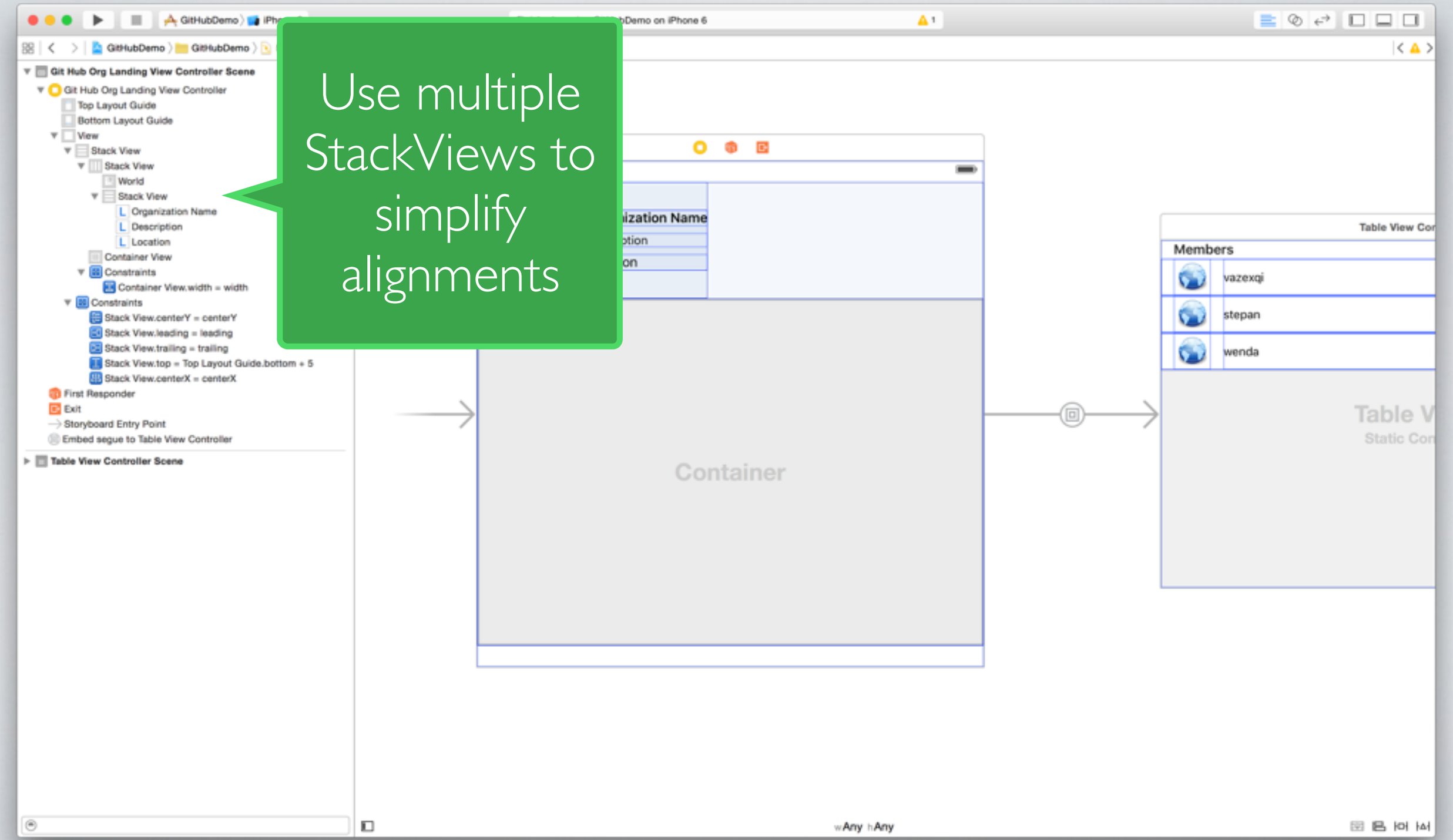


TABLE VIEW (STATIC)

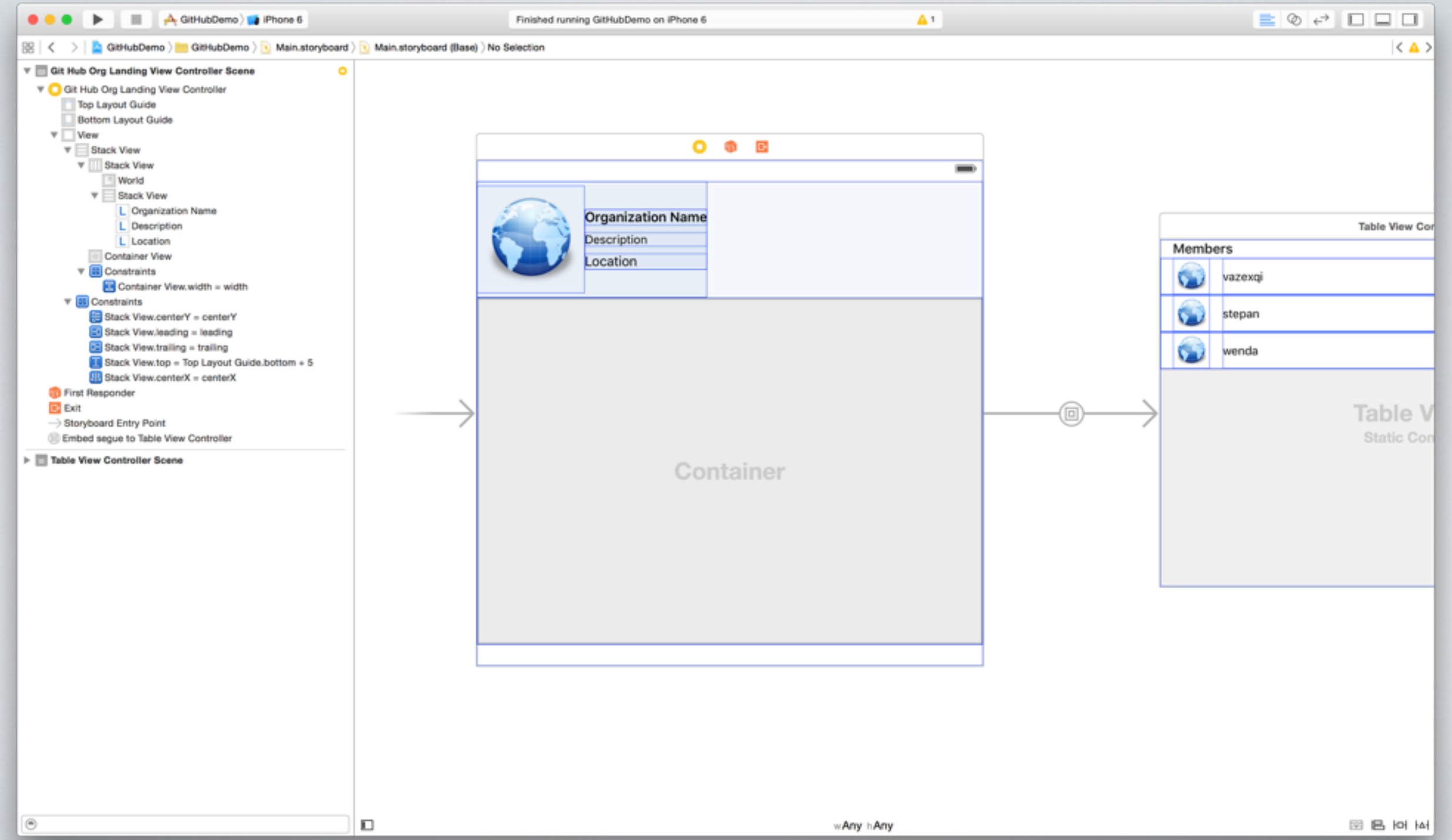


TABLE VIEW (STATIC)

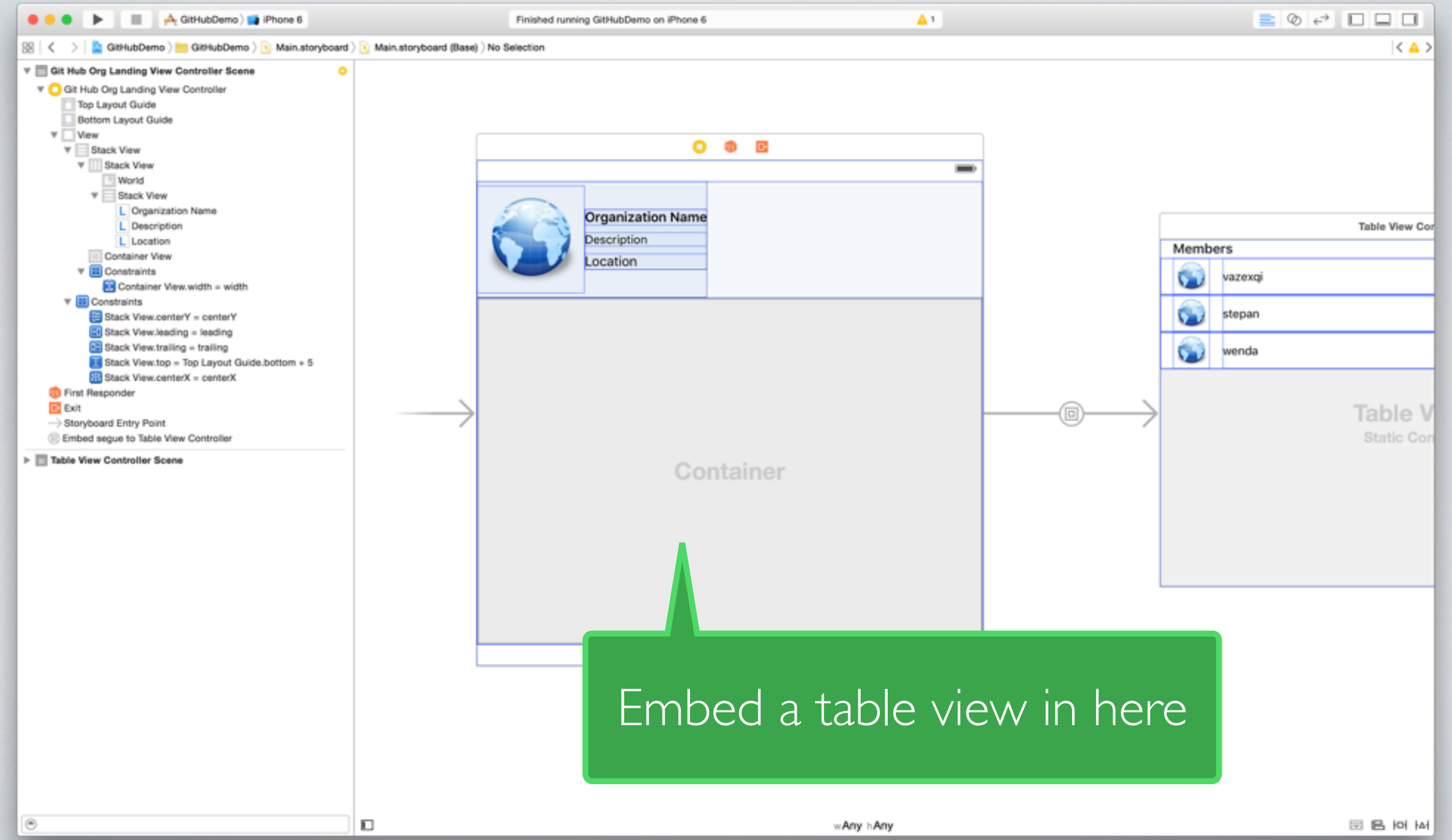


TABLE VIEW (STATIC)

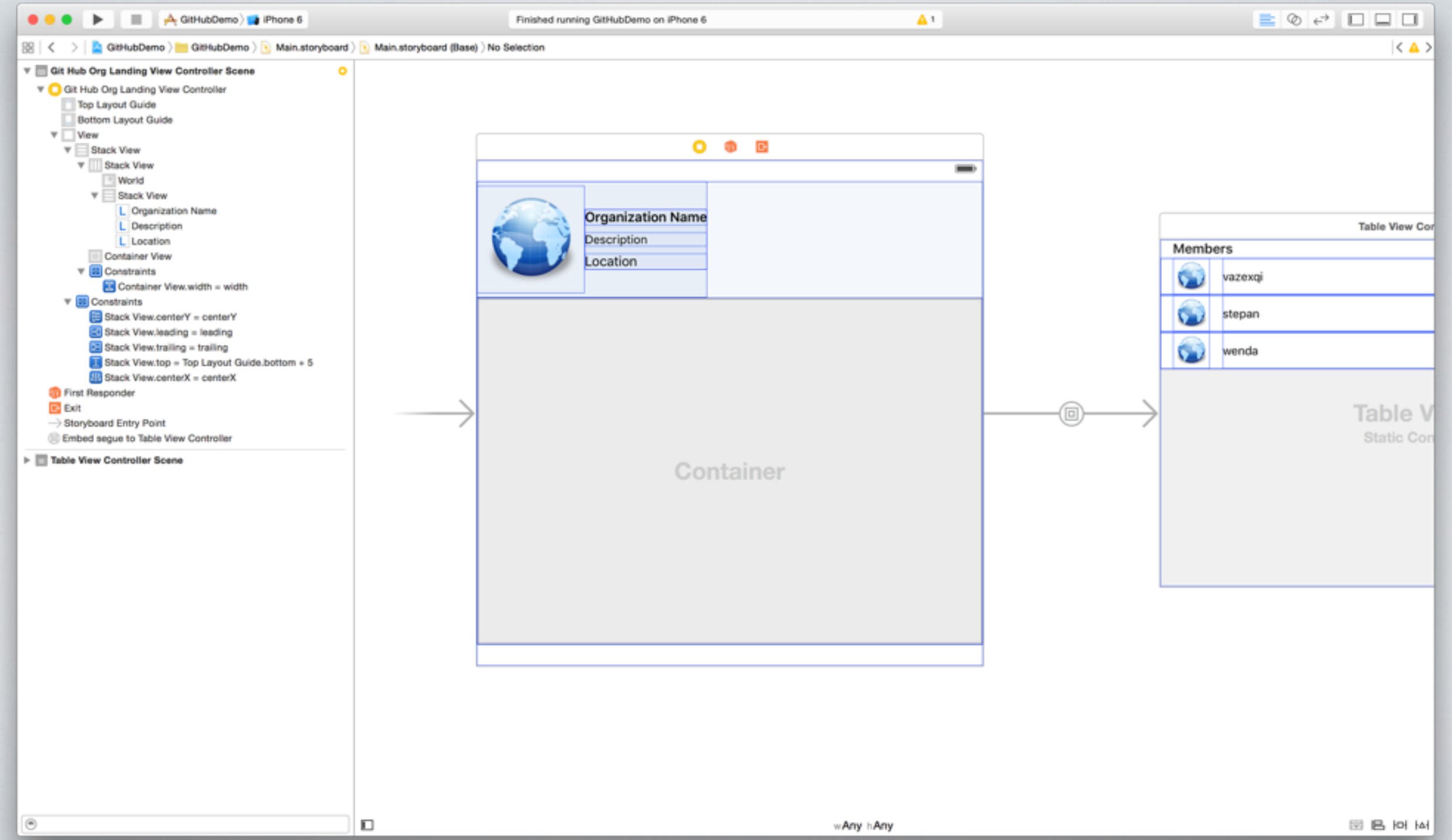
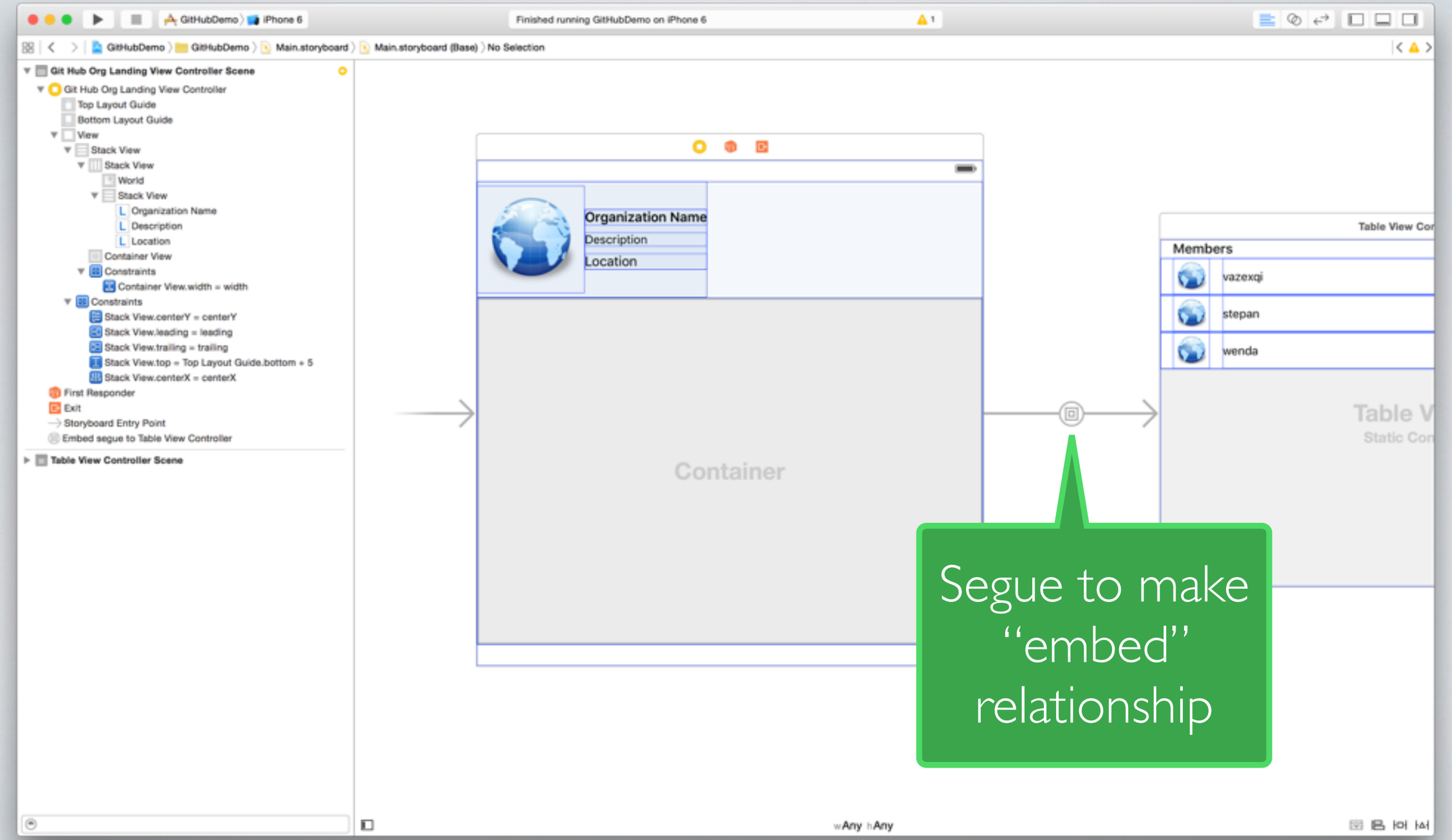


TABLE VIEW (STATIC)

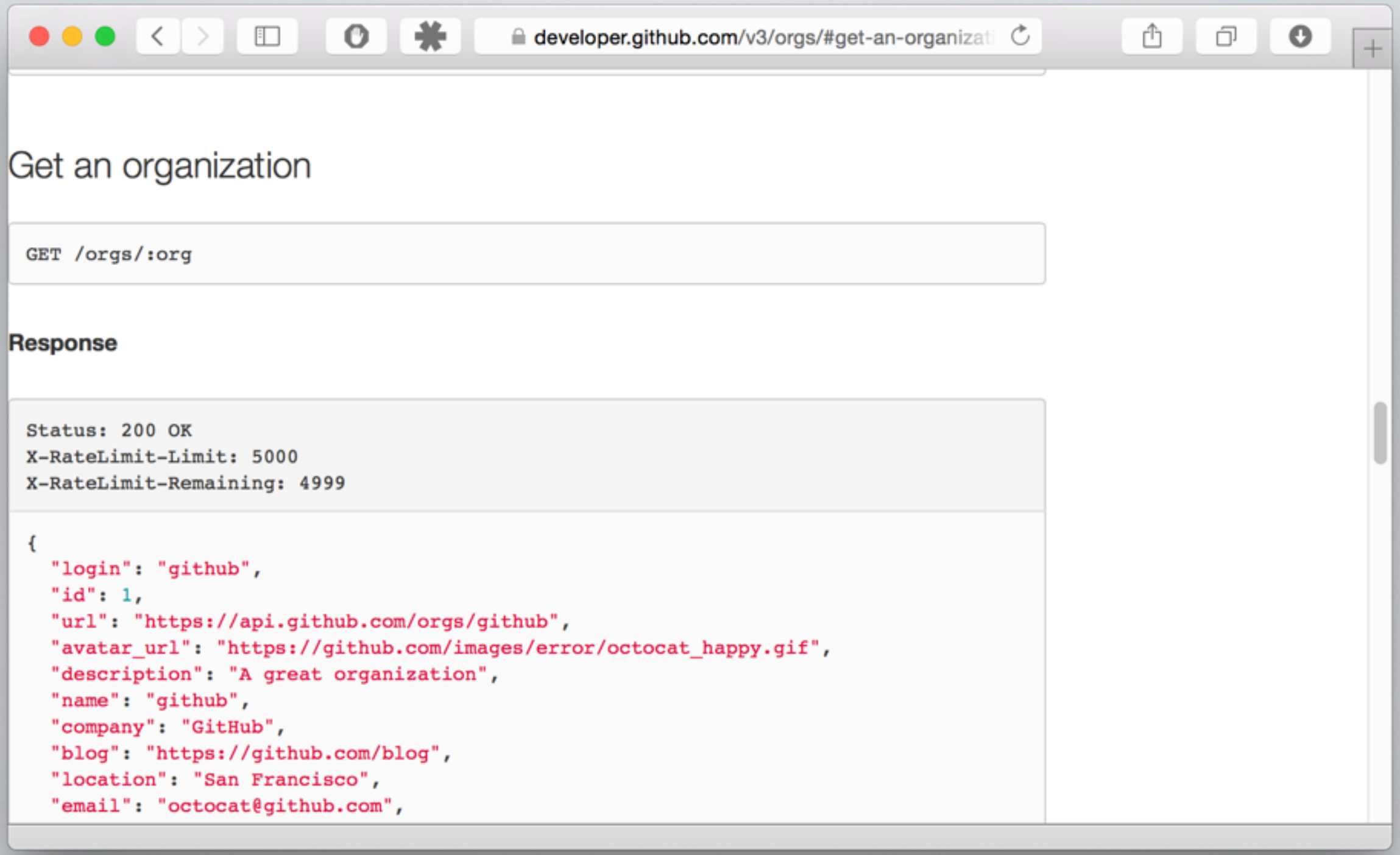


HANDS-ON

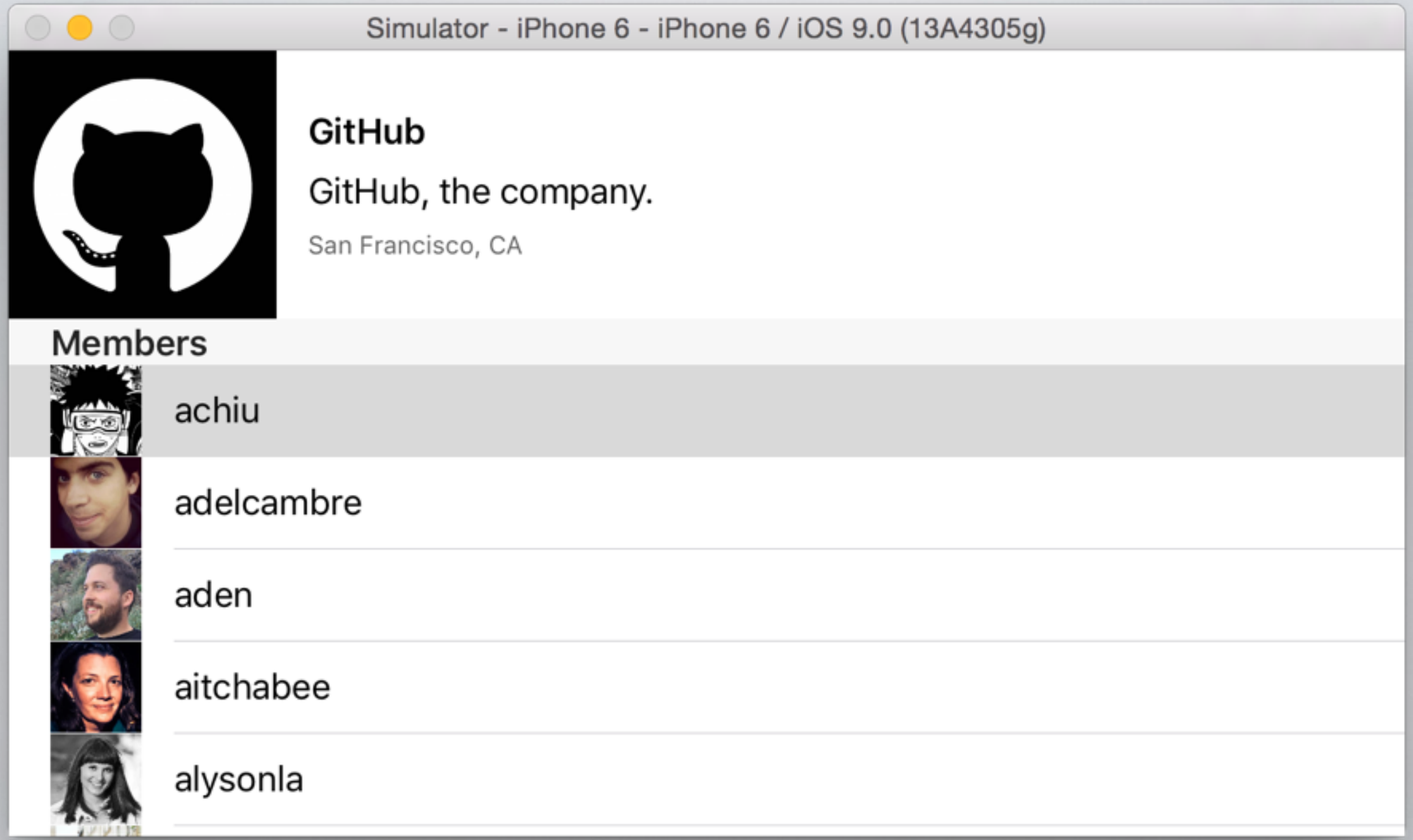
MODEL

We need some way to get the data from GitHub

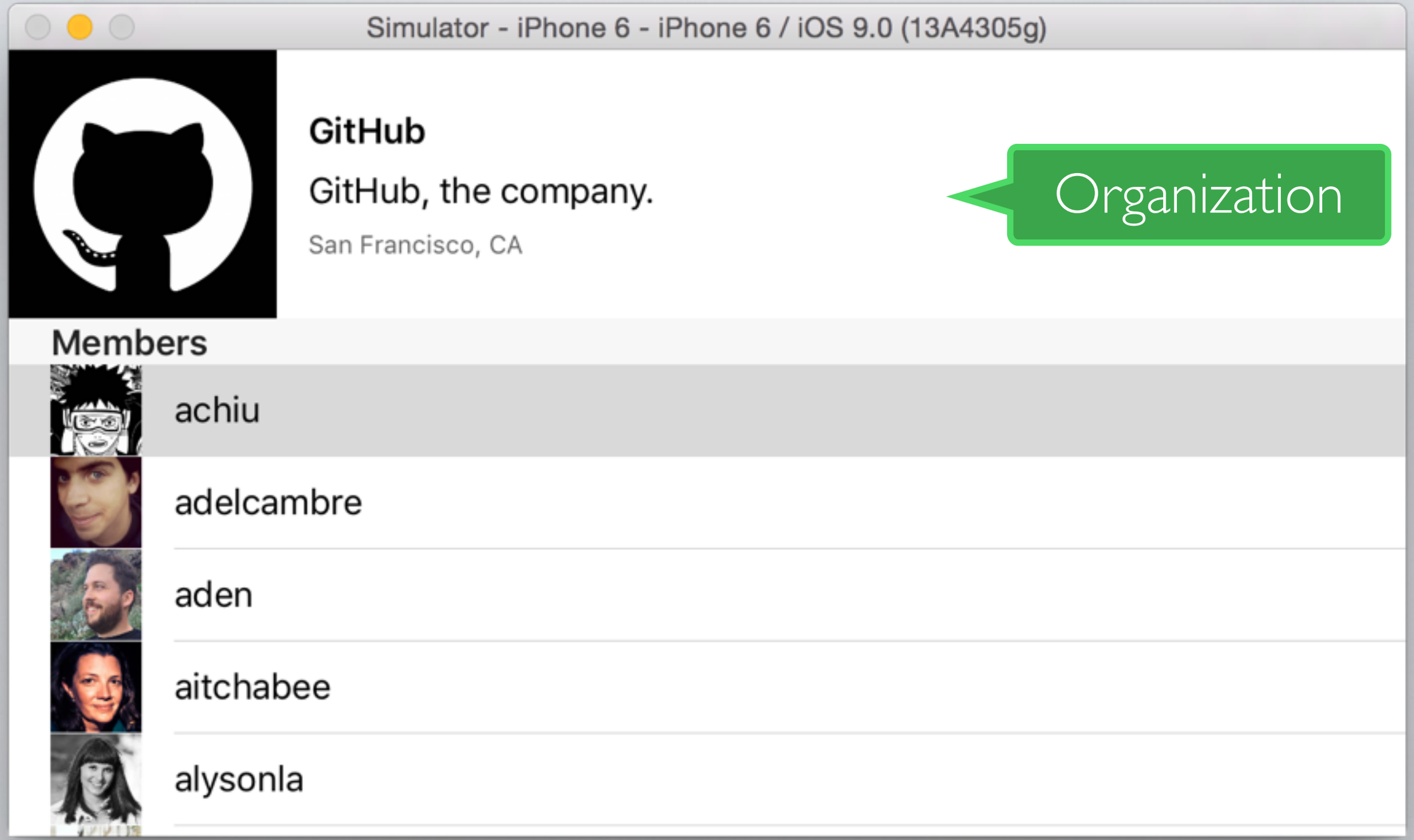
GITHUB API


developer.github.com/v3/orgs/#get-an-organizat
Get an organization
GET /orgs/:org
Response
Status: 200 OK
X-RateLimit-Limit: 5000
X-RateLimit-Remaining: 4999
{
"login": "github",
"id": 1,
"url": "https://api.github.com/orgs/github",
"avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization",
"name": "github",
"company": "GitHub",
"blog": "https://github.com/blog",
"location": "San Francisco",
"email": "octocat@github.com",

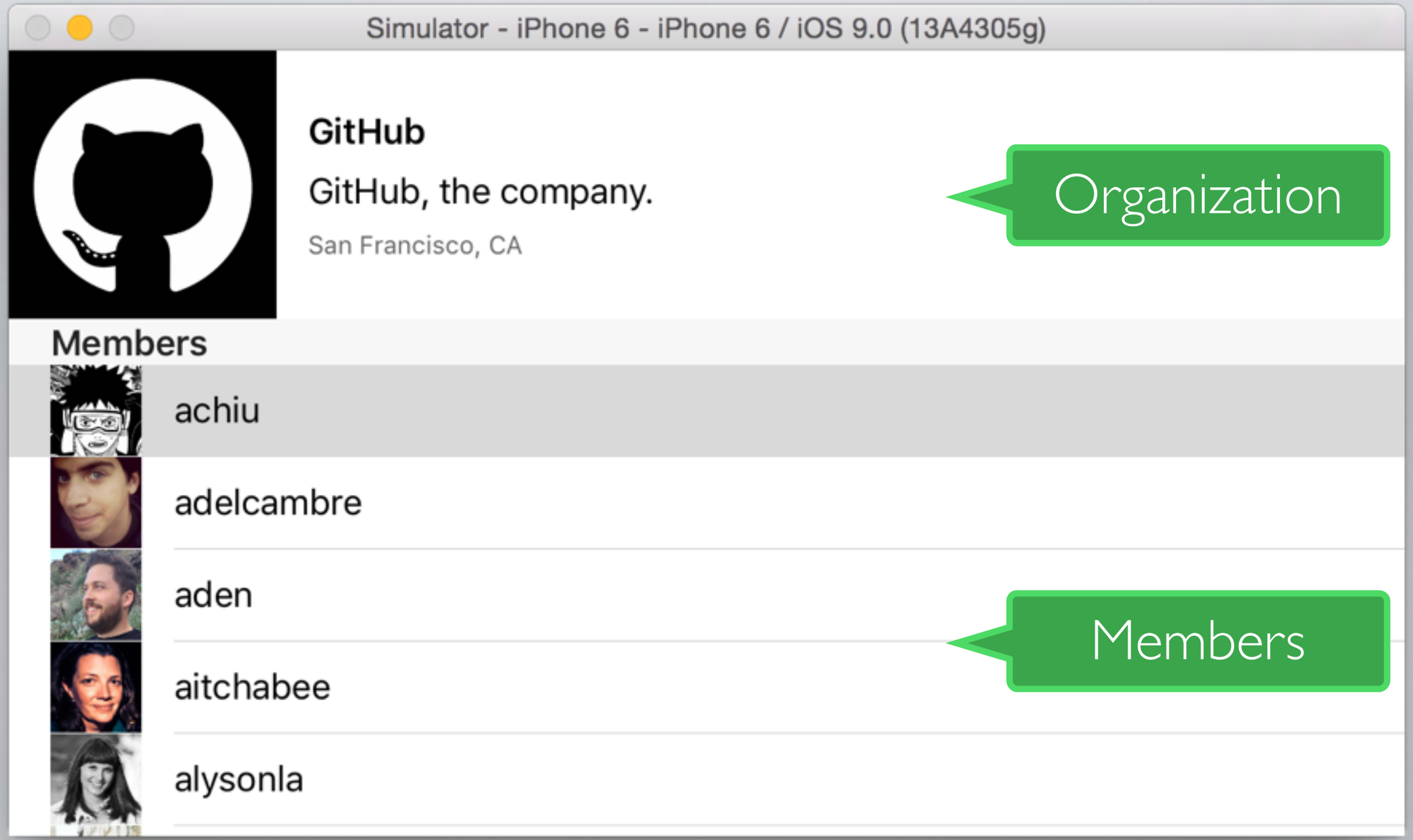
GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Members
achiu
adelcambre
aden
aitchabee
alysonla

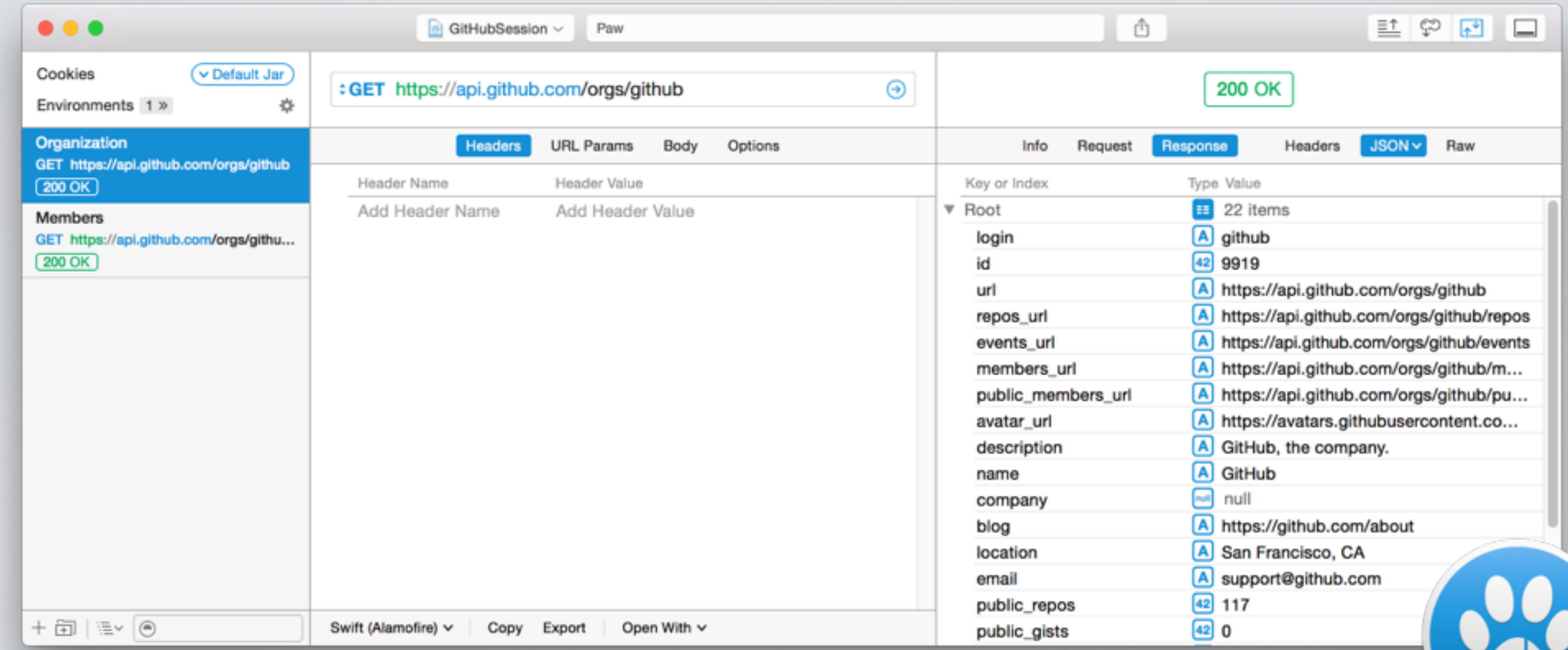
GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Organization
Members
achiu
adelcambre
aden
aitchabee
alysonla

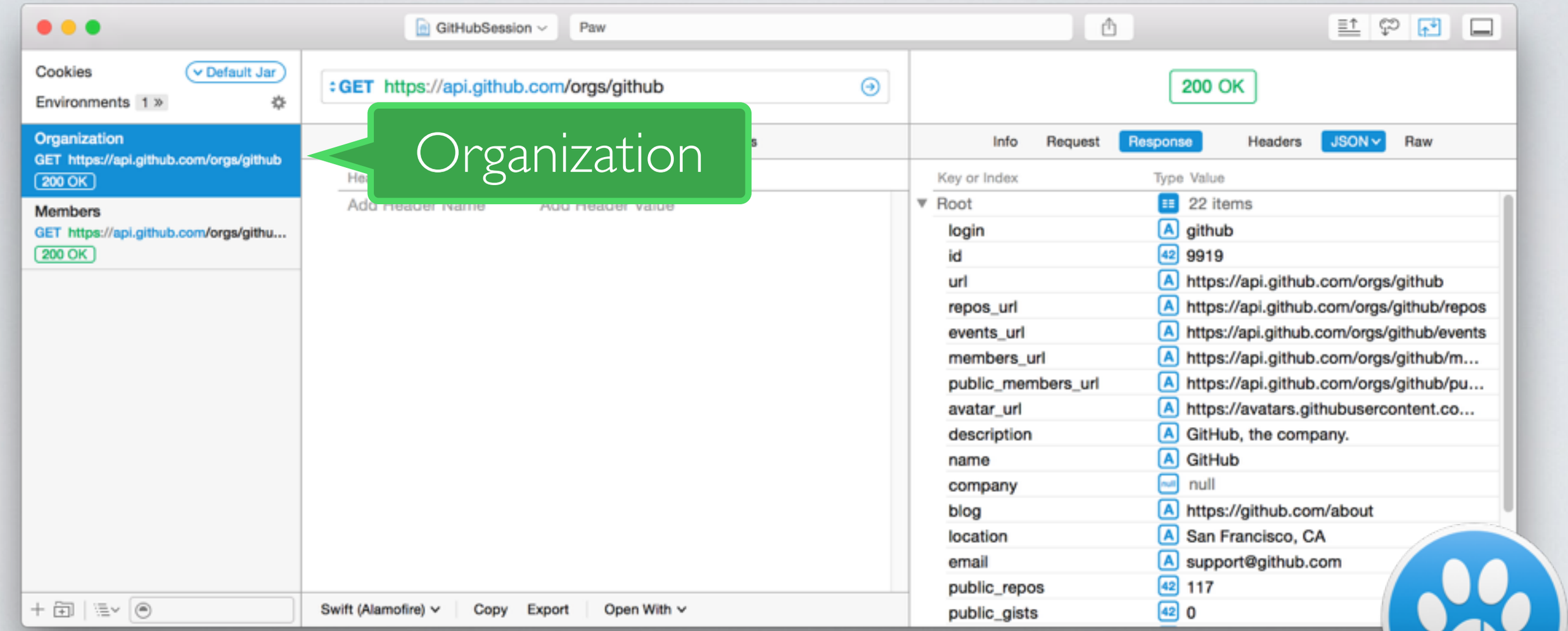
GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Organization
Members
achiu
adelcambre
aden
aitchabee
alysonla
Members

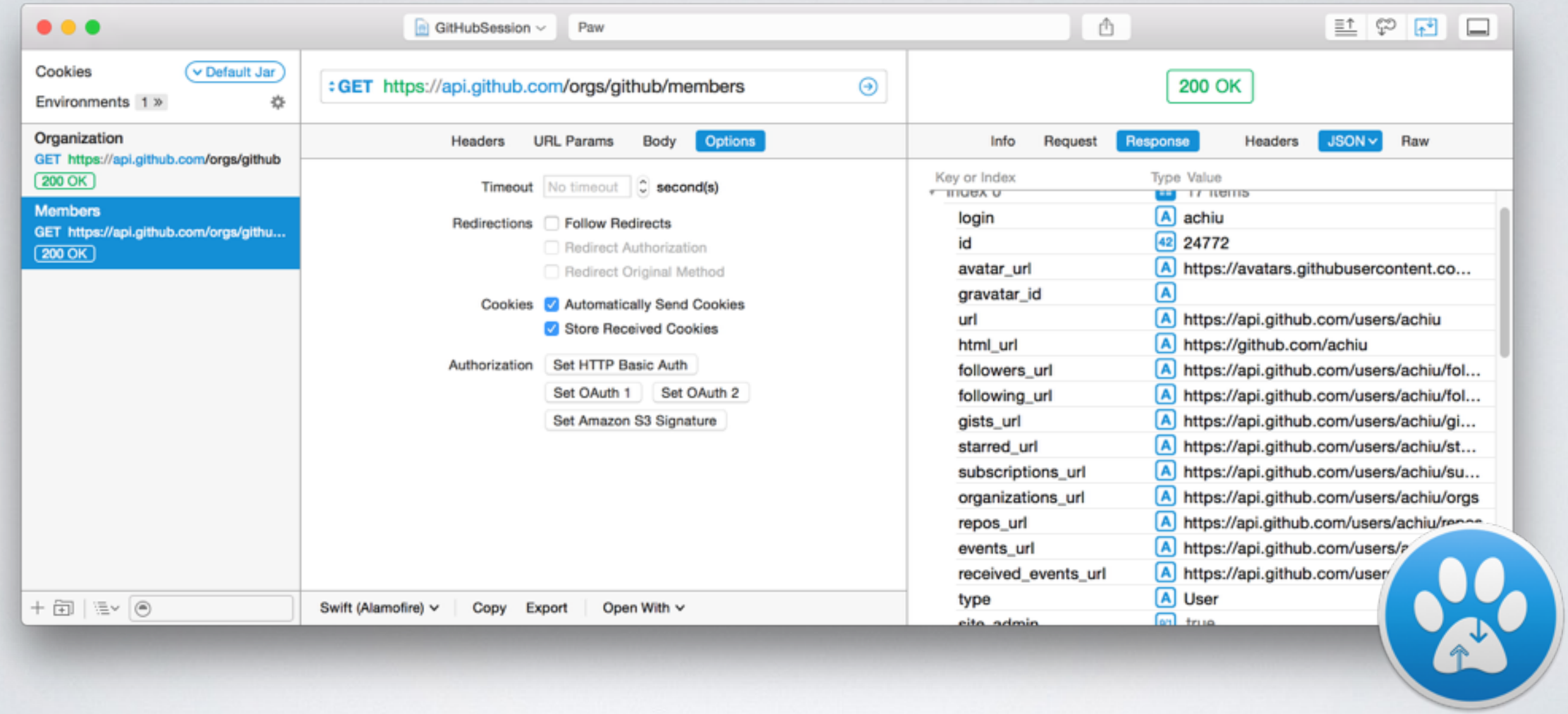
GITHUB API


GitHubSession
Paw
Cookies
Default Jar
Environments 1
Organization
GET https://api.github.com/orgs/github
200 OK
Members
GET https://api.github.com/orgs/githu...
200 OK
GET https://api.github.com/orgs/github
Headers
URL Params
Body
Options
Header Name
Header Value
Add Header Name
Add Header Value
Swift (Alamofire)
Copy
Export
Open With
200 OK
Info
Request
Response
Headers
JSON
Raw
Key or Index
Type
Value
Root
22 items
login
github
id
9919
url
https://api.github.com/orgs/github
repos_url
https://api.github.com/orgs/github/repos
events_url
https://api.github.com/orgs/github/events
members_url
https://api.github.com/orgs/github/m...
public_members_url
https://api.github.com/orgs/github/pu...
avatar_url
https://avatars.githubusercontent.co...
description
GitHub, the company.
name
GitHub
company
null
blog
https://github.com/about
location
San Francisco, CA
email
support@github.com
public_repos
117
public_gists
0

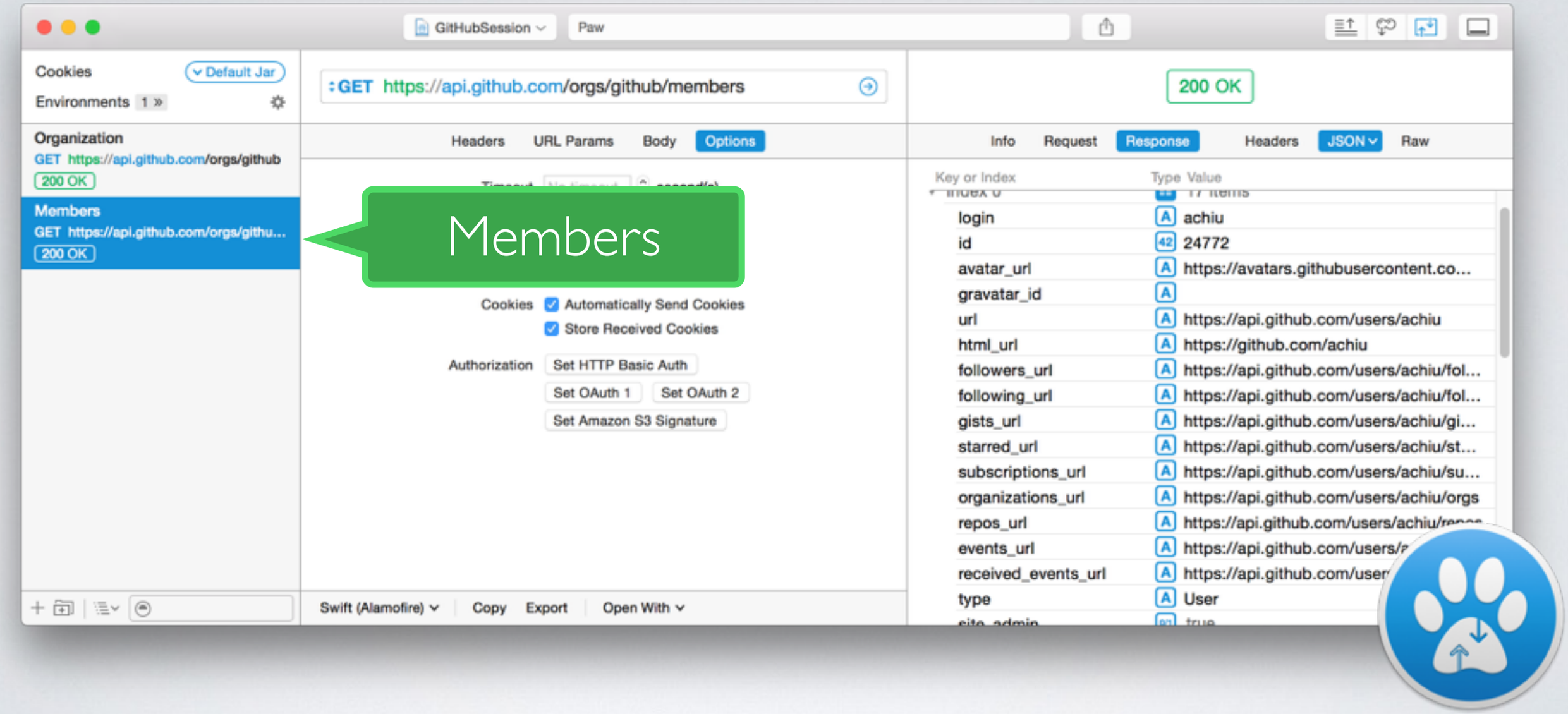
GITHUB API


GitHubSession
Paw
Cookies
Default Jar
Environments 1
GET https://api.github.com/orgs/github
Organization
GET https://api.github.com/orgs/github
200 OK
Members
GET https://api.github.com/orgs/githu...
200 OK
Add Header Name
Add Header Value
200 OK
Info
Request
Response
Headers
JSON
Raw
Key or Index
Type
Value
Root
22 items
login
github
id
9919
url
https://api.github.com/orgs/github
repos_url
https://api.github.com/orgs/github/repos
events_url
https://api.github.com/orgs/github/events
members_url
https://api.github.com/orgs/github/m...
public_members_url
https://api.github.com/orgs/github/pu...
avatar_url
https://avatars.githubusercontent.co...
description
GitHub, the company.
name
GitHub
company
null
blog
https://github.com/about
location
San Francisco, CA
email
support@github.com
public_repos
117
public_gists
0
Swift (Alamofire)
Copy
Export
Open With

GITHUB API


GitHubSession
Paw
Cookies
Default Jar
Environments 1
GET https://api.github.com/orgs/github/members
200 OK
Organization
GET https://api.github.com/orgs/github
200 OK
Members
GET https://api.github.com/orgs/githu...
200 OK
Headers
URL Params
Body
Options
Timeout
No timeout
second(s)
Redirections
Follow Redirects
Redirect Authorization
Redirect Original Method
Cookies
Automatically Send Cookies
Store Received Cookies
Authorization
Set HTTP Basic Auth
Set OAuth 1
Set OAuth 2
Set Amazon S3 Signature
Swift (Alamofire)
Copy
Export
Open With
200 OK
Info
Request
Response
Headers
JSON
Raw
Key or Index
Type
Value
login
achiu
id
24772
avatar_url
https://avatars.githubusercontent.co...
gravatar_id
url
https://api.github.com/users/achiu
html_url
https://github.com/achiu
followers_url
https://api.github.com/users/achiu/fol...
following_url
https://api.github.com/users/achiu/fol...
gists_url
https://api.github.com/users/achiu/gi...
starred_url
https://api.github.com/users/achiu/st...
subscriptions_url
https://api.github.com/users/achiu/su...
organizations_url
https://api.github.com/users/achiu/orgs
repos_url
events_url
received_events_url
type
User

GITHUB API


GitHubSession
Paw
Cookies
Default Jar
Environments 1
GET https://api.github.com/orgs/github/members
200 OK
Organization
GET https://api.github.com/orgs/github
200 OK
Members
GET https://api.github.com/orgs/githu...
200 OK
Members
Headers
URL Params
Body
Options
Cookies
Automatically Send Cookies
Store Received Cookies
Authorization
Set HTTP Basic Auth
Set OAuth 1
Set OAuth 2
Set Amazon S3 Signature
Swift (Alamofire)
Copy
Export
Open With
Info
Request
Response
Headers
JSON
Raw
Key or Index
Type
Value
login
achiu
id
24772
avatar_url
https://avatars.githubusercontent.co...
gravatar_id
url
https://api.github.com/users/achiu
html_url
https://github.com/achiu
followers_url
https://api.github.com/users/achiu/fol...
following_url
https://api.github.com/users/achiu/fol...
gists_url
https://api.github.com/users/achiu/gi...
starred_url
https://api.github.com/users/achiu/st...
subscriptions_url
https://api.github.com/users/achiu/su...
organizations_url
https://api.github.com/users/achiu/orgs
repos_url
events_url
received_events_url
type
User

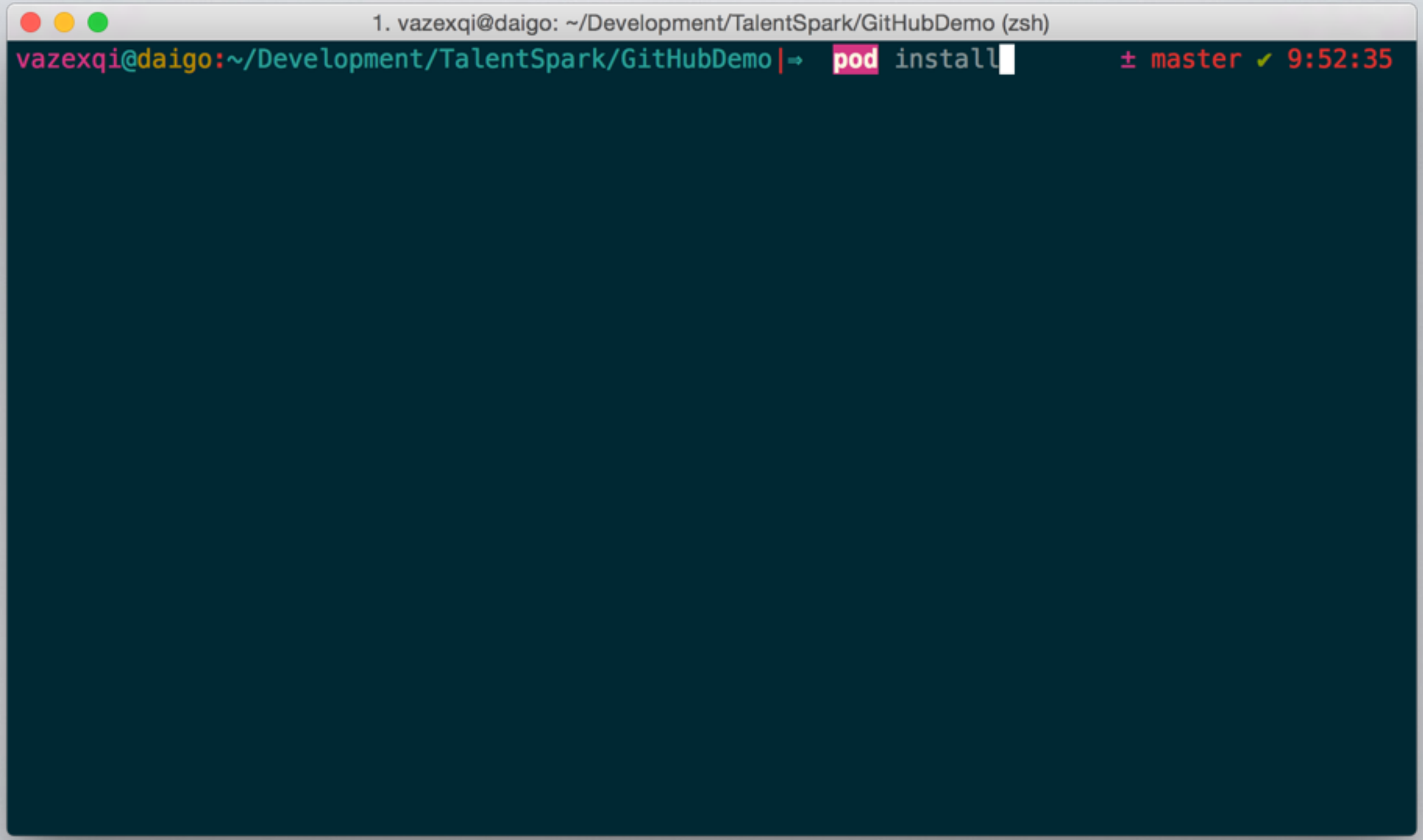
SWIFTYJSON

1. mvim -vf Podfile (Vim)

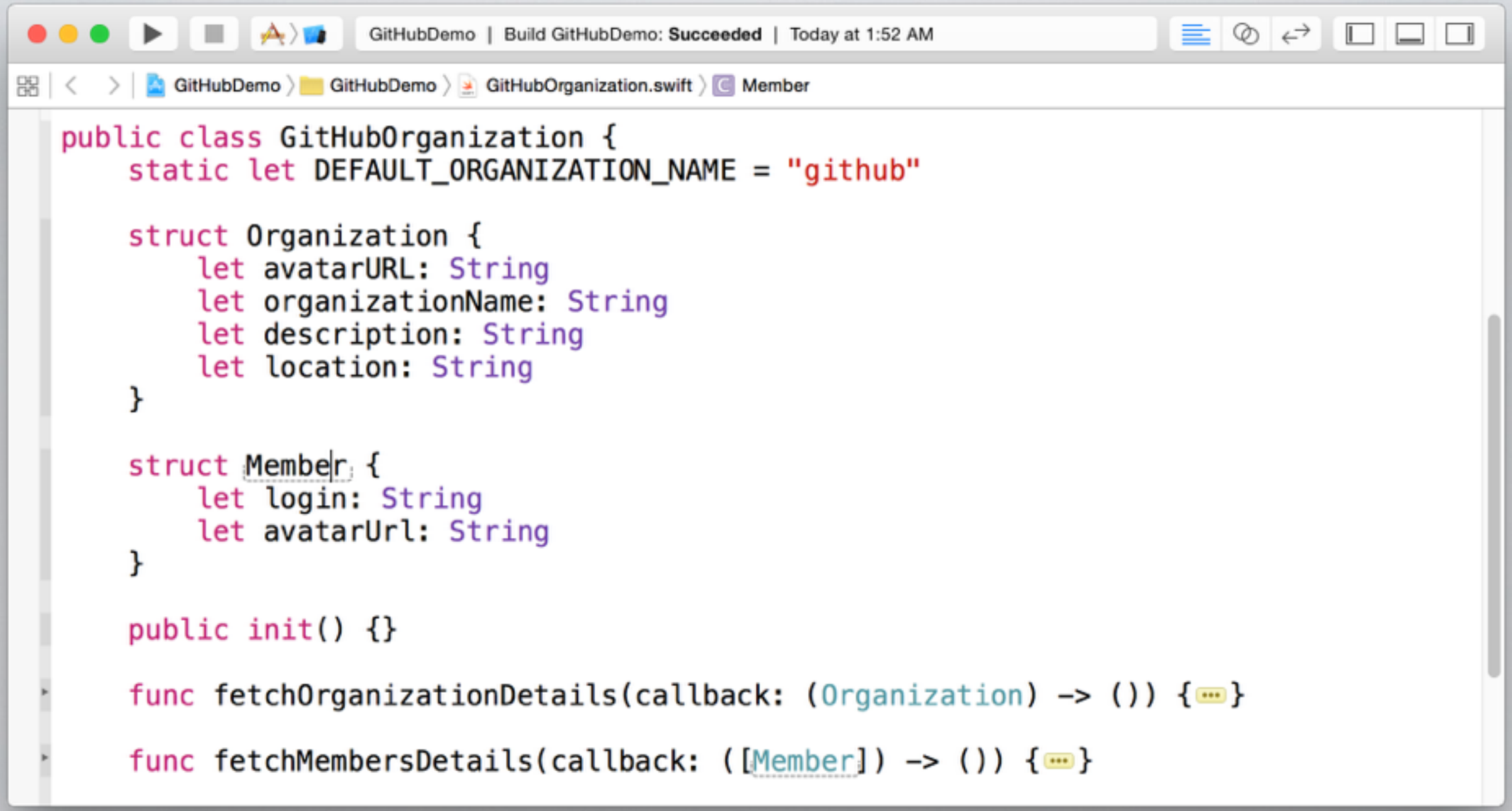
```ruby
pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git', :branch => 'swift-2.0'
pod 'SwiftyJSON', :git => 'https://github.com/SwiftyJSON/SwiftyJSON.git', :branch => 'xcode7'
use_frameworks!
```

NORMAL master Podfile unix utf-8 ruby 100% 3:15

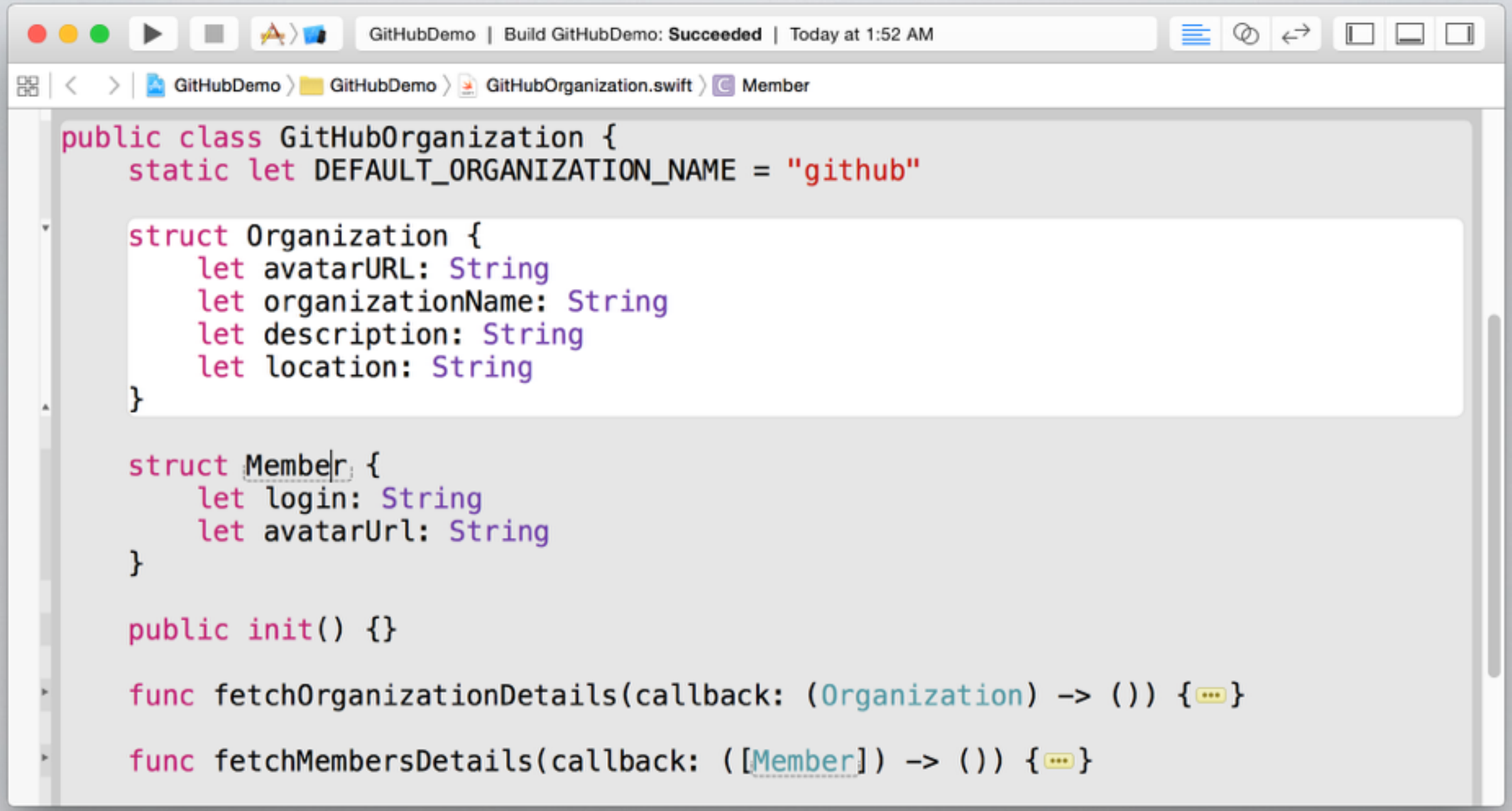
SWIFTYJSON


1. vazexqi@daigo: ~/Development/TalentSpark/GitHubDemo (zsh)
vazexqi@daigo:~/Development/TalentSpark/GitHubDemo|➜ pod install ± master ✔ 9:52:35

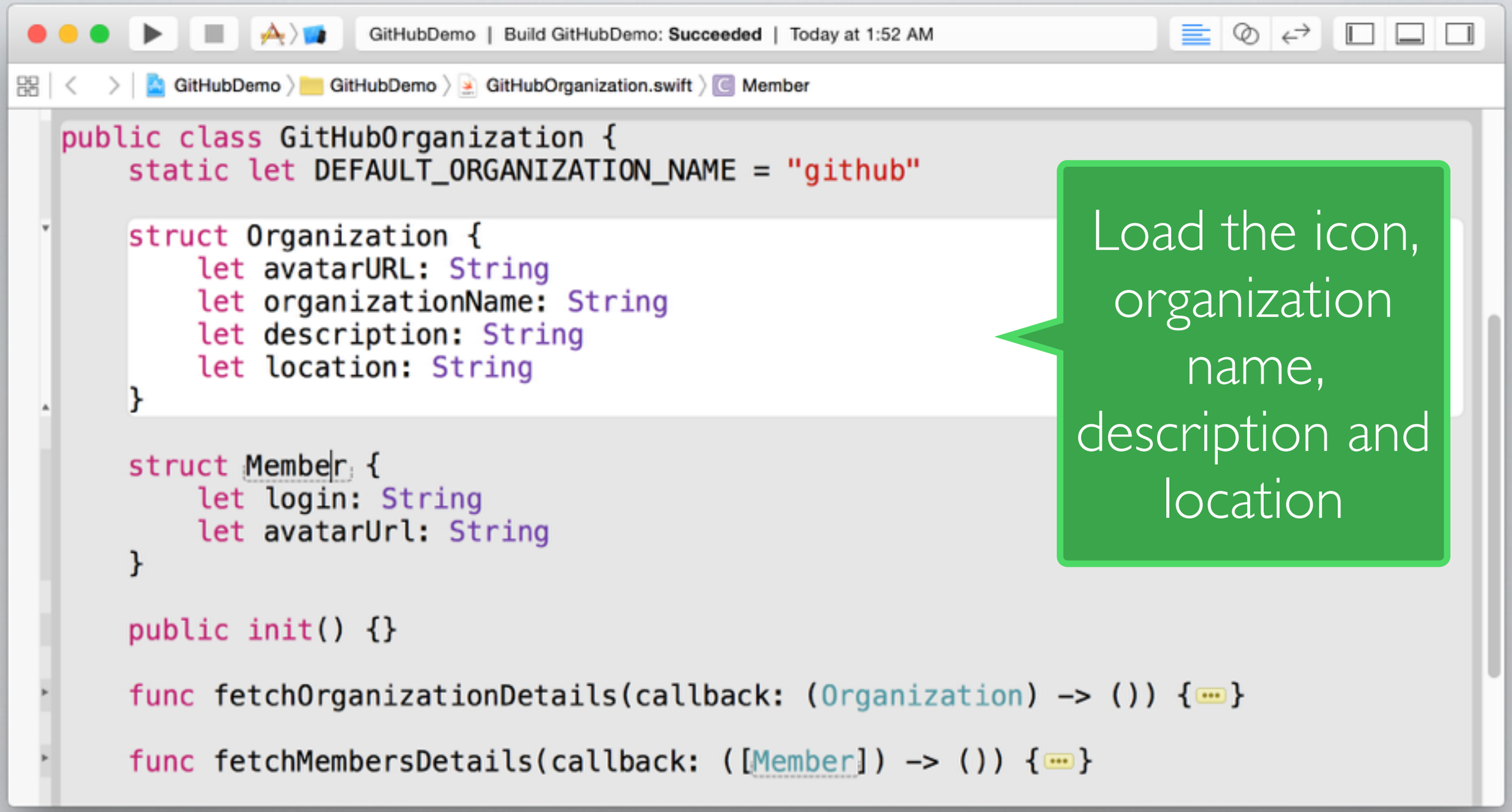
MODEL


GitHubDemo | Build GitHubDemo: Succeeded | Today at 1:52 AM
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member
public class GitHubOrganization {
static let DEFAULT_ORGANIZATION_NAME = "github"
struct Organization {
let avatarURL: String
let organizationName: String
let description: String
let location: String
}
struct Member {
let login: String
let avatarUrl: String
}
public init() {}
func fetchOrganizationDetails(callback: (Organization) -> ()) {...}
func fetchMembersDetails(callback: ([Member]) -> ()) {...}

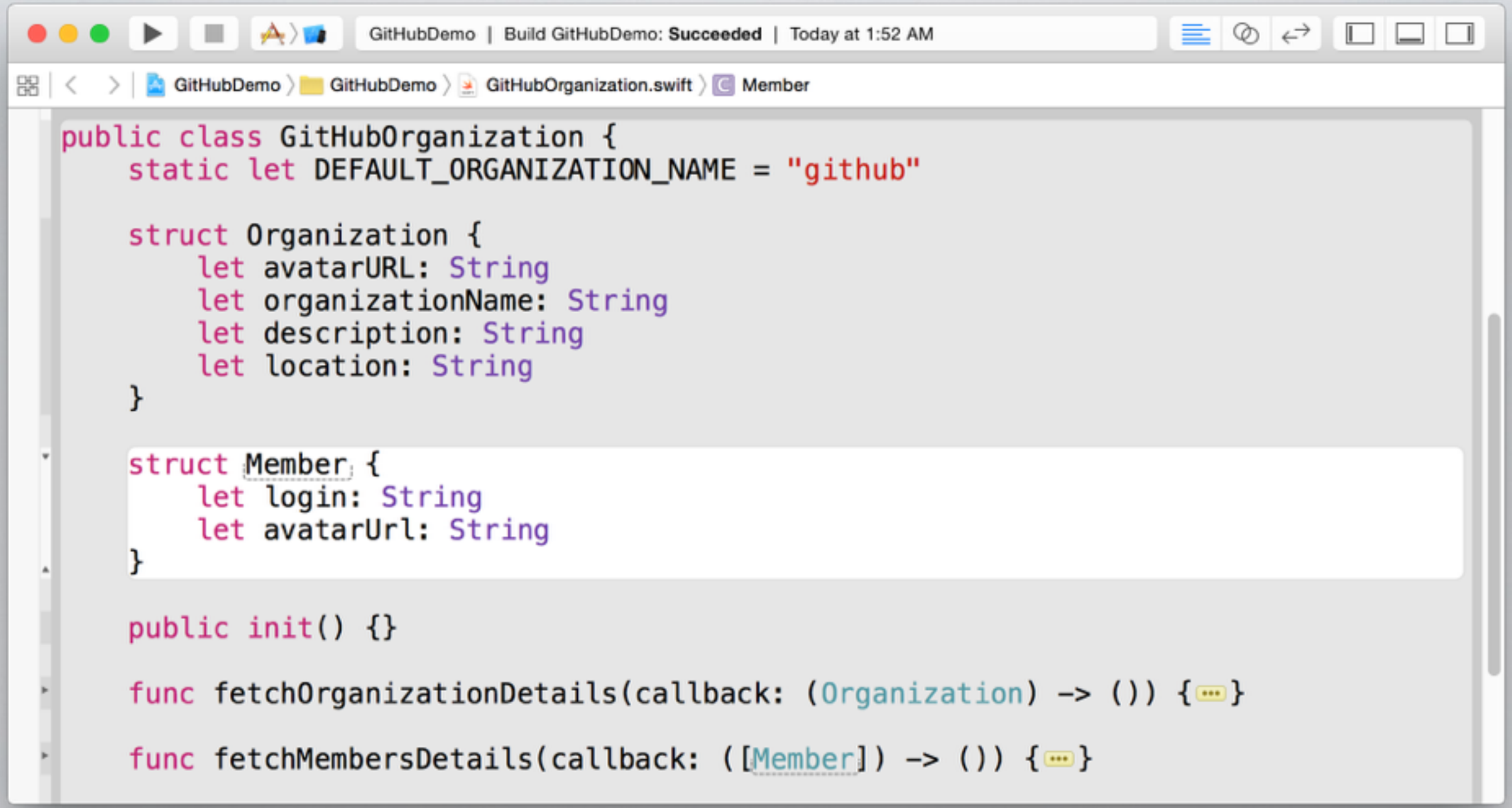
MODEL


GitHubDemo | Build GitHubDemo: Succeeded | Today at 1:52 AM
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member
public class GitHubOrganization {
static let DEFAULT_ORGANIZATION_NAME = "github"
struct Organization {
let avatarURL: String
let organizationName: String
let description: String
let location: String
}
struct Member {
let login: String
let avatarUrl: String
}
public init() {}
func fetchOrganizationDetails(callback: (Organization) -> ()) {...}
func fetchMembersDetails(callback: ([Member]) -> ()) {...}

MODEL


GitHubDemo | Build GitHubDemo: Succeeded | Today at 1:52 AM
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member
public class GitHubOrganization {
static let DEFAULT_ORGANIZATION_NAME = "github"
struct Organization {
let avatarURL: String
let organizationName: String
let description: String
let location: String
}
struct Member {
let login: String
let avatarUrl: String
}
public init() {}
func fetchOrganizationDetails(callback: (Organization) -> ()) {...}
func fetchMembersDetails(callback: ([Member]) -> ()) {...}
Load the icon, organization name, description and location

MODEL


GitHubDemo | Build GitHubDemo: Succeeded | Today at 1:52 AM
GitHubDemo
GitHubDemo
GitHubOrganization.swift
Member
public class GitHubOrganization {
static let DEFAULT_ORGANIZATION_NAME = "github"
struct Organization {
let avatarURL: String
let organizationName: String
let description: String
let location: String
}
struct Member {
let login: String
let avatarUrl: String
}
public init() {}
func fetchOrganizationDetails(callback: (Organization) -> ()) {...}
func fetchMembersDetails(callback: ([Member]) -> ()) {...}

MODEL


GitHubDemo | Build GitHubDemo: Succeeded | Today at 1:52 AM
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member
public class GitHubOrganization {
static let DEFAULT_ORGANIZATION_NAME = "github"
struct Organization {
let avatarURL: String
let organizationName: String
let description: String
let location: String
}
struct Member {
let login: String
let avatarUrl: String
}
public init() {}
func fetchOrganizationDetails(callback: (Organization) -> ()) {...}
func fetchMembersDetails(callback: ([Member]) -> ()) {...}
Just need the icon and username

MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Void) {
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!, callback:
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].stringValue, description:
                json["description"].stringValue, location:
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Vo
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].stringValue, description:
                json["description"].stringValue, location:
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

We've seen how to use Alamofire before

MODEL

GitHubOrganization.swift

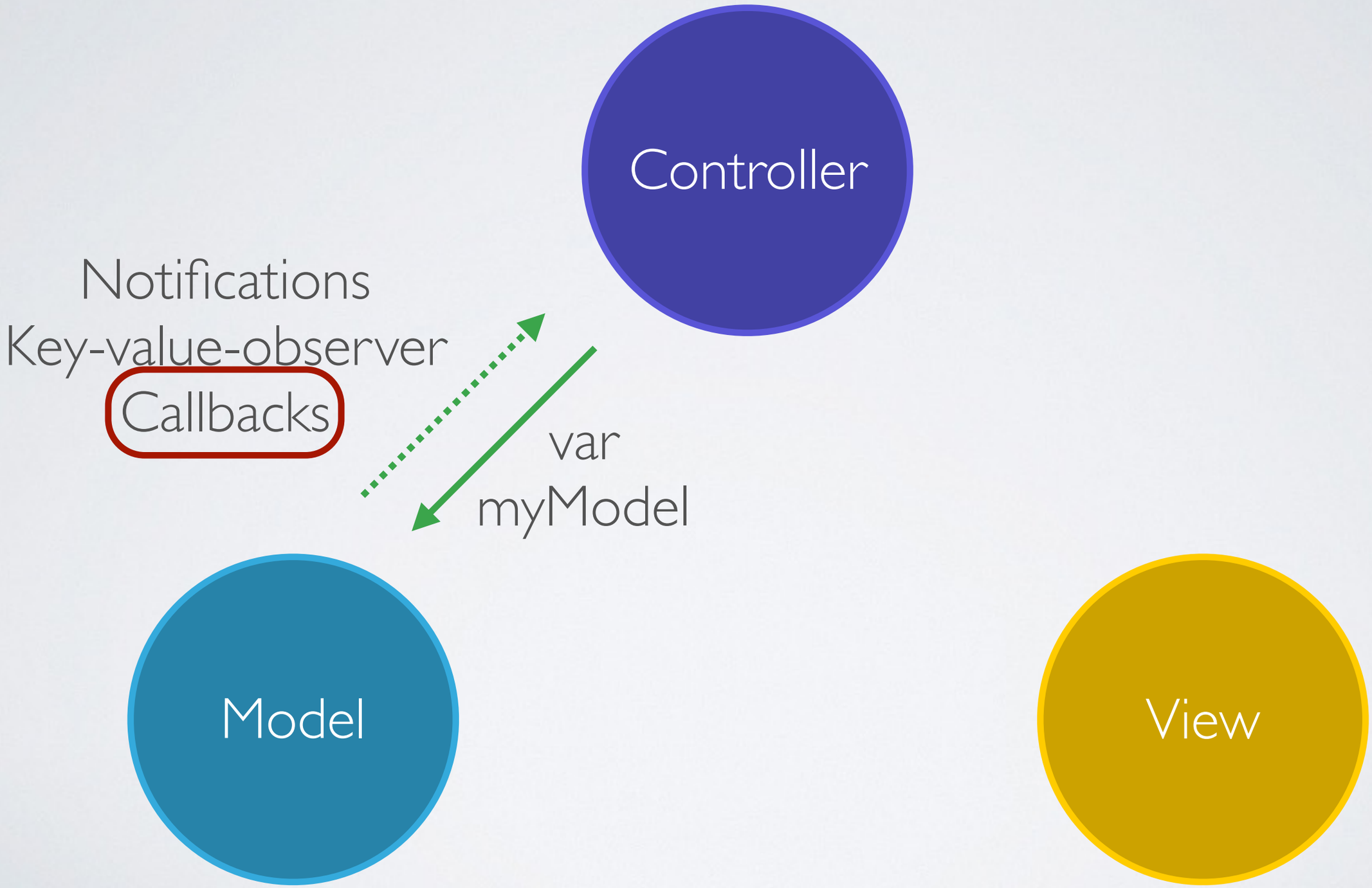
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Vo
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                                  son["name"].stringValue, description:
                                  .stringValue, location:
                                  ringValue))
39      }
40
41      func                      llback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

We've seen how to use Alamofire before

This is SwiftyJSON

NON-SPAGHETTI CODE



MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Void) {
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!, callback:
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].stringValue, description:
                json["description"].stringValue, location:
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Void) {
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!, callback:
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].str
                json["description"].stringValue, l
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Membe
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.g
                GitHubOrganization.DEFAULT_ORGANIZ                    )
```

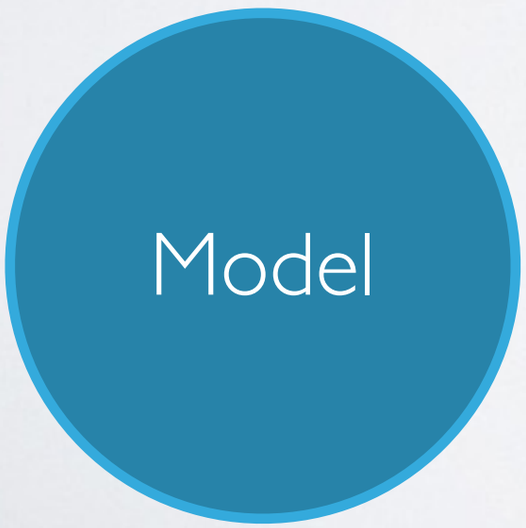
How we communicate with our controller

CONTROLLER

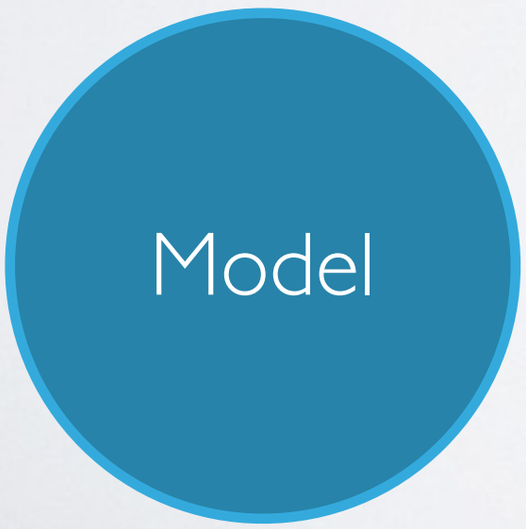
Model

View

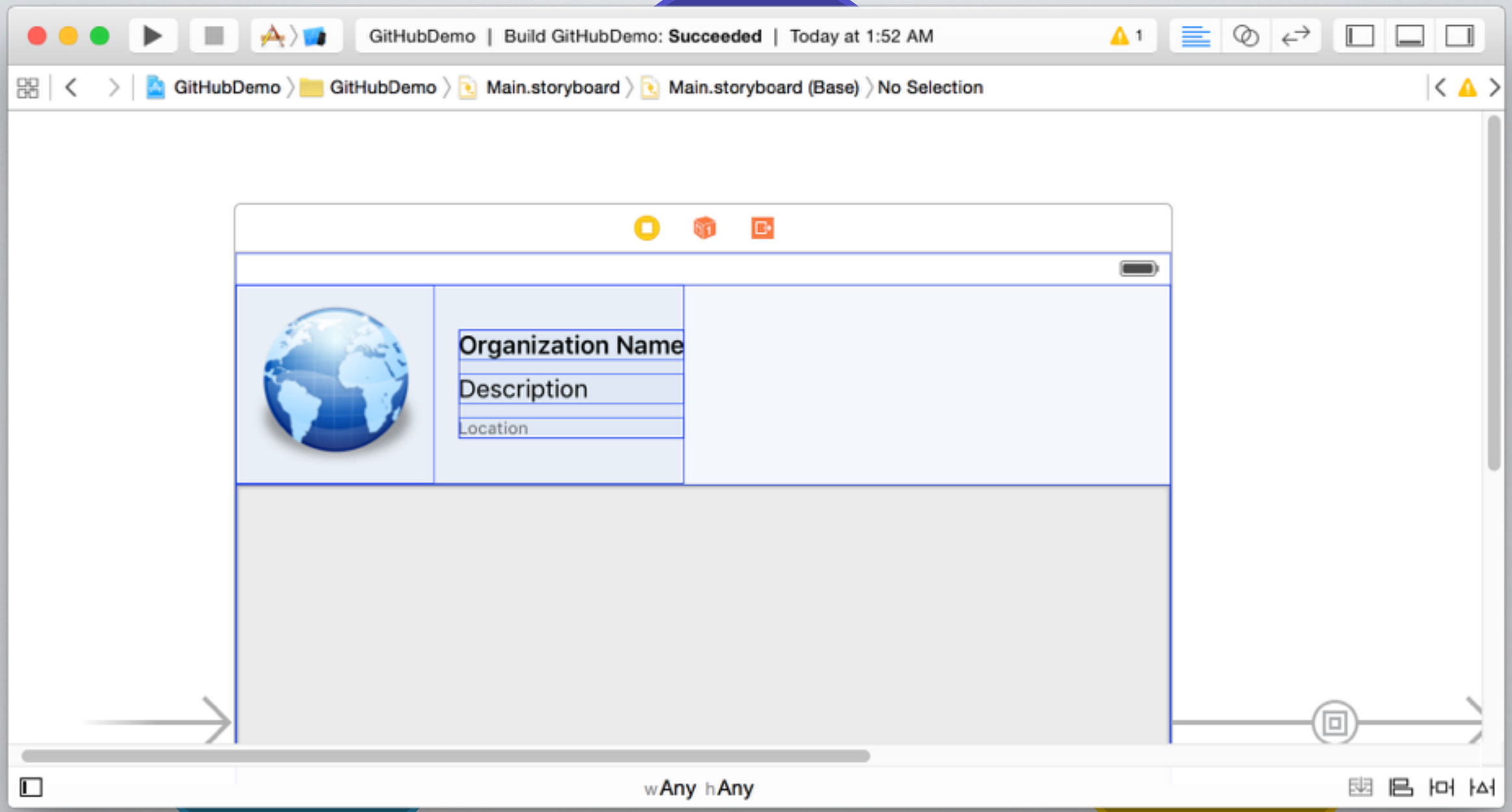
CONTROLLER



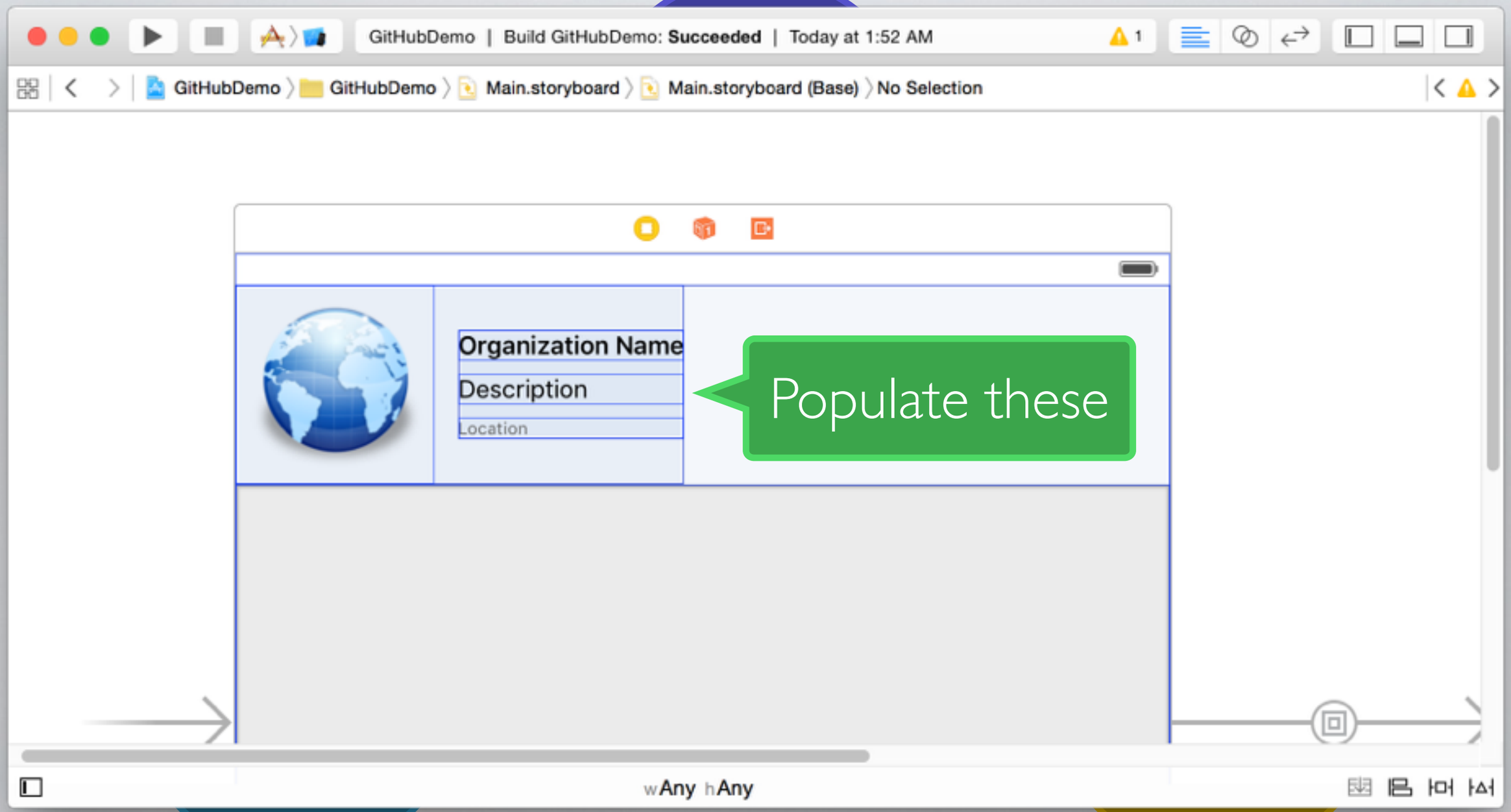
CONTROLLER # I



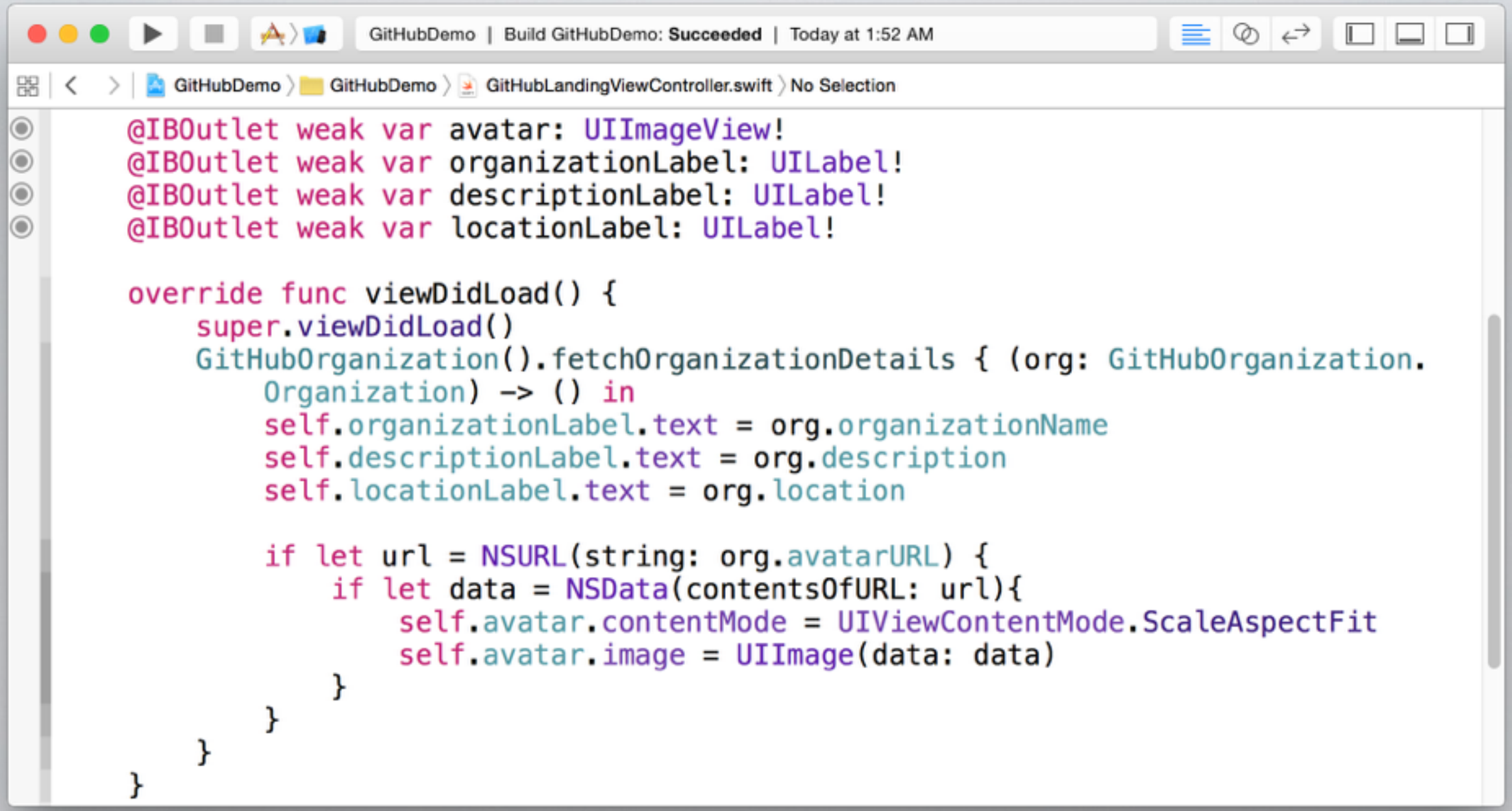
CONTROLLER # I



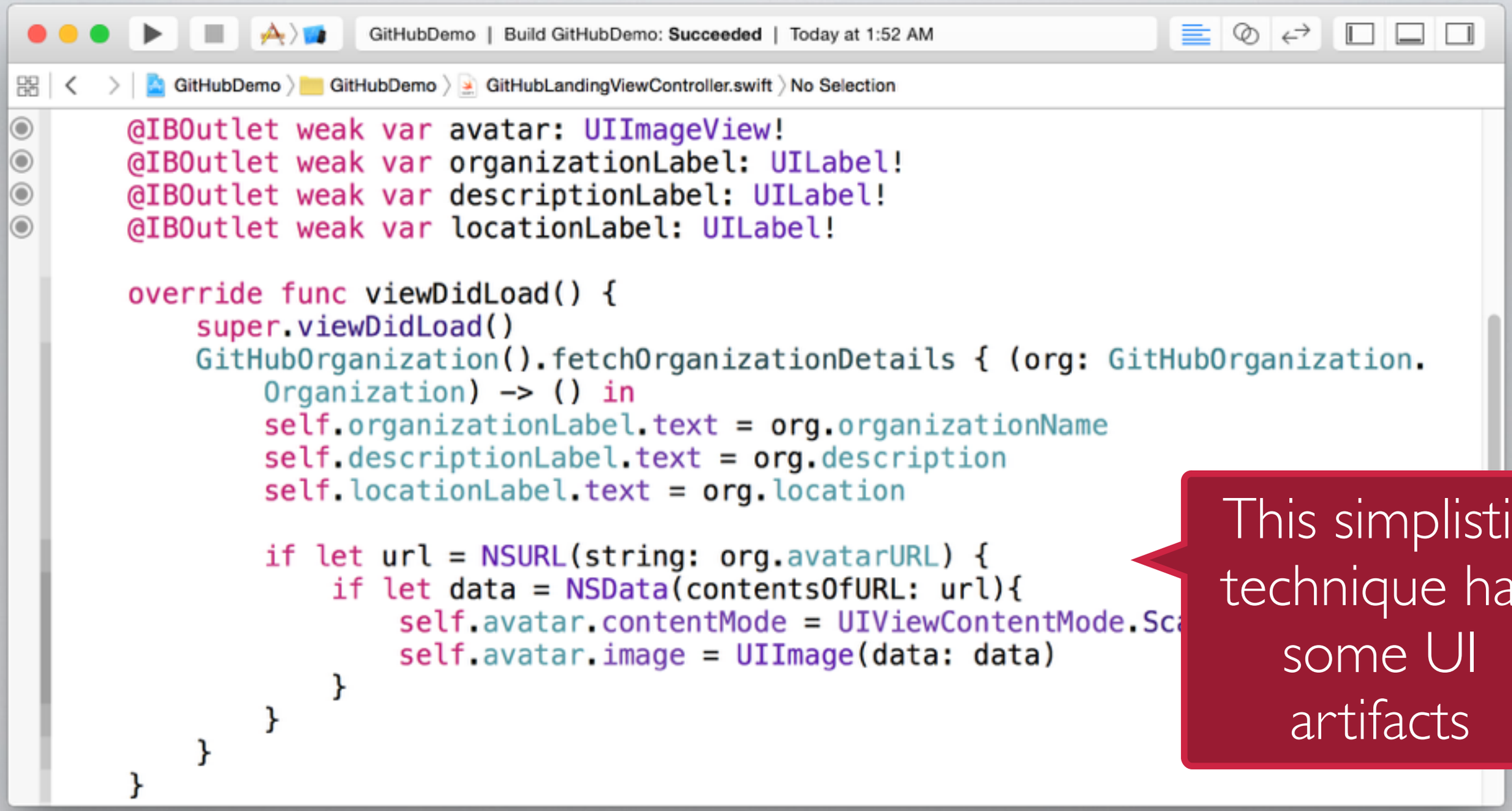
CONTROLLER # I



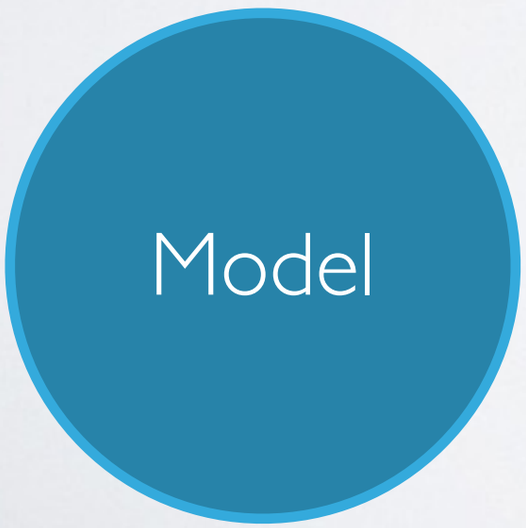
CONTROLLER # I



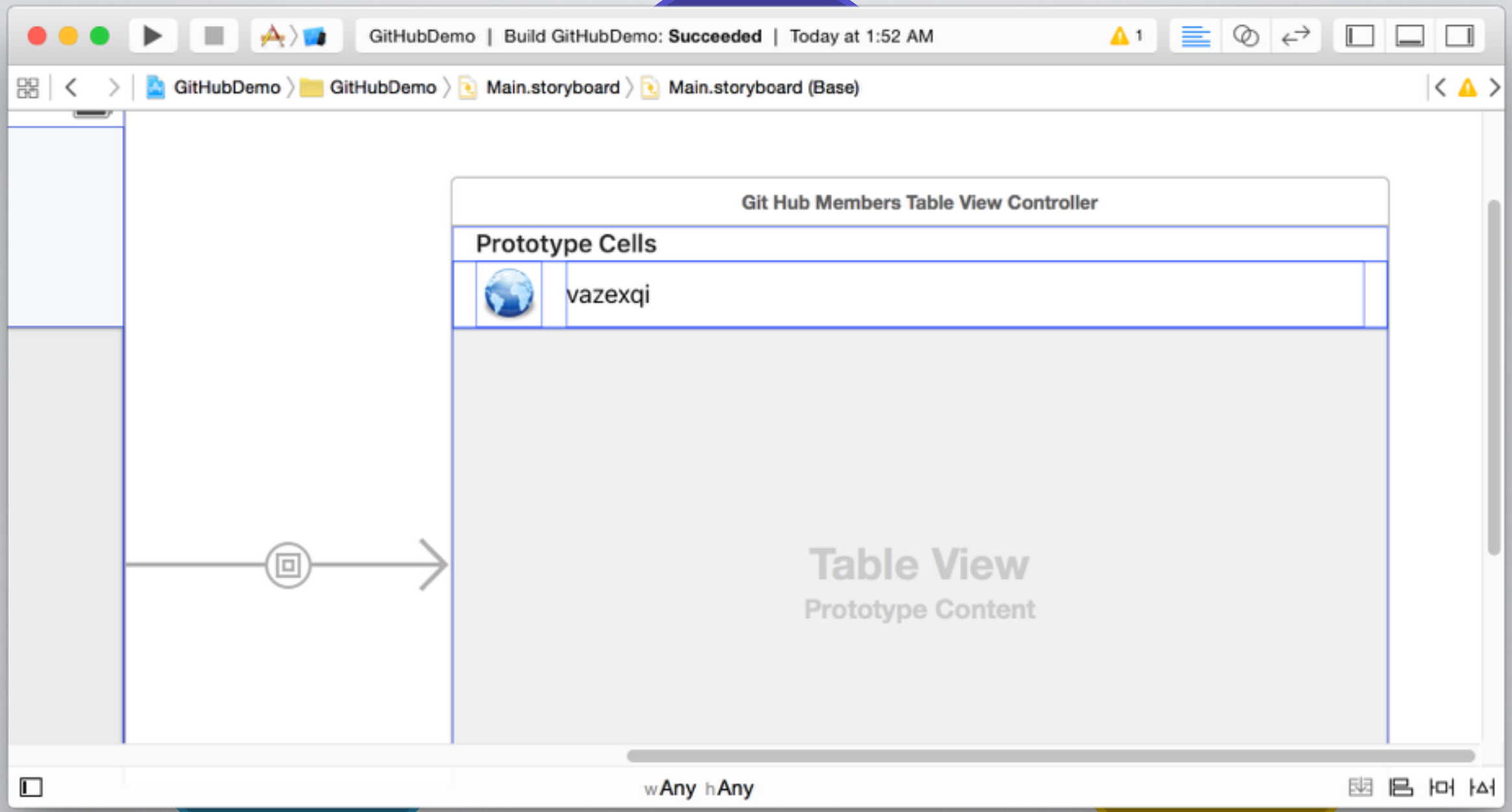
CONTROLLER # I



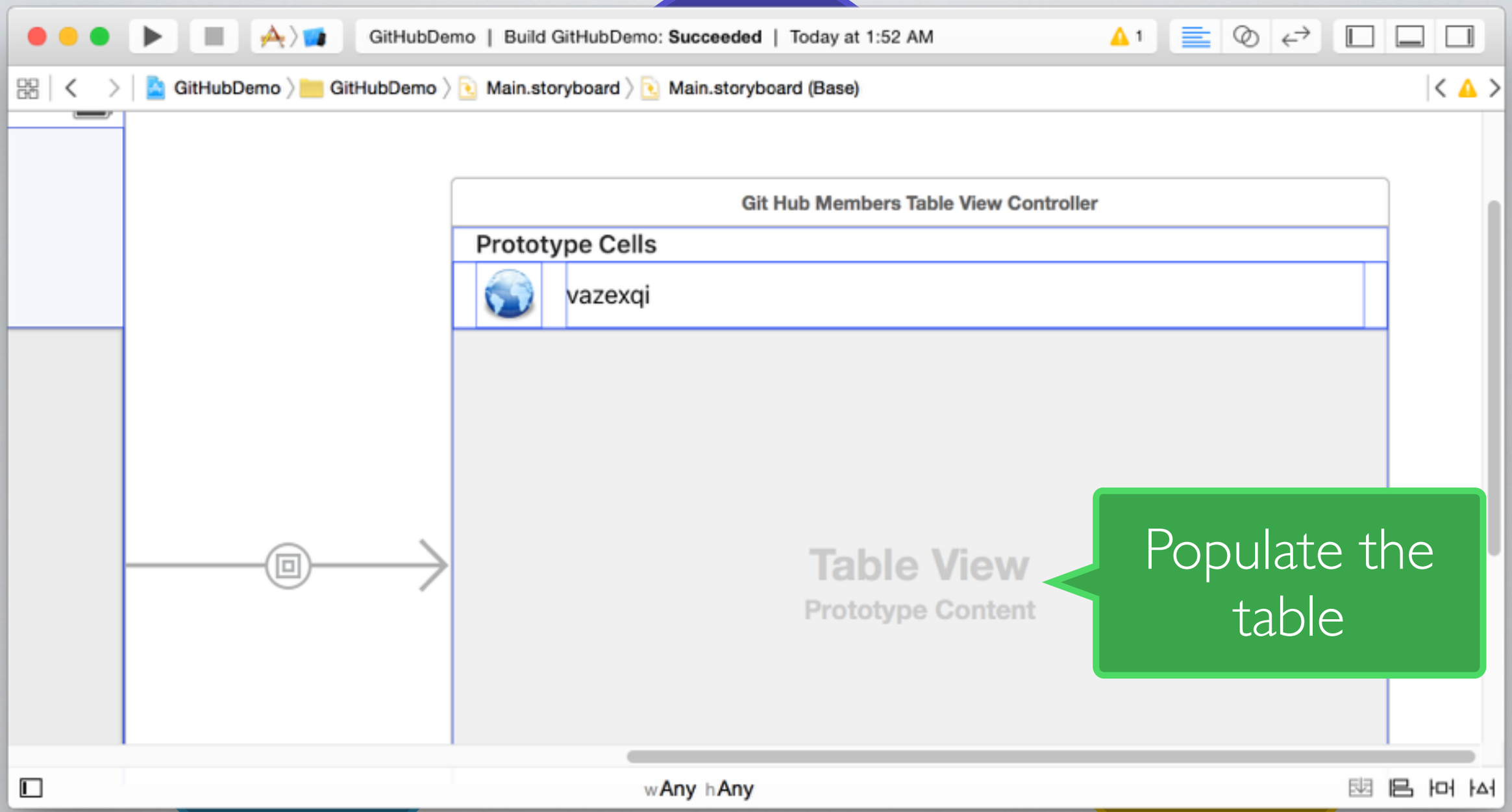
CONTROLLER #2



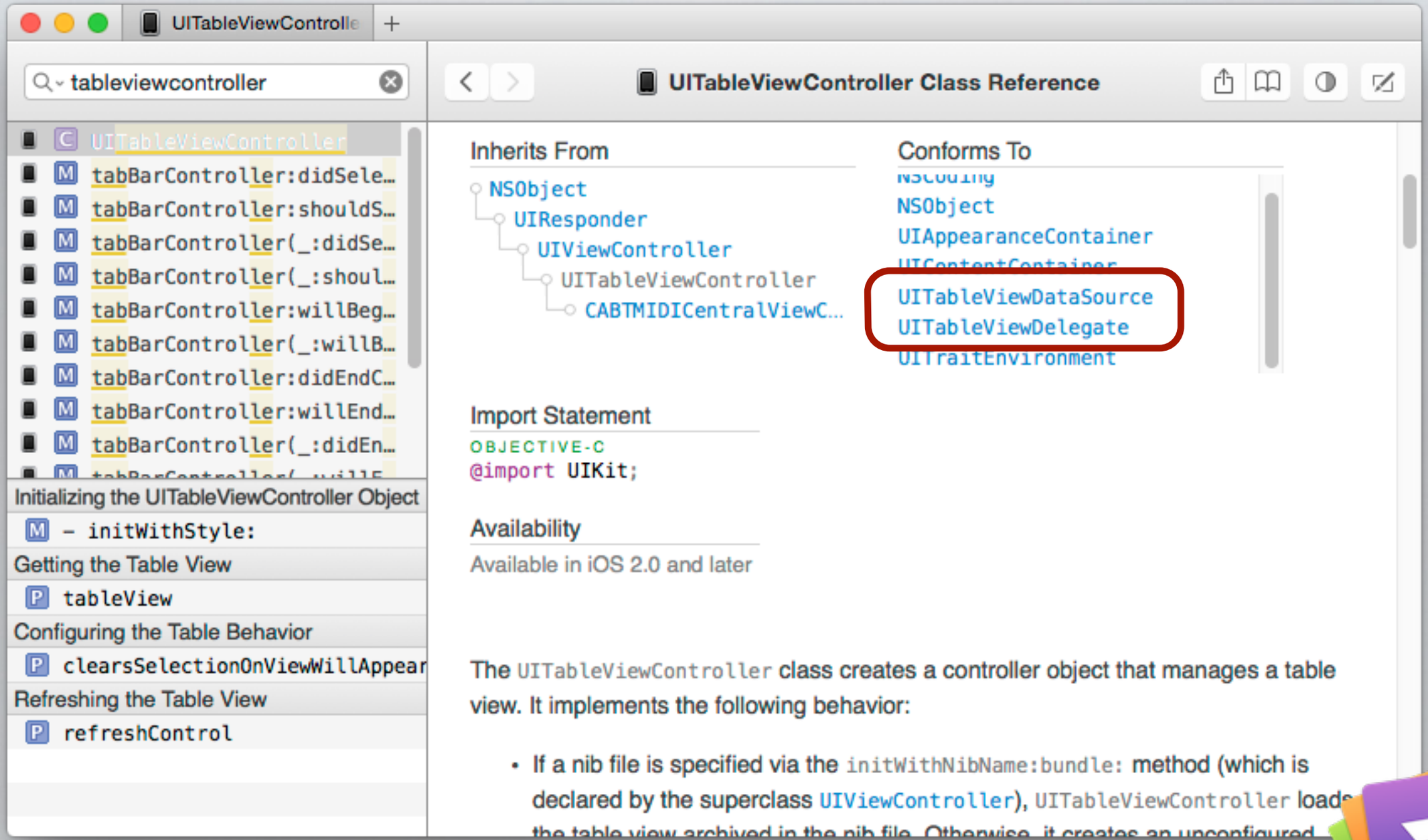
CONTROLLER #2



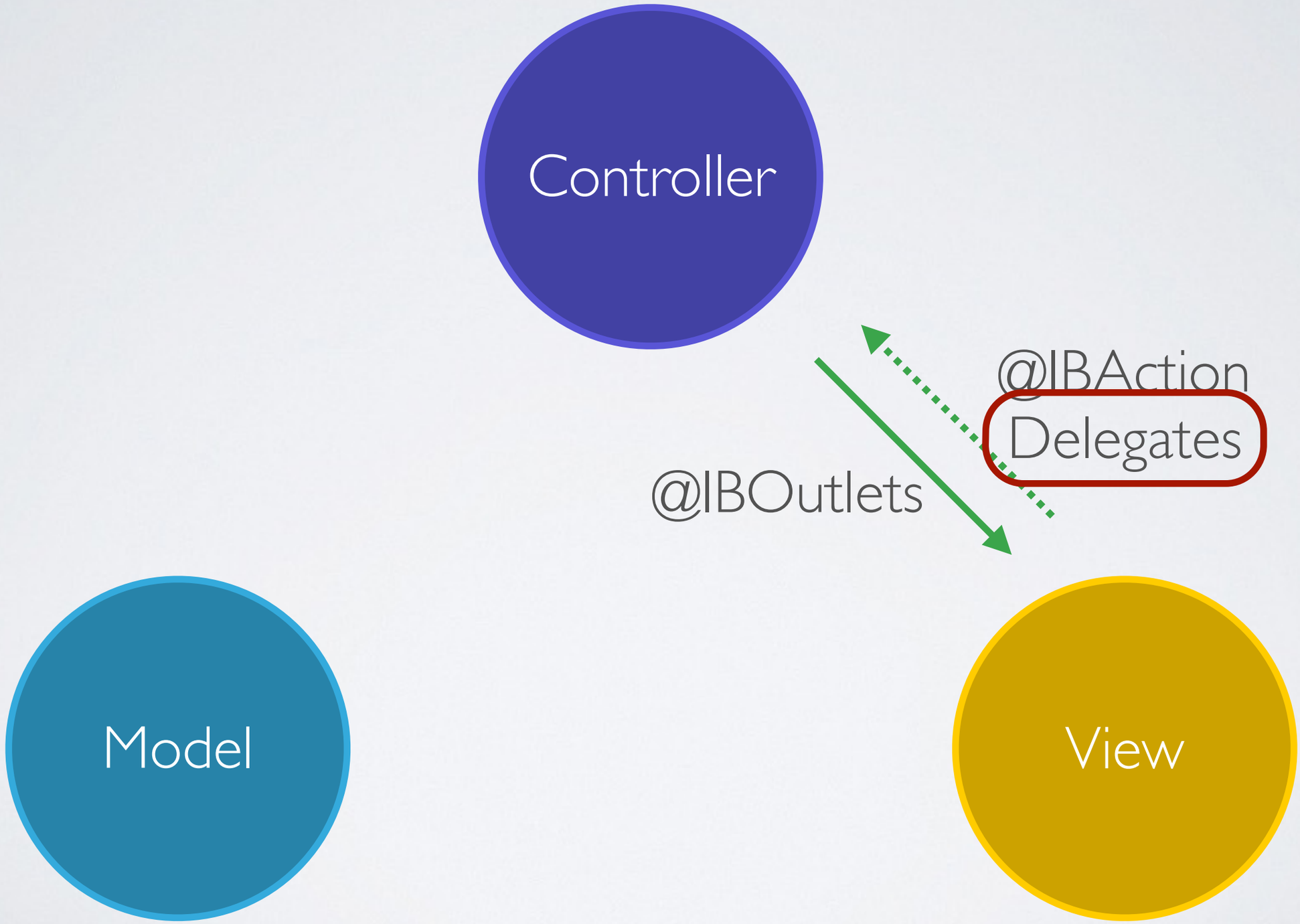
CONTROLLER #2



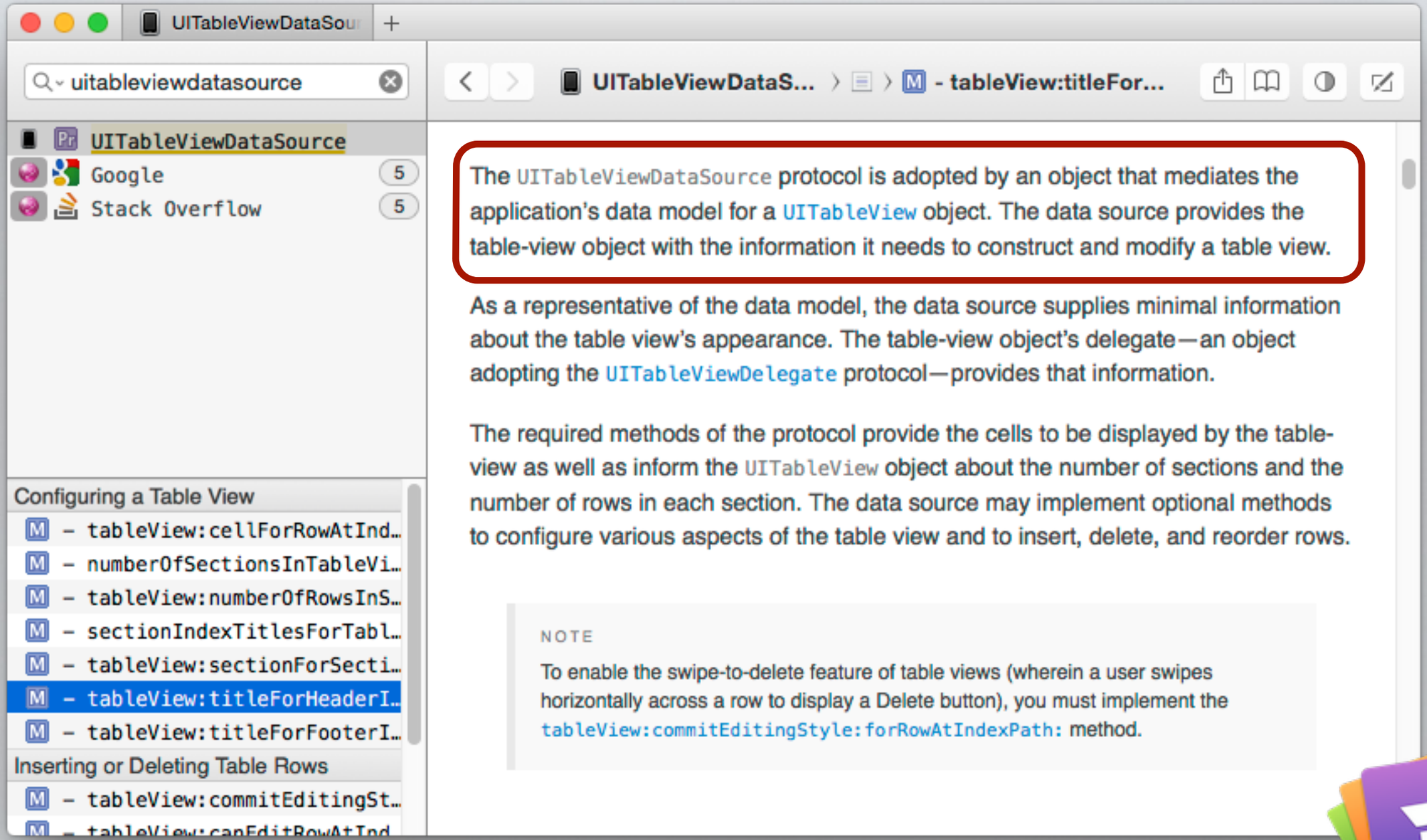
TABLEVIEWCONTROLLER



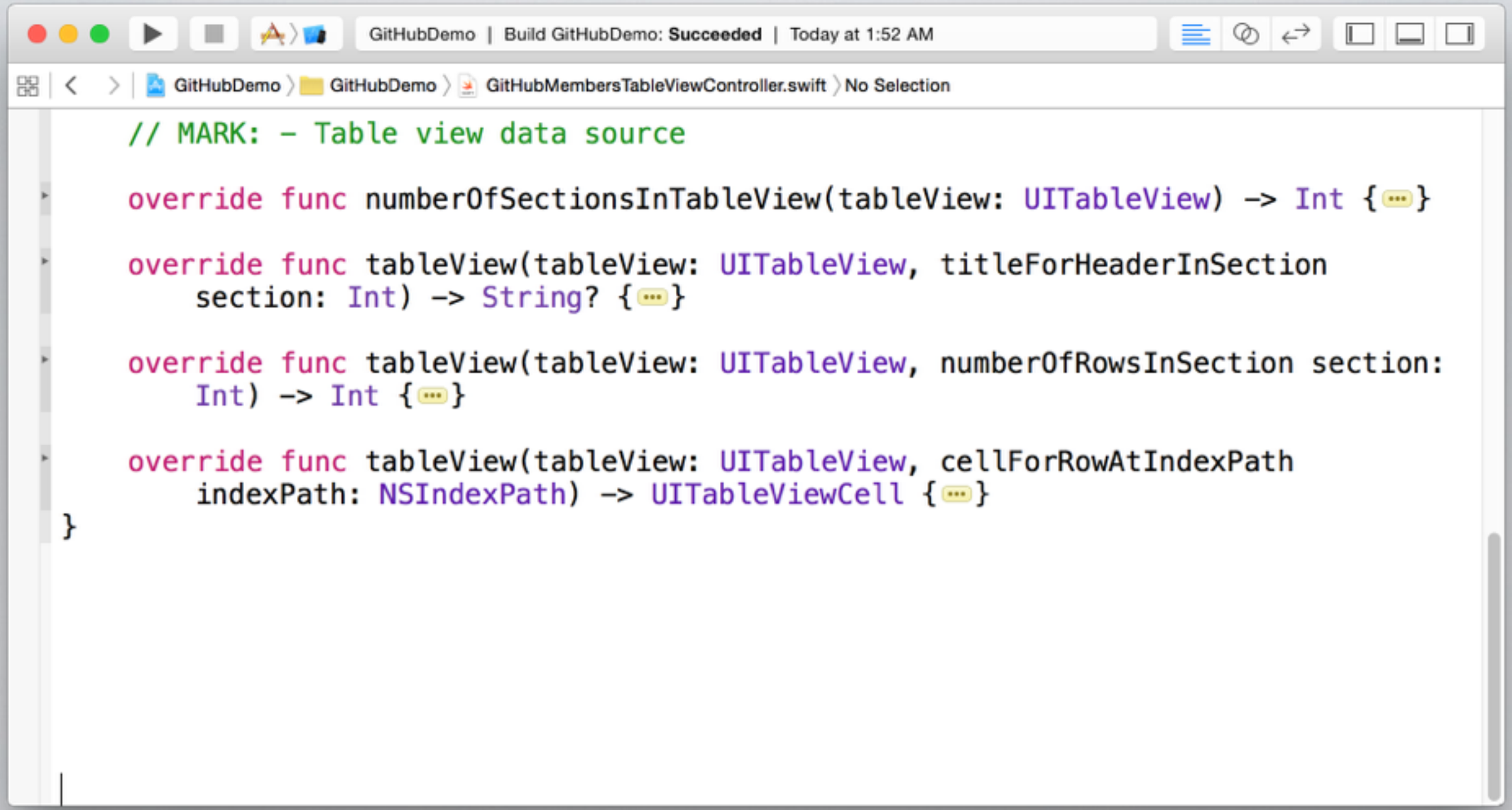
NON-SPAGHETTI CODE



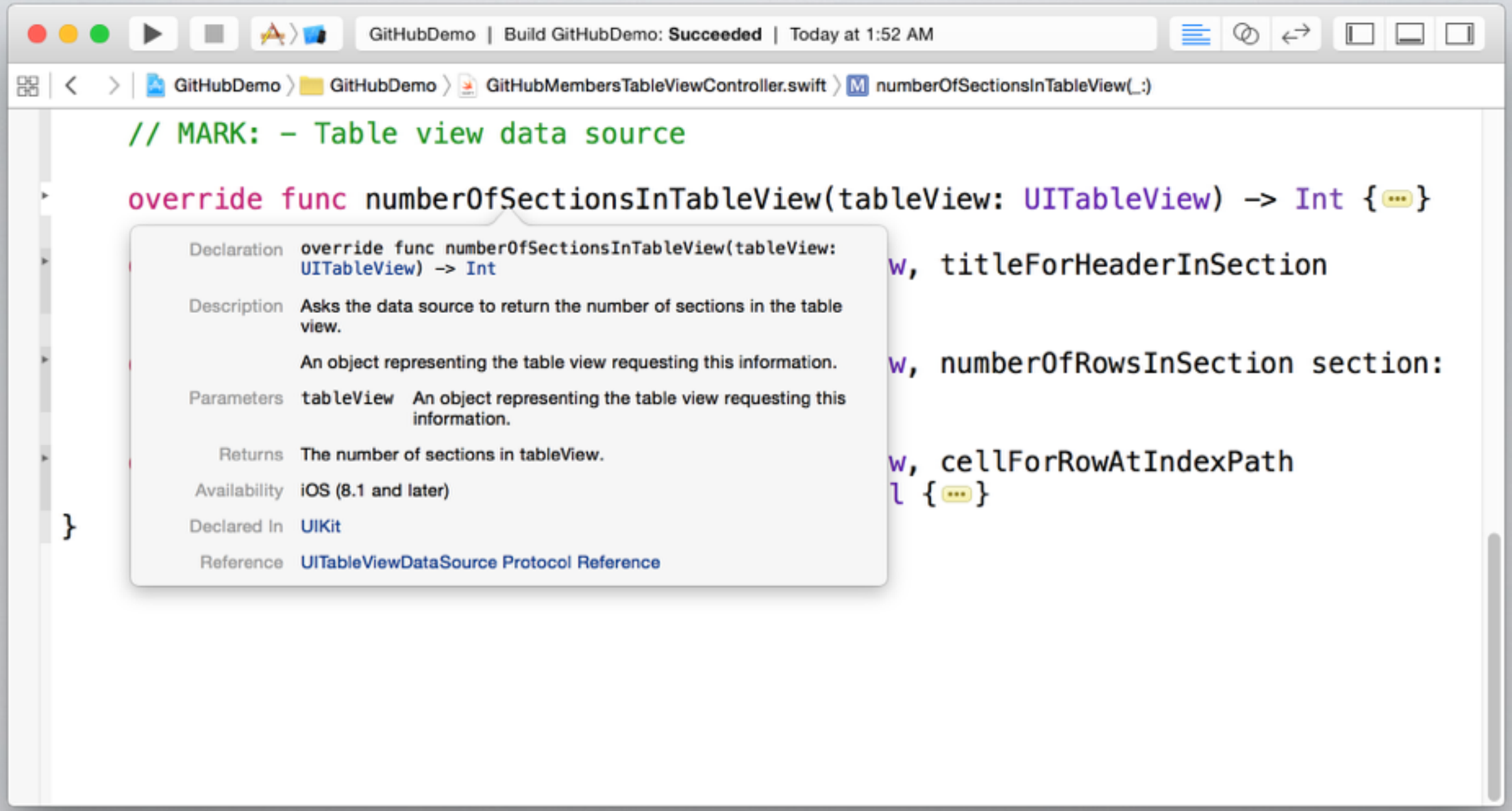
UITABLEVIEWDATASOURCE



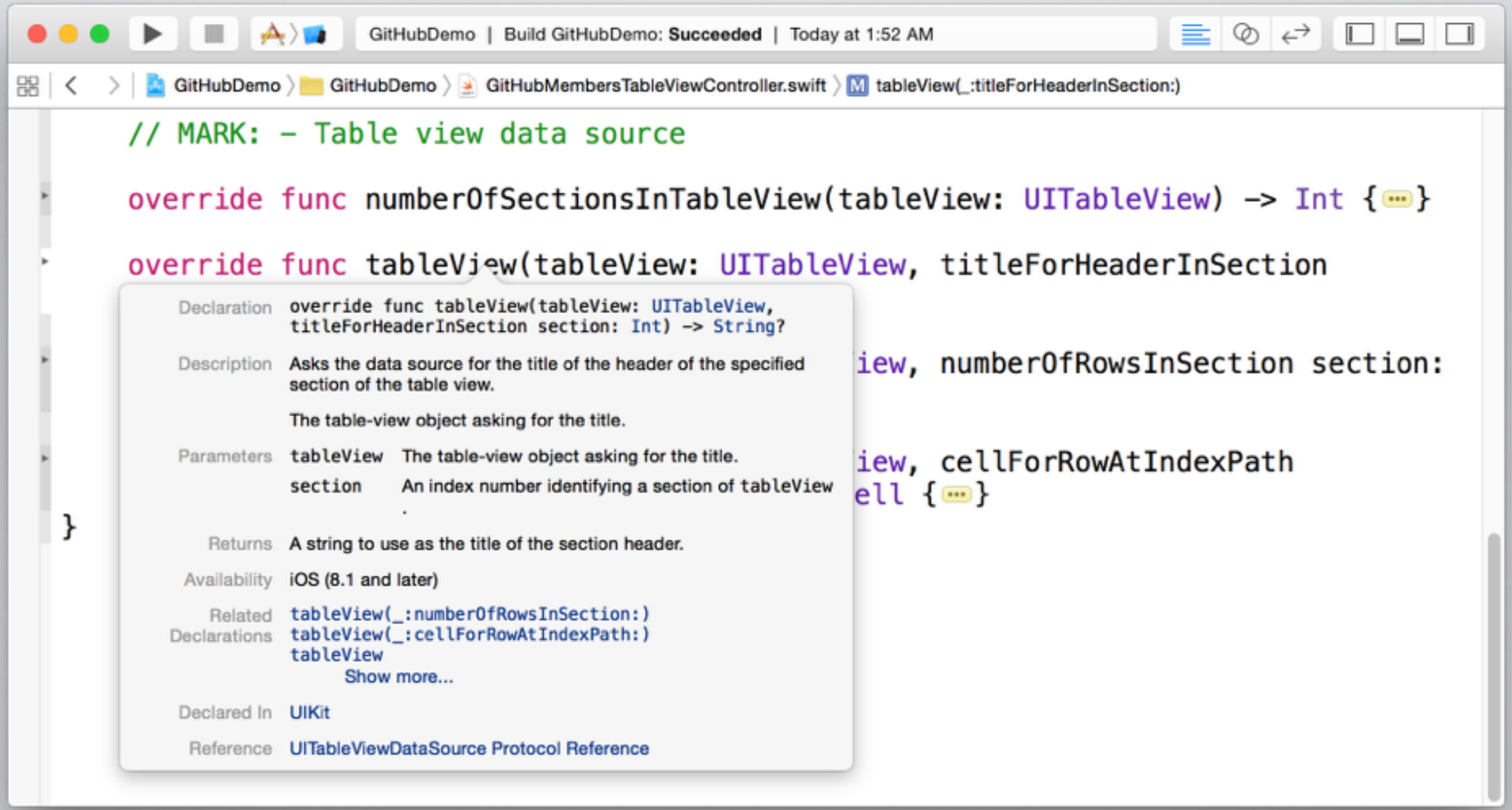
UITABLEVIEWDATASOURCE



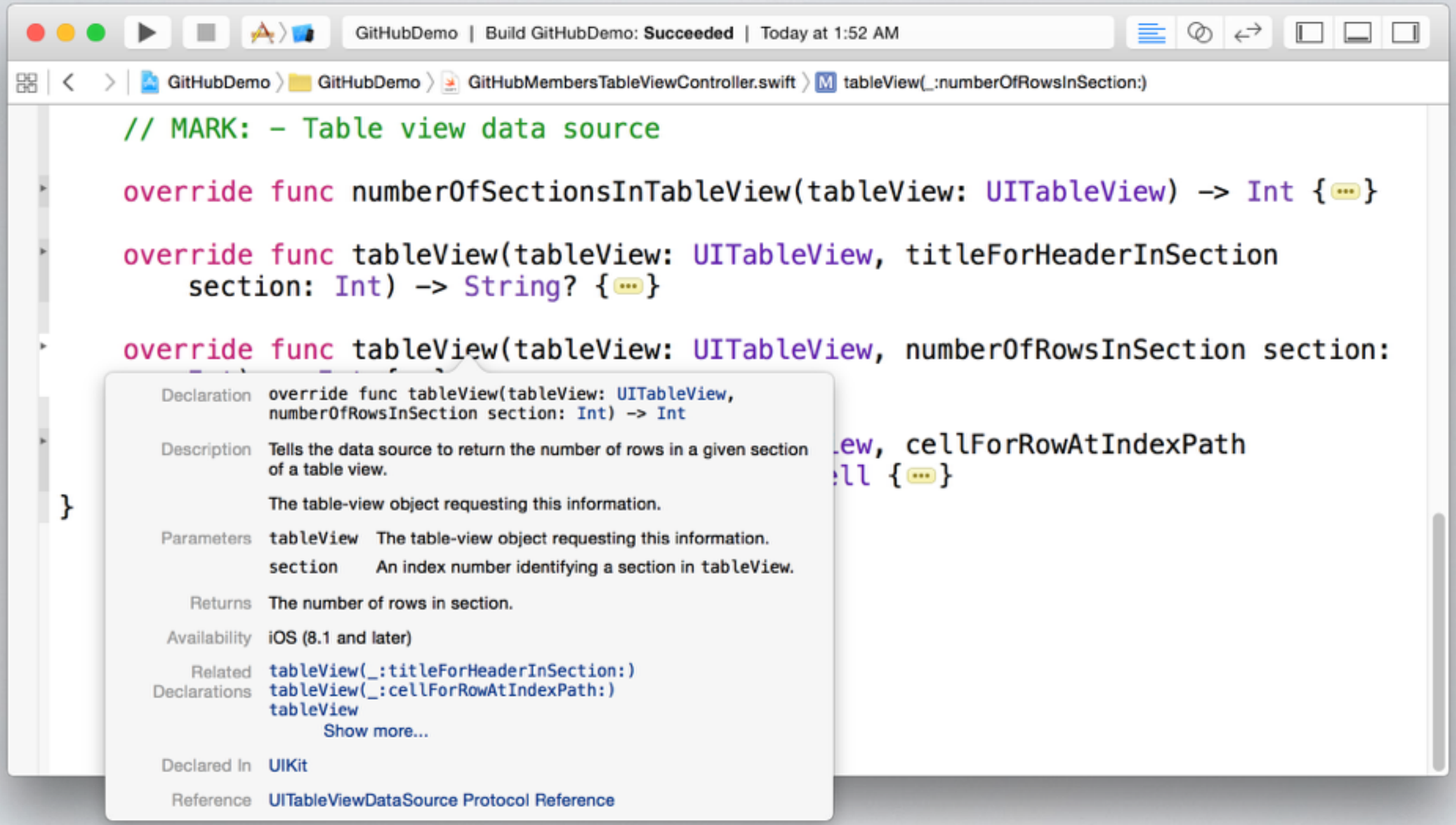
UITABLEVIEWDATASOURCE



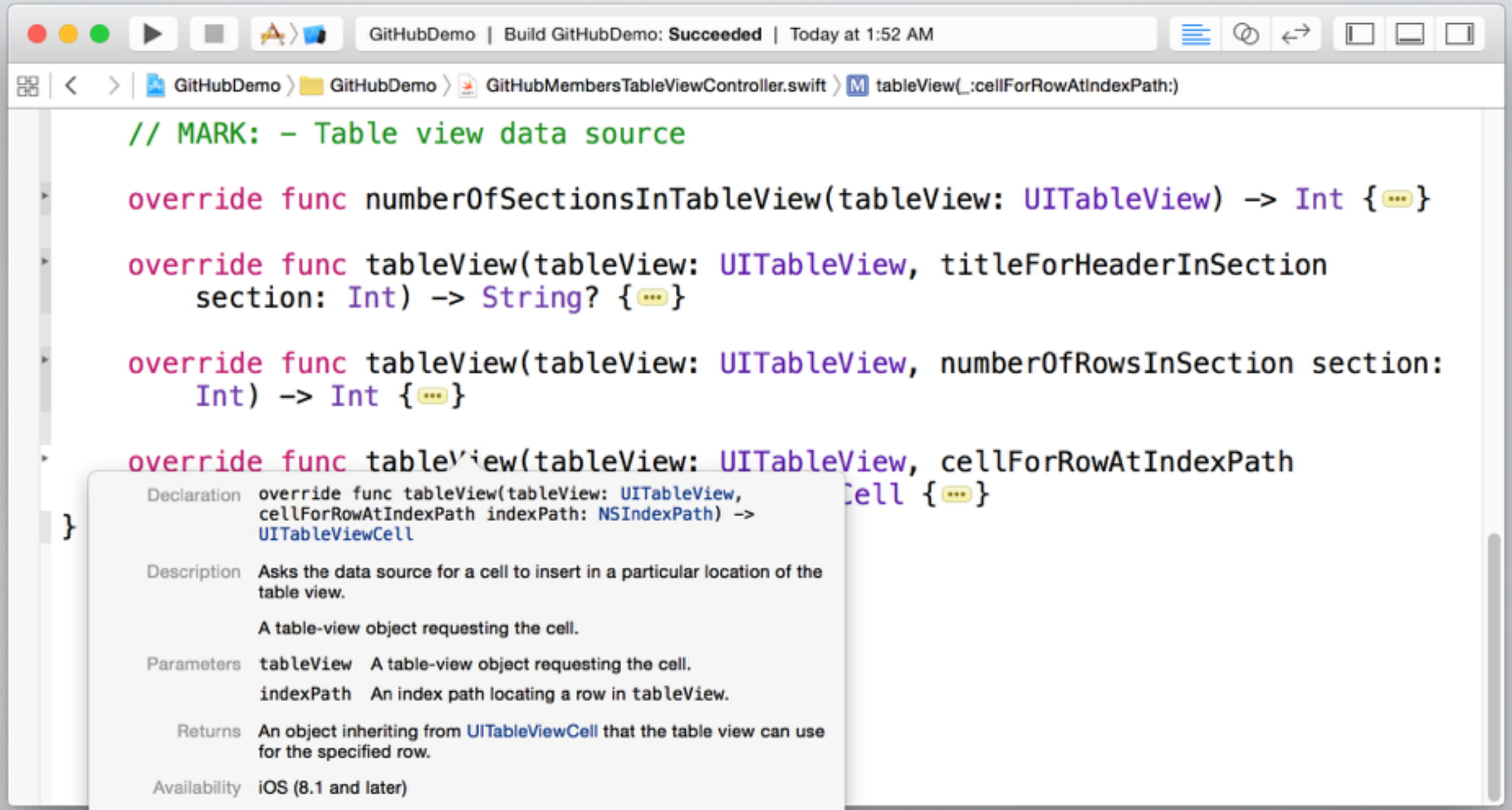
UITABLEVIEWDATASOURCE



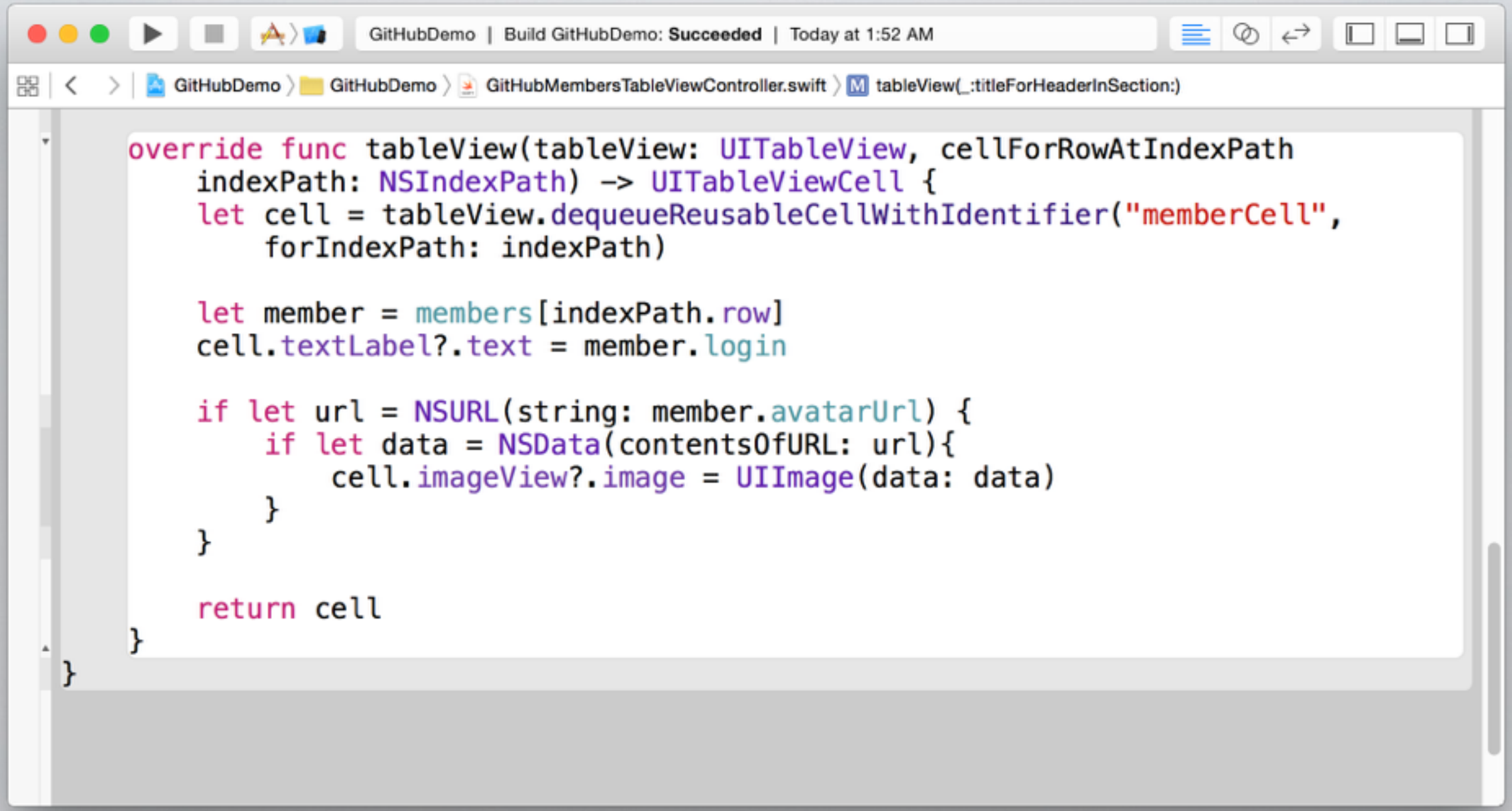
UITABLEVIEWDATASOURCE



UITABLEVIEWDATASOURCE

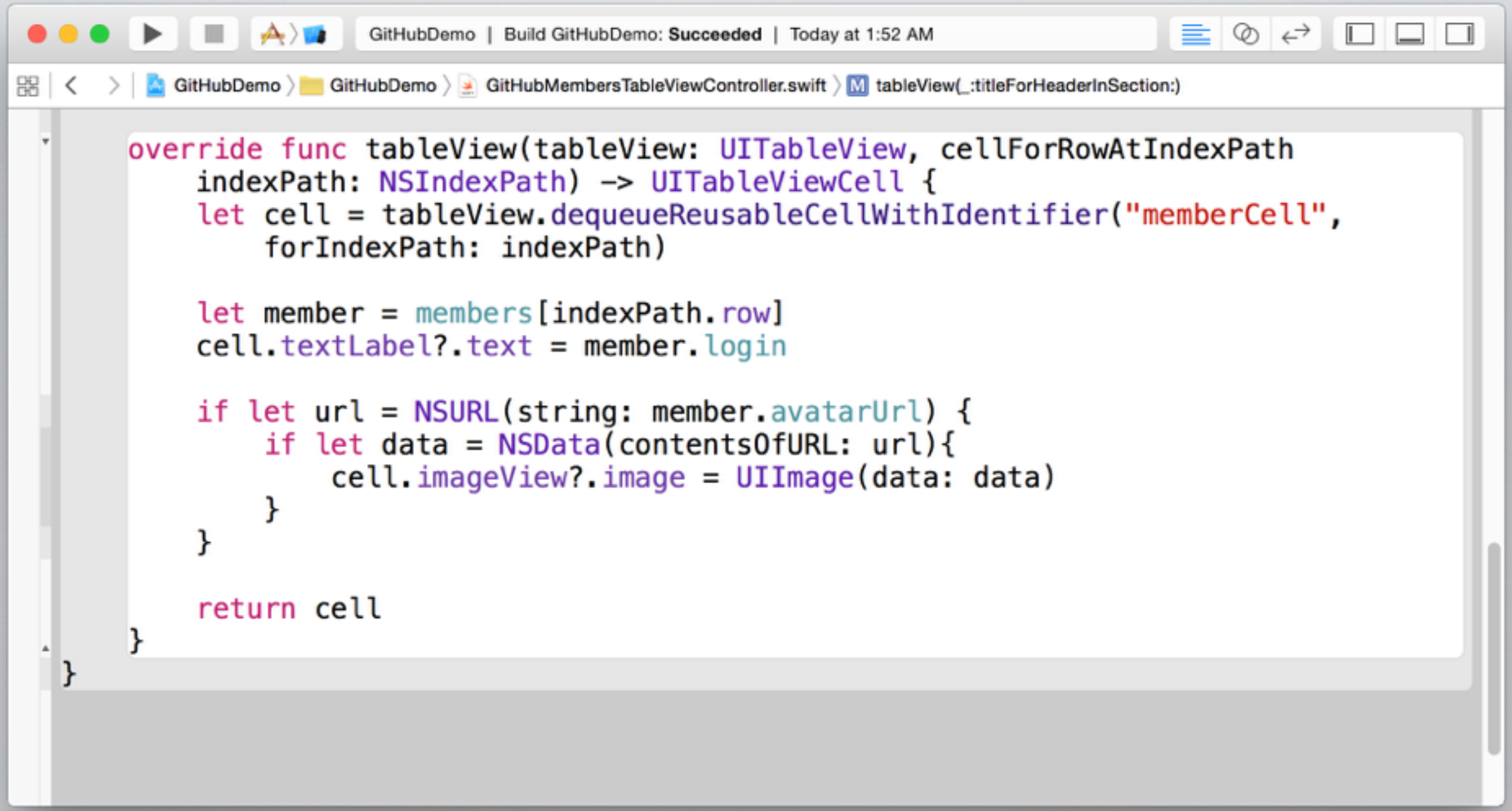


UITABLEVIEWDATASOURCE

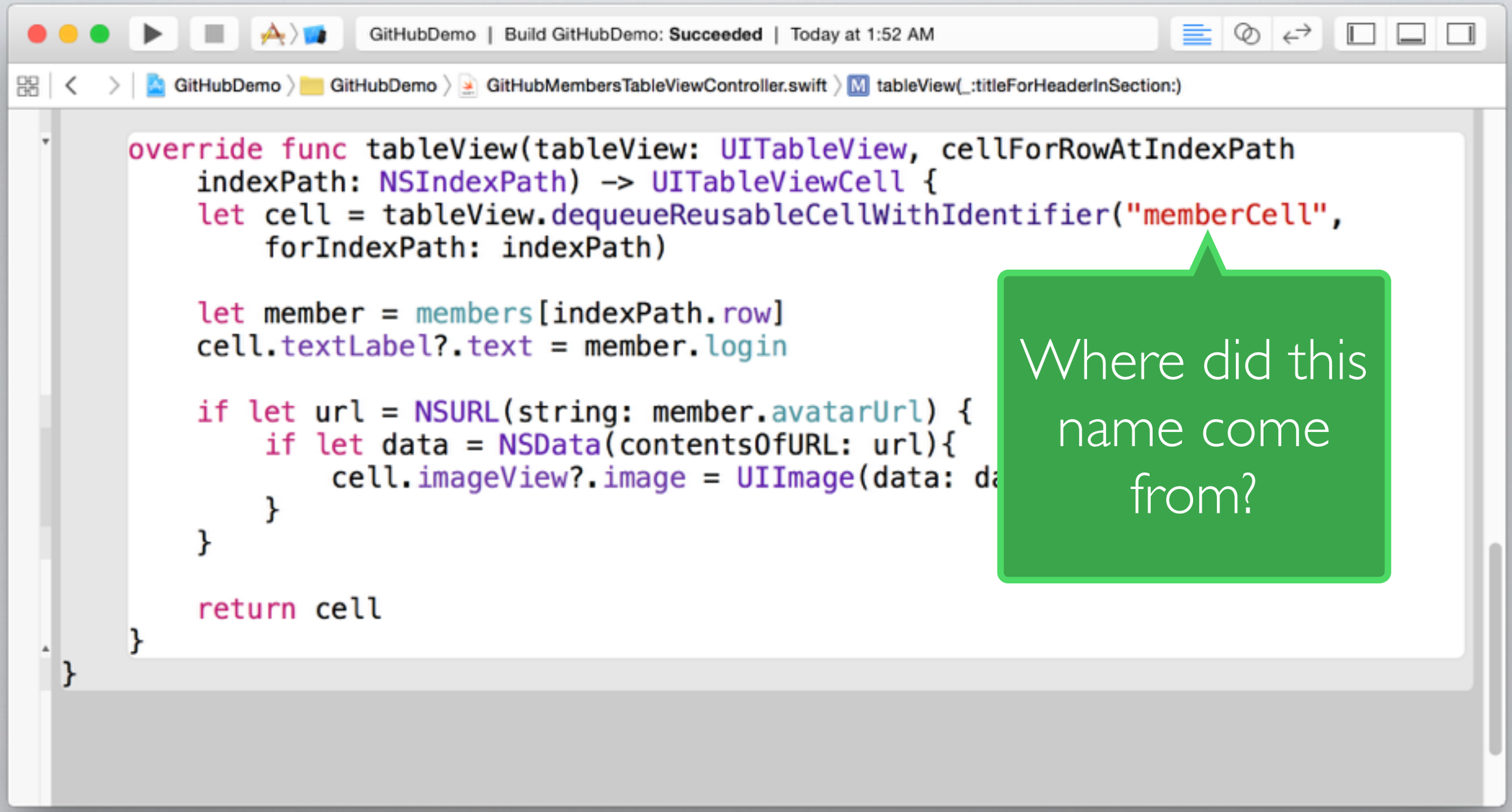


UITABLEVIEWDATASOURCE

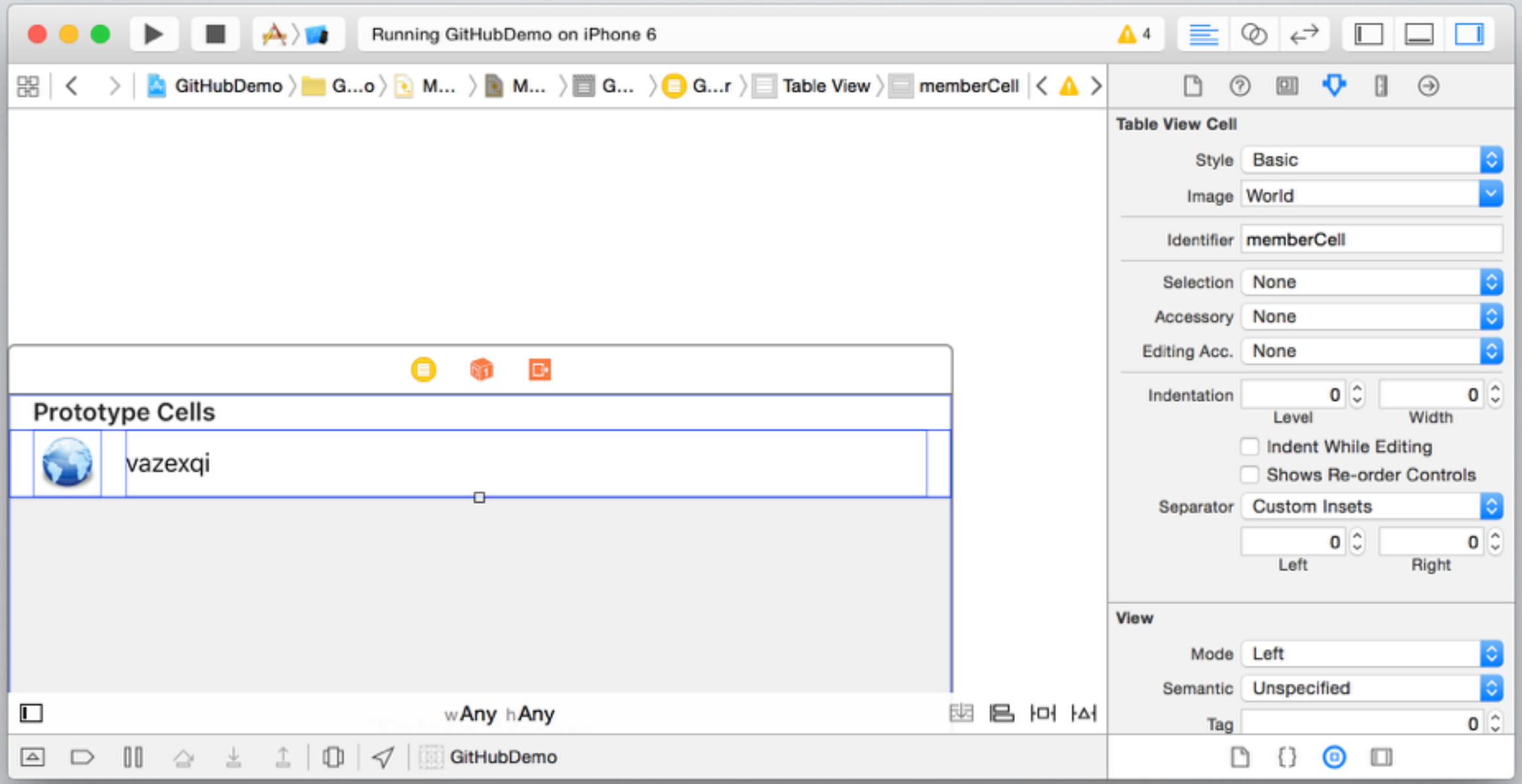
UITABLEVIEWDATASOURCE



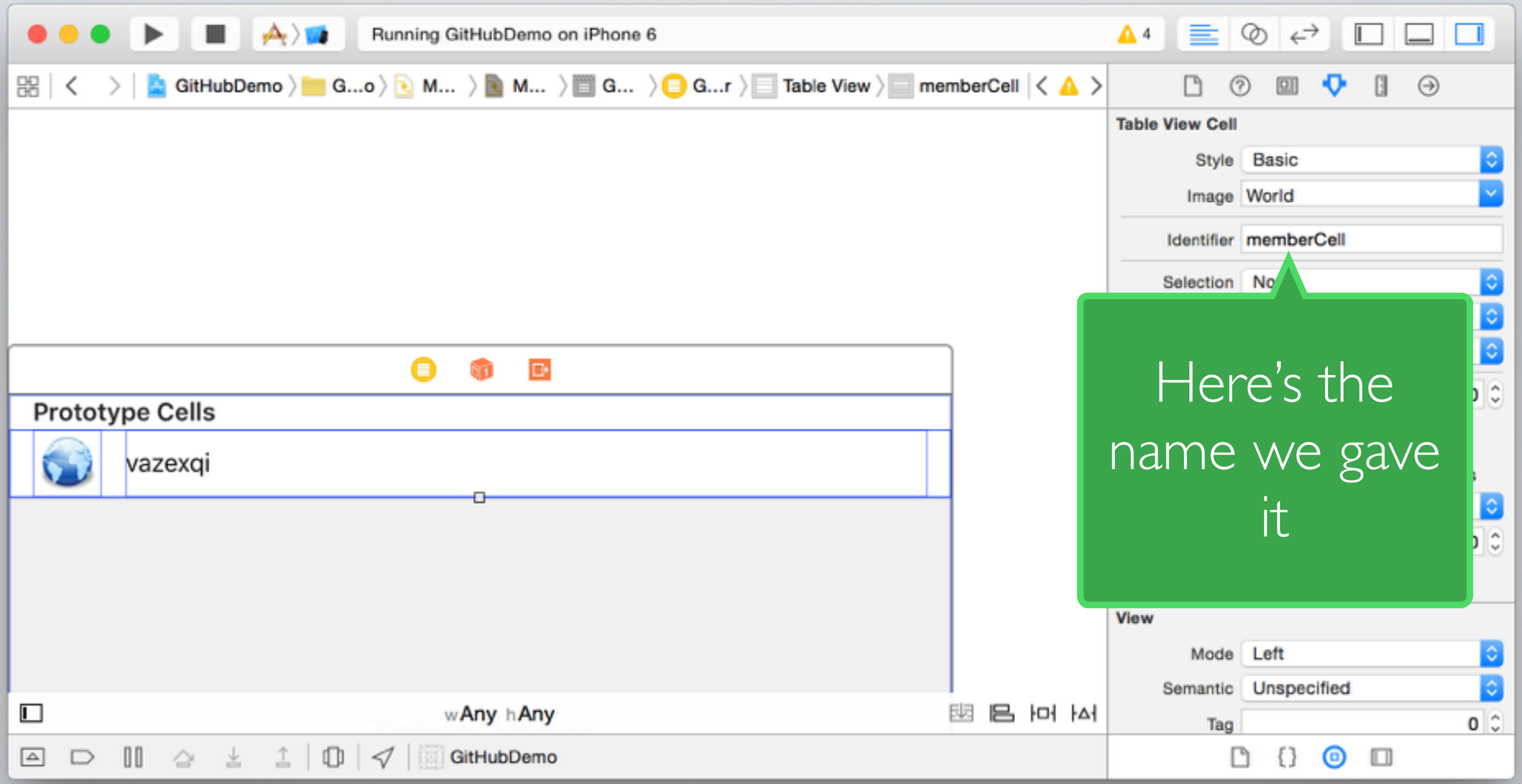
UITABLEVIEWDATASOURCE



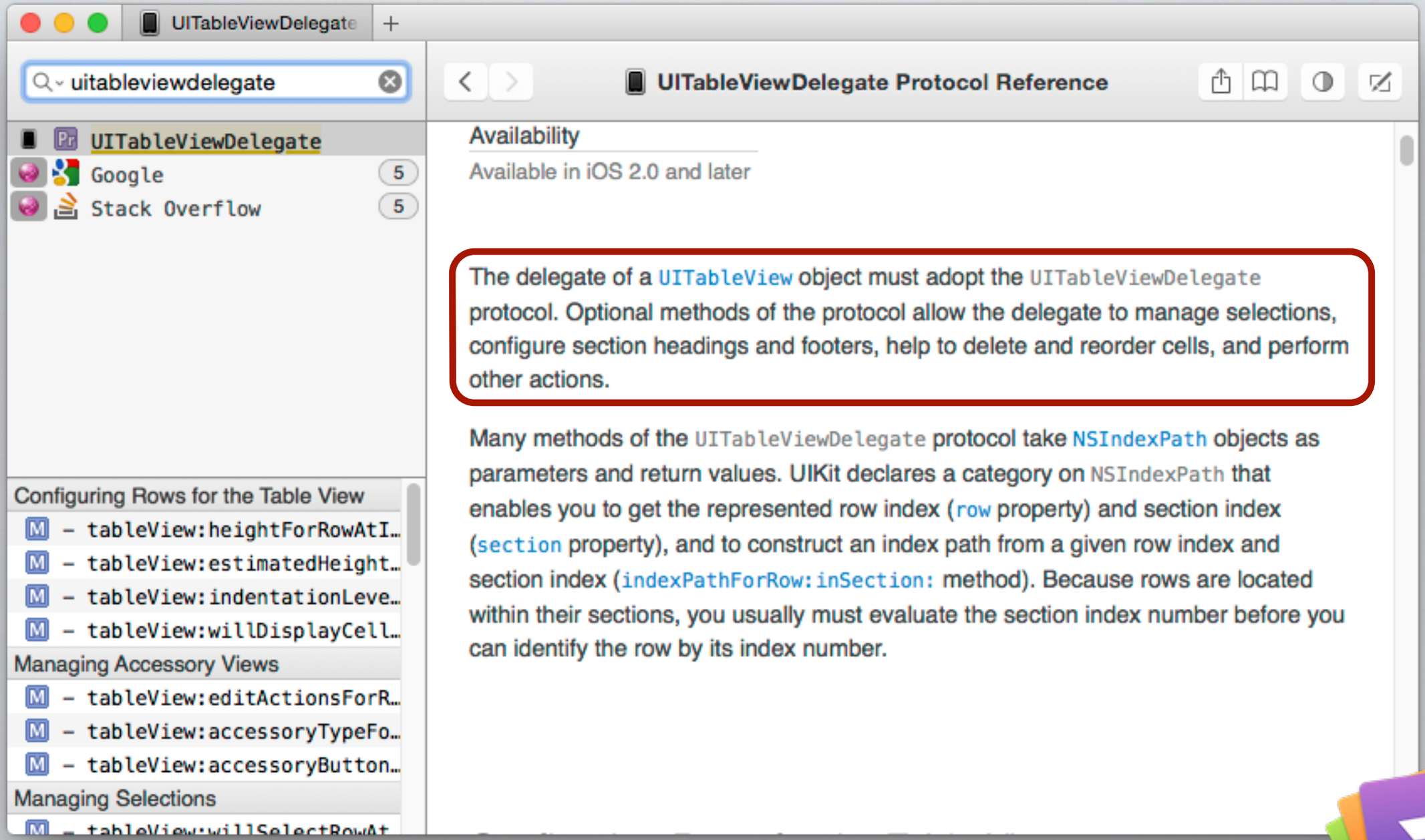
UITABLEVIEWDATASOURCE



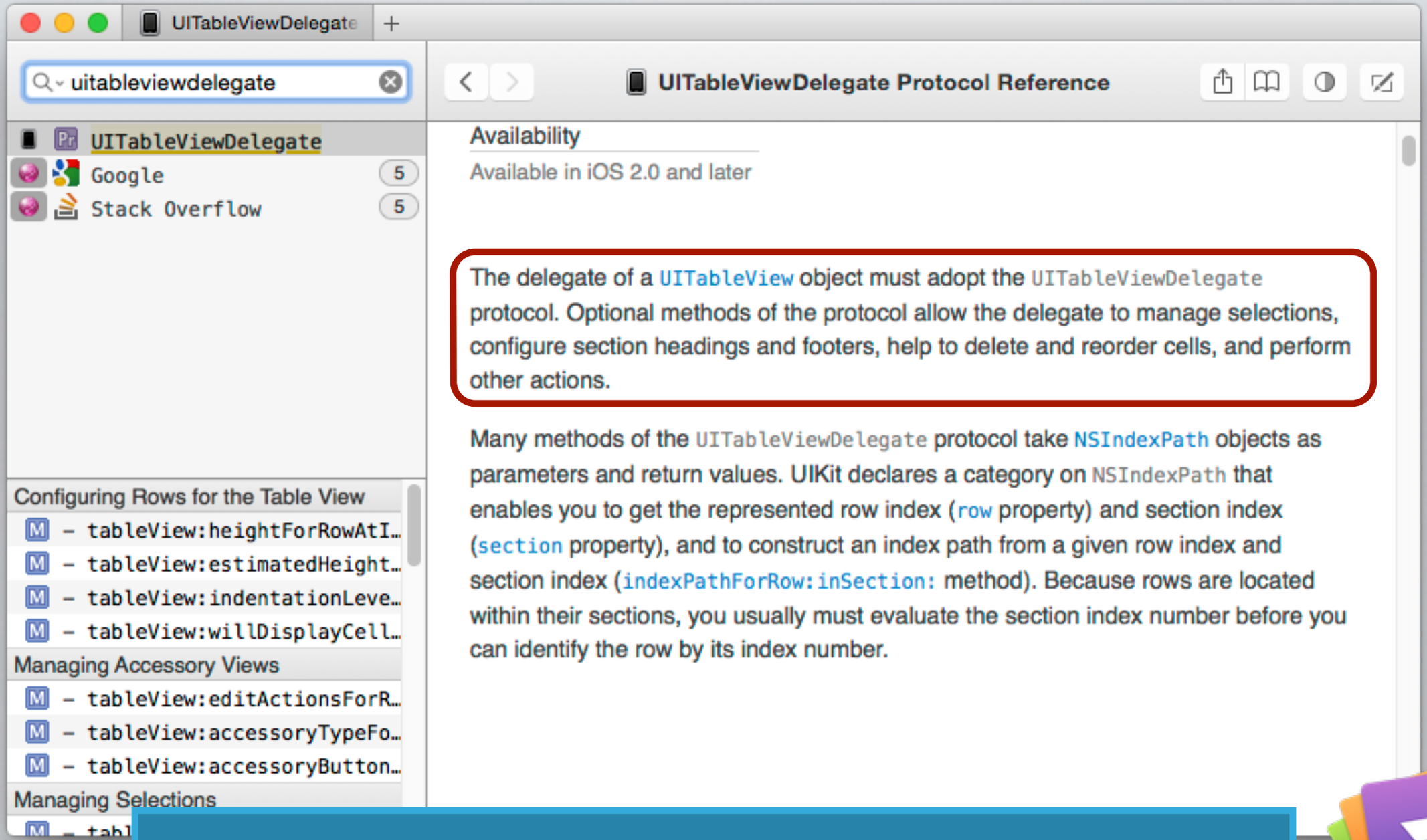
UITABLEVIEWDATASOURCE



UITABLEVIEWDELEGATE

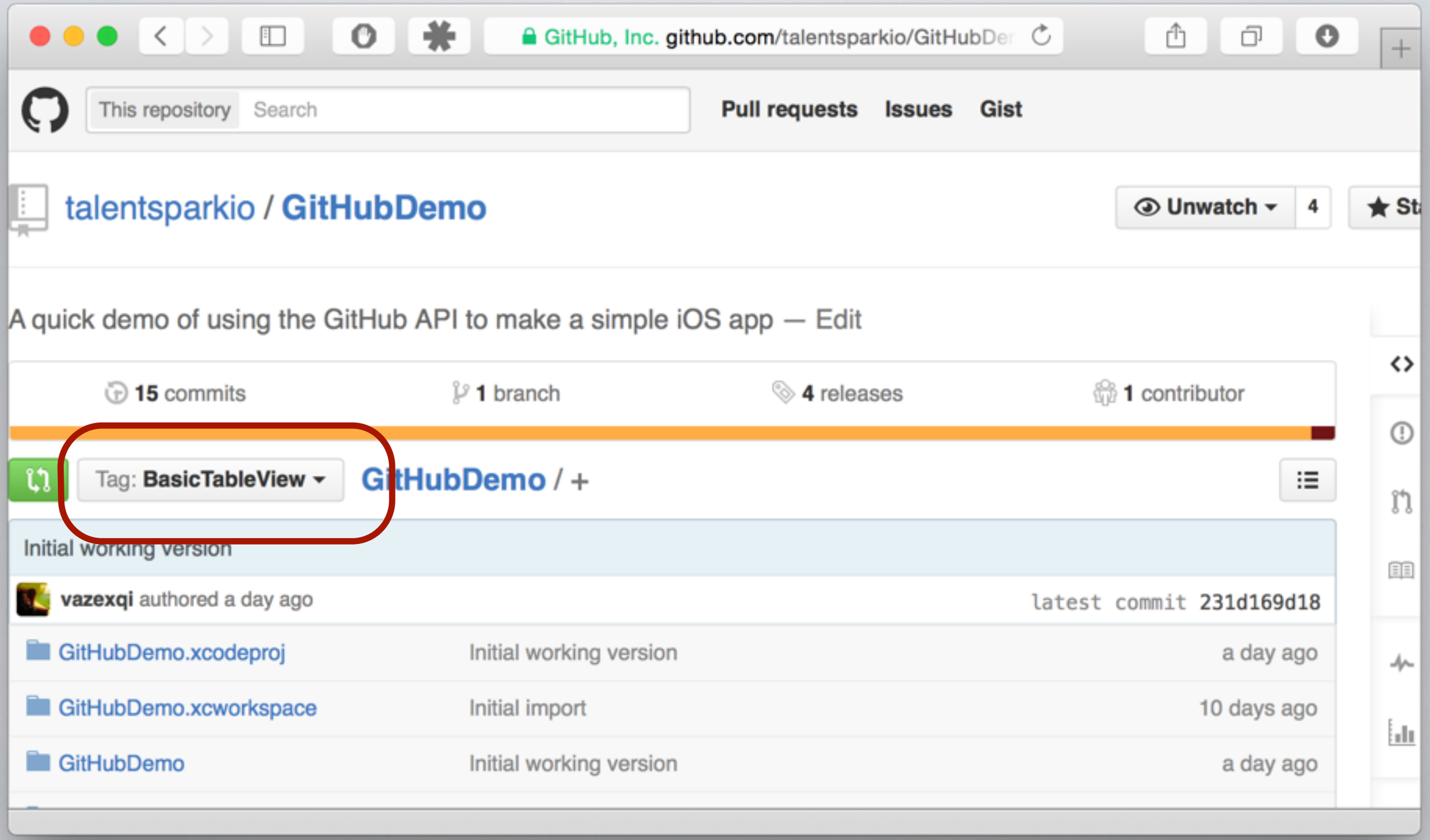


UITABLEVIEWDELEGATE

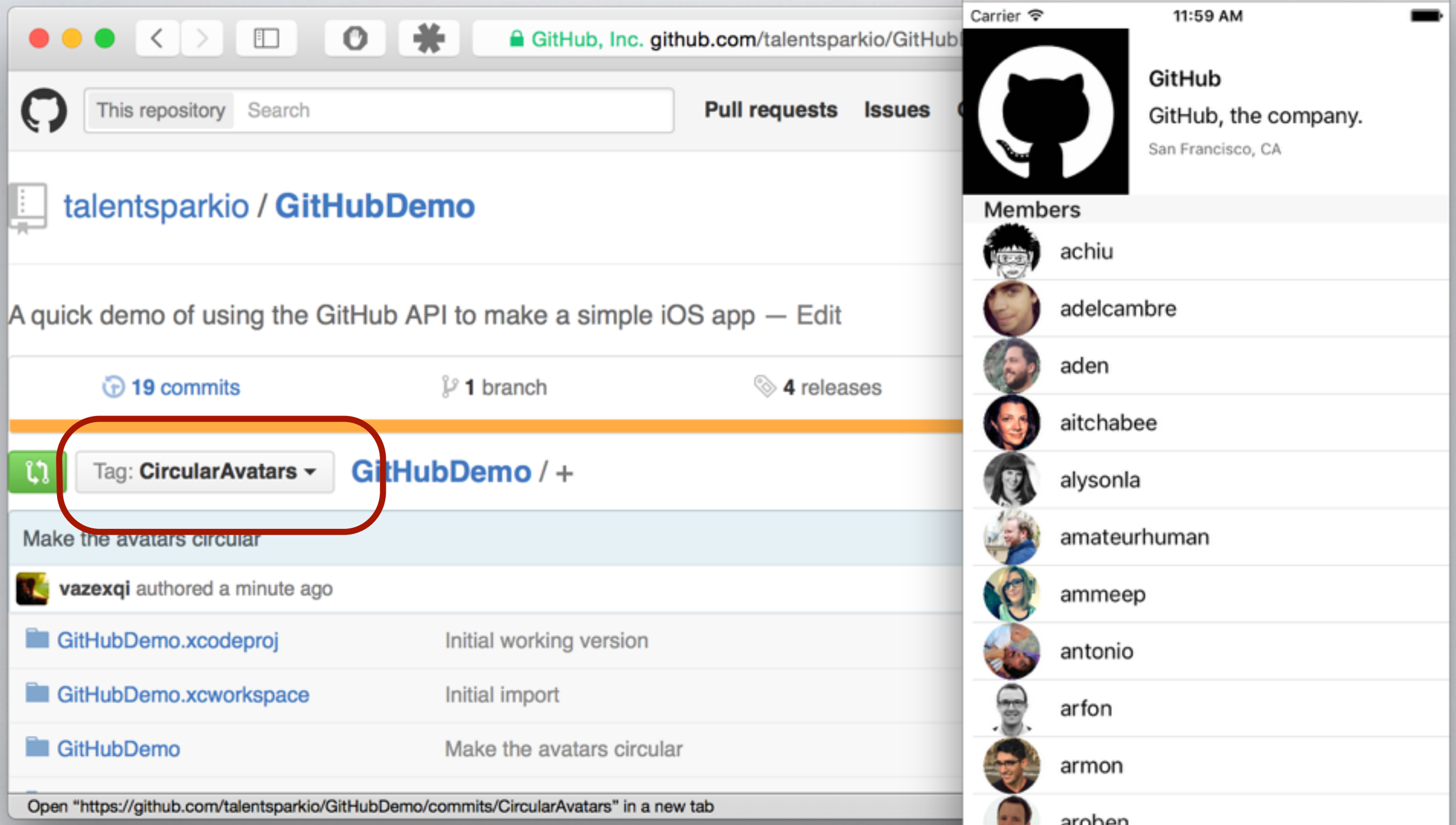


Didn’t use it this time

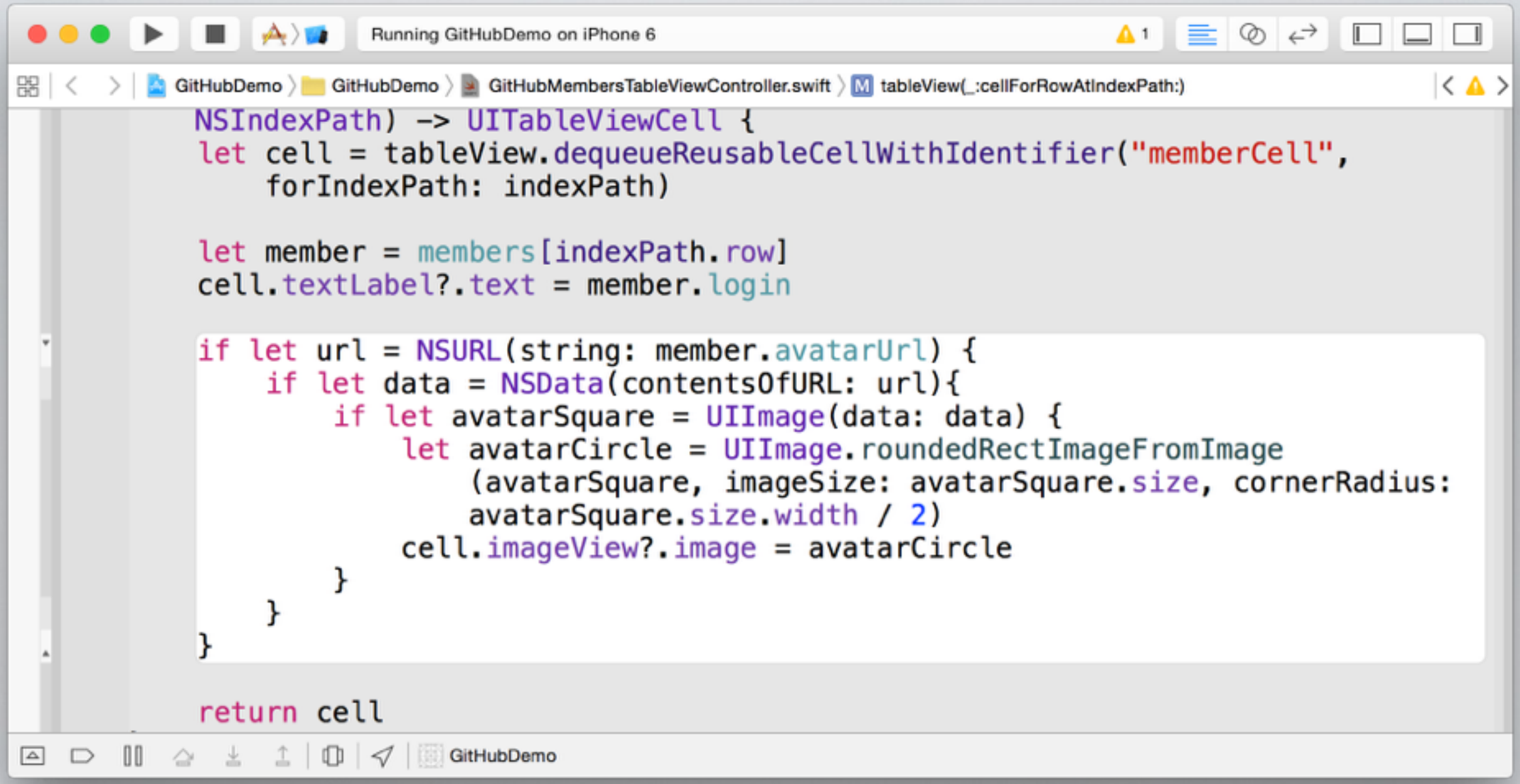
CODE


GitHub, Inc. github.com/talentsparkio/GitHubDe
This repository
Search
Pull requests
Issues
Gist
talentsparkio / GitHubDemo
Unwatch
4
A quick demo of using the GitHub API to make a simple iOS app — Edit
15 commits
1 branch
4 releases
1 contributor
Tag: BasicTableView
GitHubDemo / +
Initial working version
vazexqi authored a day ago
latest commit 231d169d18
GitHubDemo.xcodeproj
Initial working version
a day ago
GitHubDemo.xcworkspace
Initial import
10 days ago
GitHubDemo
Initial working version
a day ago

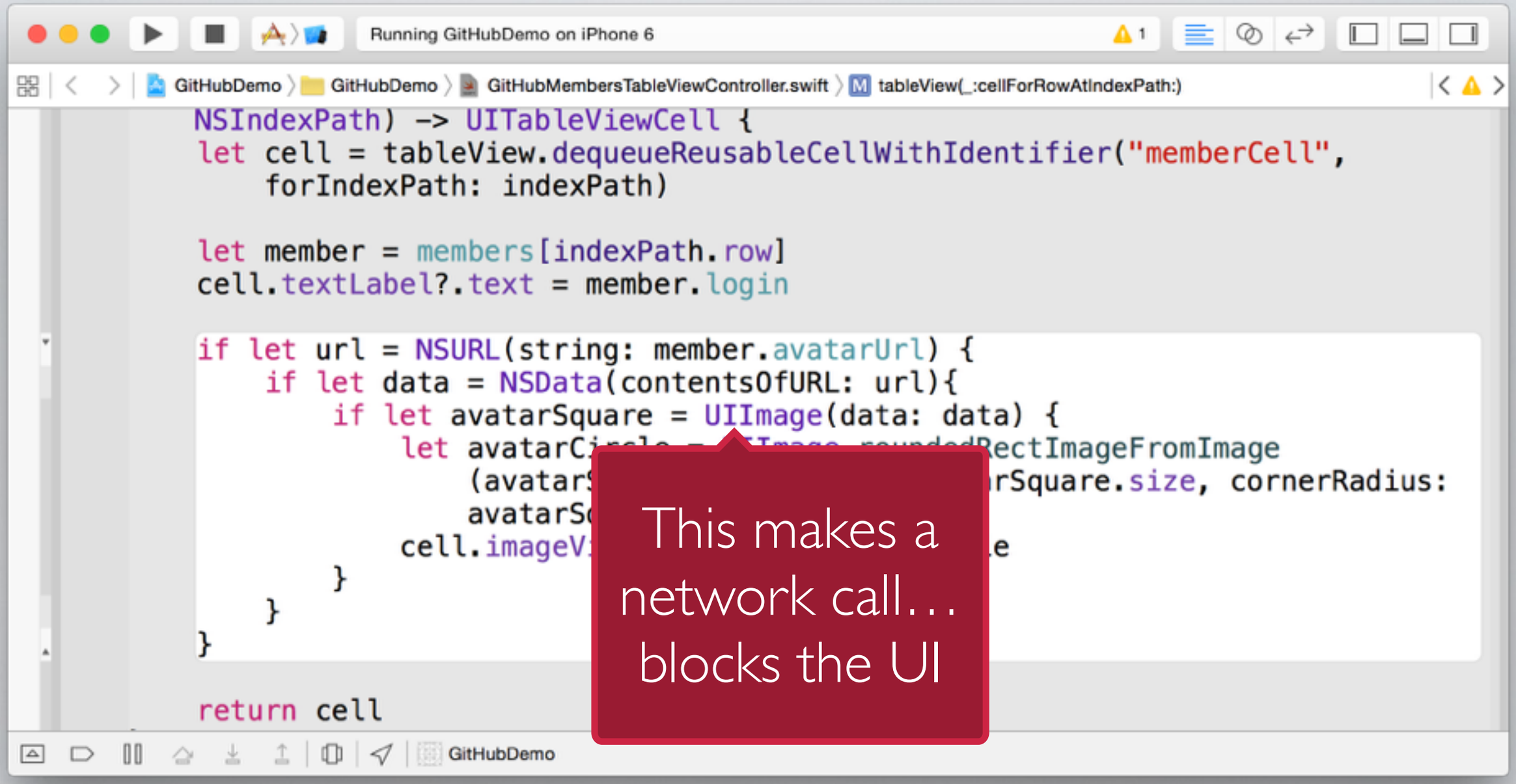
CODE


GitHub, Inc. github.com/talentsparkio/GitHub
This repository
Search
Pull requests
Issues
talentsparkio / GitHubDemo
A quick demo of using the GitHub API to make a simple iOS app — Edit
19 commits
1 branch
4 releases
Tag: CircularAvatars
GitHubDemo / +
Make the avatars circular
vazexqi authored a minute ago
GitHubDemo.xcodeproj
Initial working version
GitHubDemo.xcworkspace
Initial import
GitHubDemo
Make the avatars circular
Open "https://github.com/talentsparkio/GitHubDemo/commits/CircularAvatars" in a new tab
iPhone 6 - iPhone 6 / iOS 9.0 (13A4325c)
Carrier
11:59 AM
GitHub
GitHub, the company.
San Francisco, CA
Members
achiu
adelcambre
aden
aitchabee
alysonla
amateurhuman
ammeep
antonio
arfon
armon
aroben

BONUS (GCD)

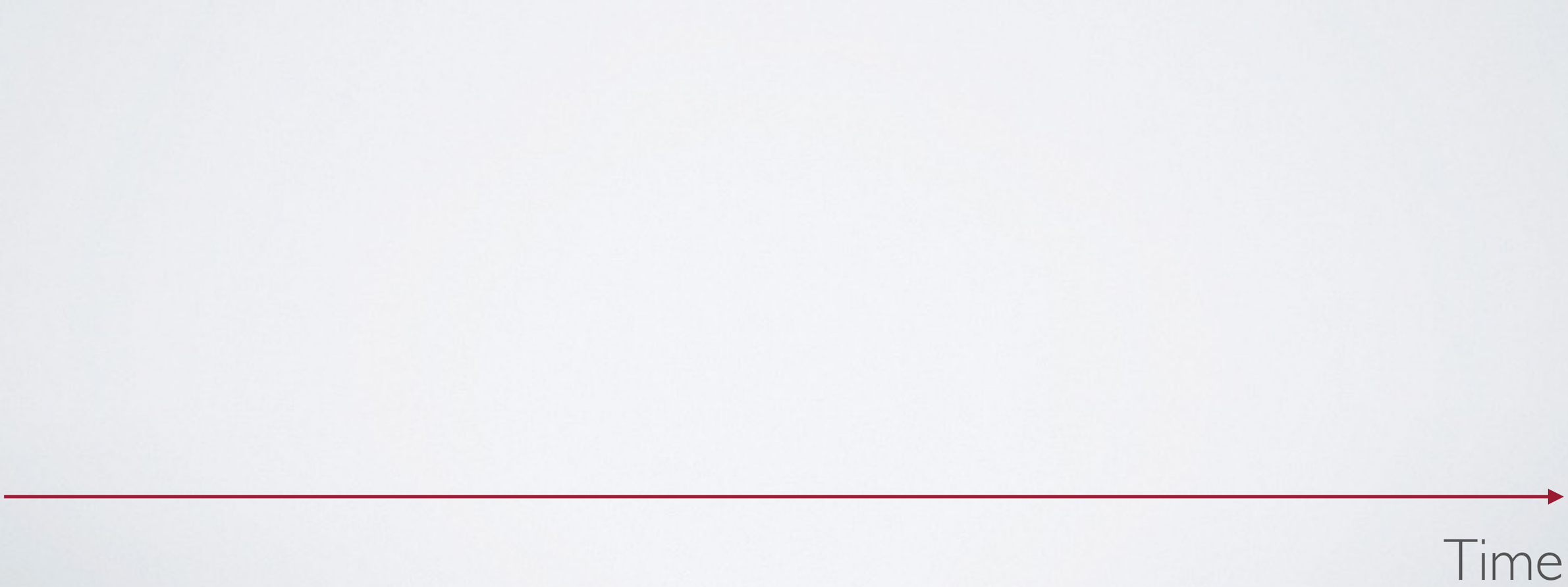

Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
GitHubMembersTableViewController.swift
tableView(_:cellForRowAtIndexPath:)
NSIndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
forIndexPath: indexPath)
let member = members[indexPath.row]
cell.textLabel?.text = member.login
if let url = NSURL(string: member.avatarUrl) {
if let data = NSData(contentsOfURL: url){
if let avatarSquare = UIImage(data: data) {
let avatarCircle = UIImage.roundedRectImageFromImage
(avatarSquare, imageSize: avatarSquare.size, cornerRadius:
avatarSquare.size.width / 2)
cell.imageView?.image = avatarCircle
}
}
}
return cell
GitHubDemo

BONUS (GCD)


Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
GitHubMembersTableViewController.swift
tableView(_:cellForRowAtIndexPath:)
NSIndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
forIndexPath: indexPath)
let member = members[indexPath.row]
cell.textLabel?.text = member.login
if let url = NSURL(string: member.avatarUrl) {
if let data = NSData(contentsOfURL: url){
if let avatarSquare = UIImage(data: data) {
return cell
This makes a network call… blocks the UI
GitHubDemo

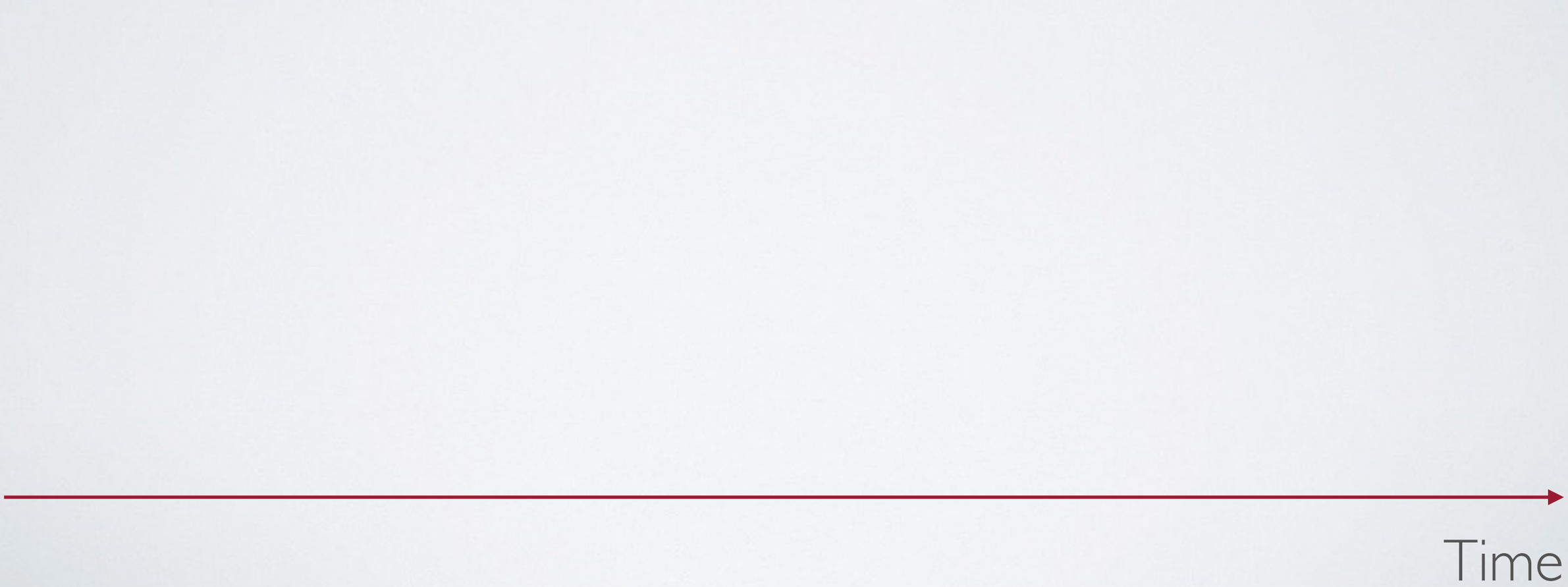
BONUS (GCD)

BONUS (GCD)

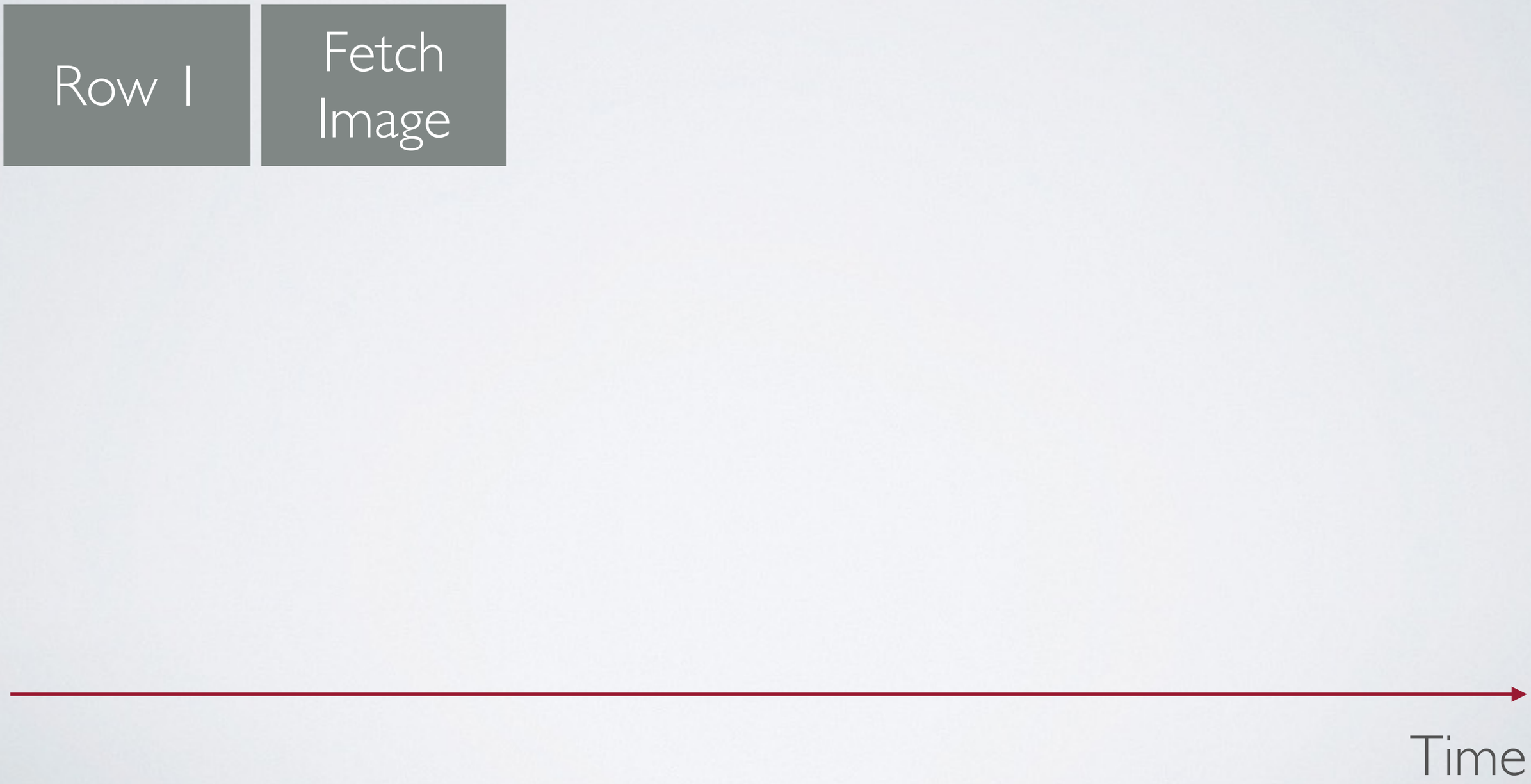


BONUS (GCD)

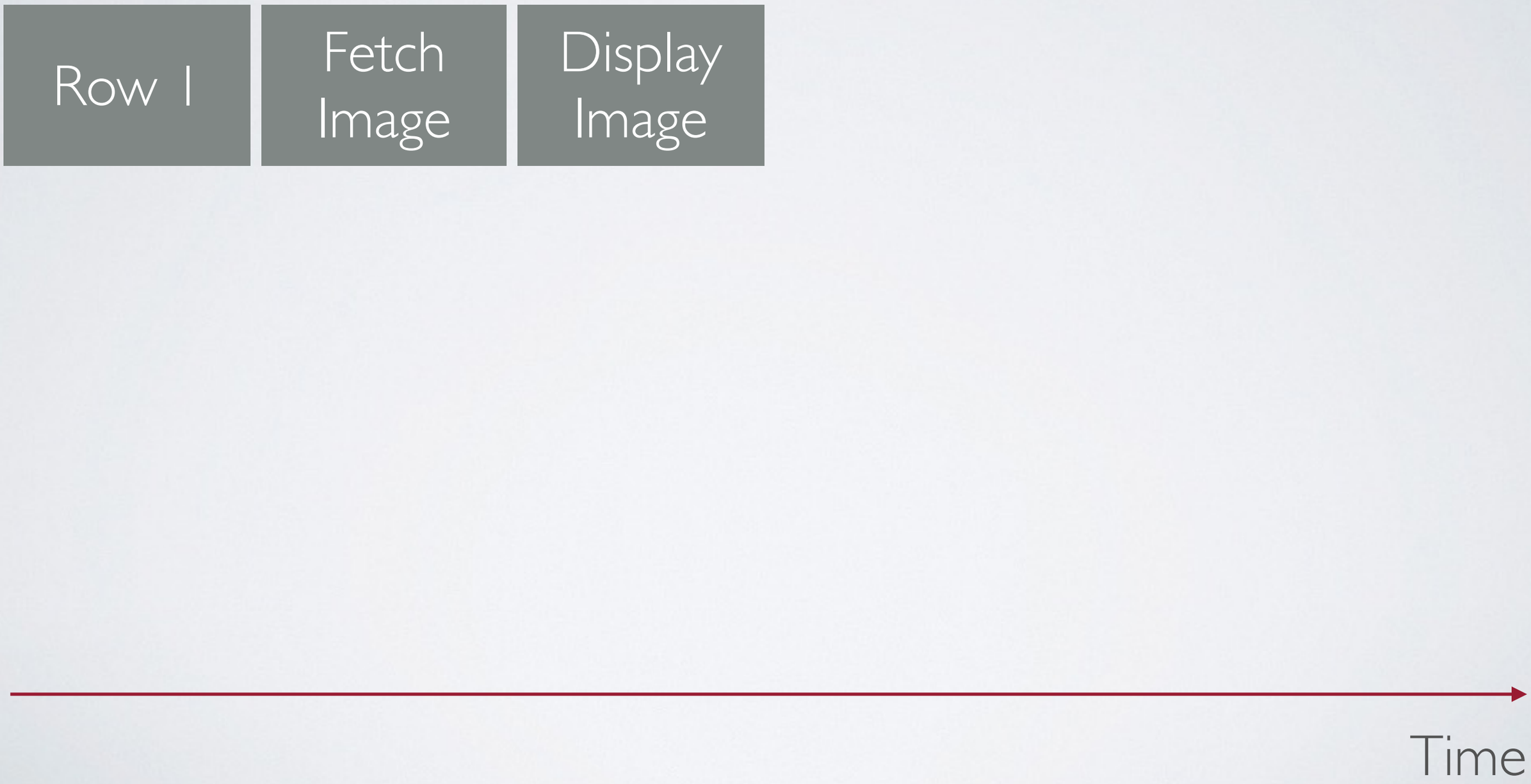
Row 1



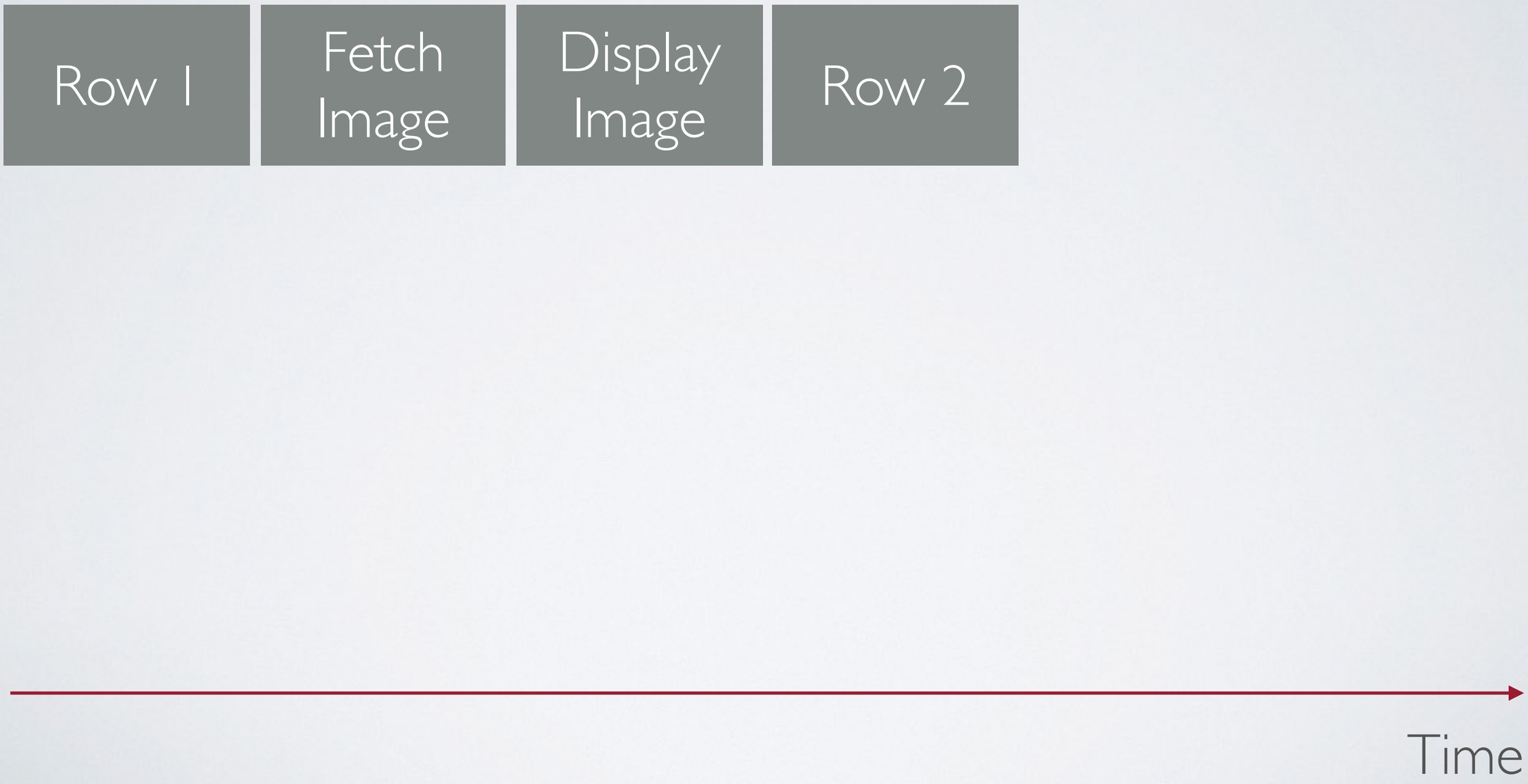
BONUS (GCD)



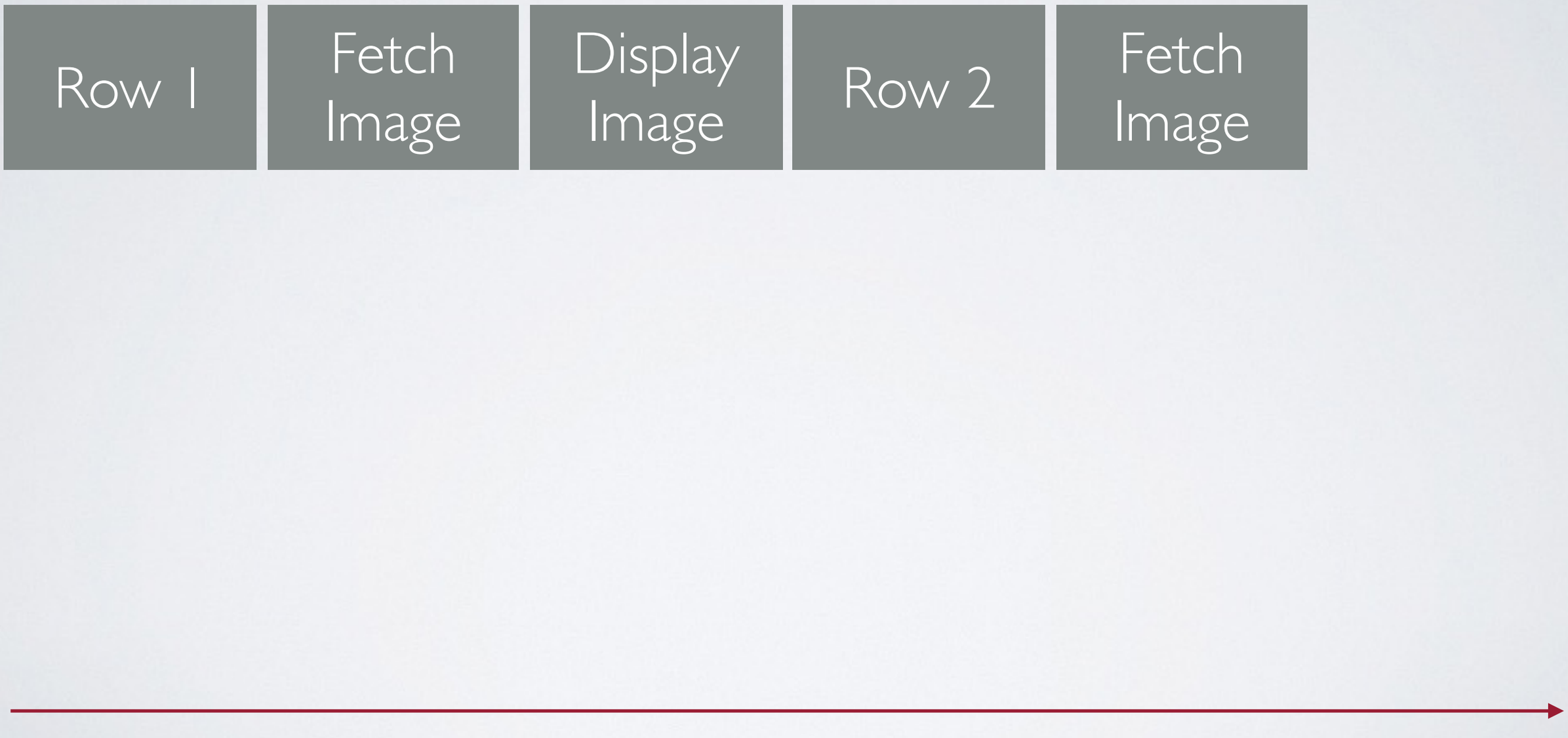
BONUS (GCD)



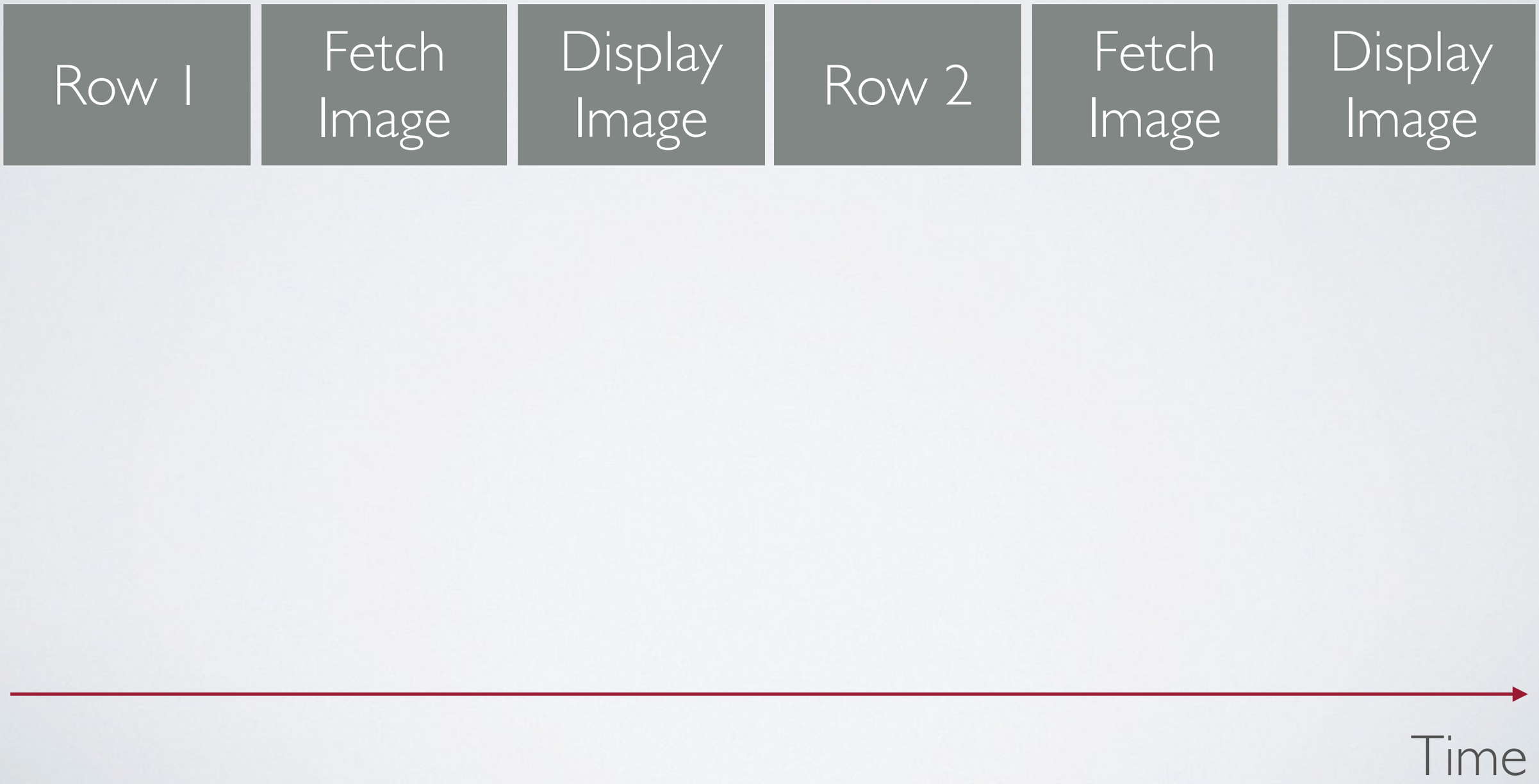
BONUS (GCD)



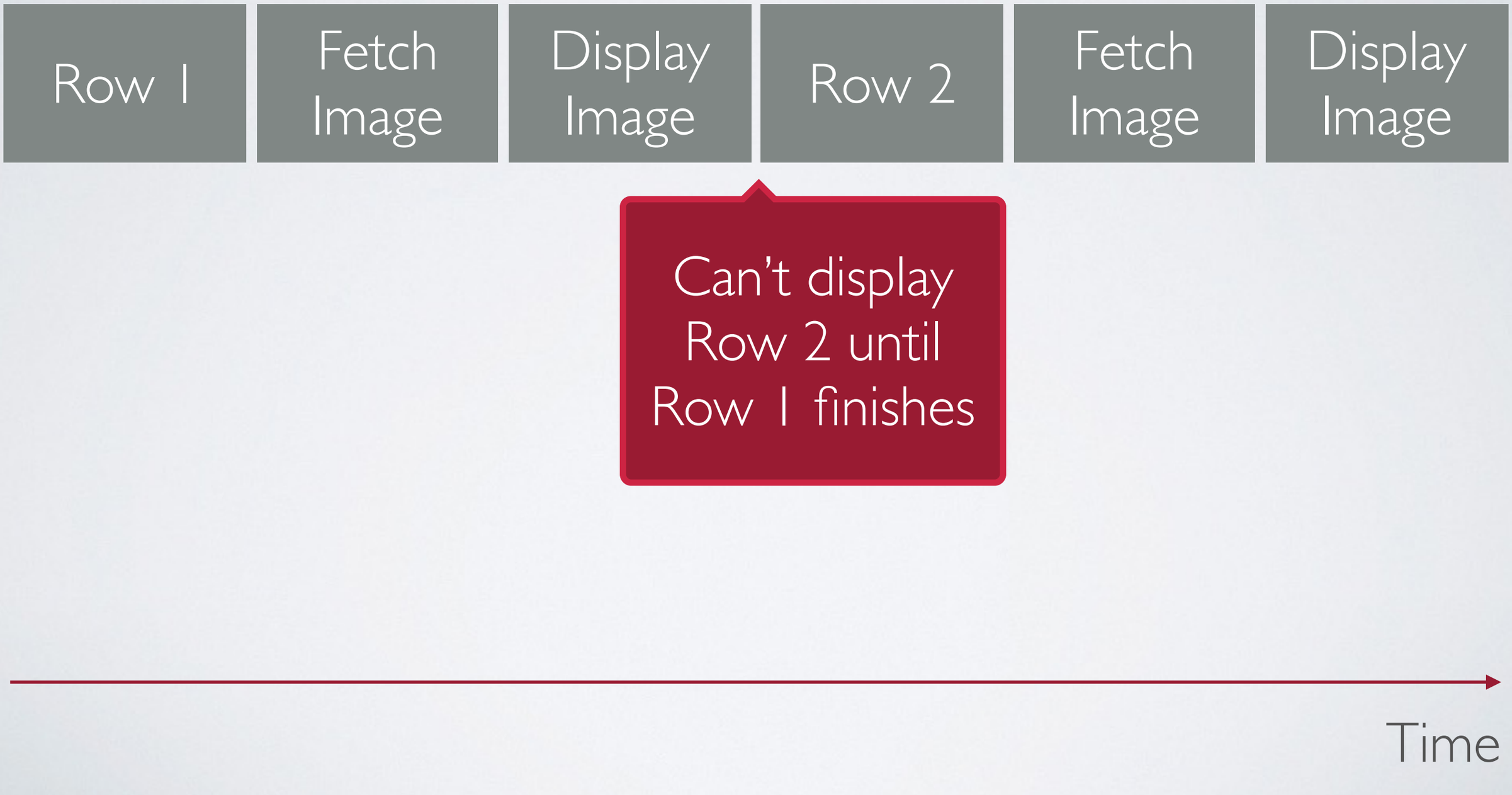
BONUS (GCD)



BONUS (GCD)


Row 1
Fetch Image
Display Image
Row 2
Fetch Image
Display Image
Time

BONUS (GCD)


Row 1
Fetch Image
Display Image
Row 2
Fetch Image
Display Image
Can't display Row 2 until Row 1 finishes
Time

BONUS (GCD)

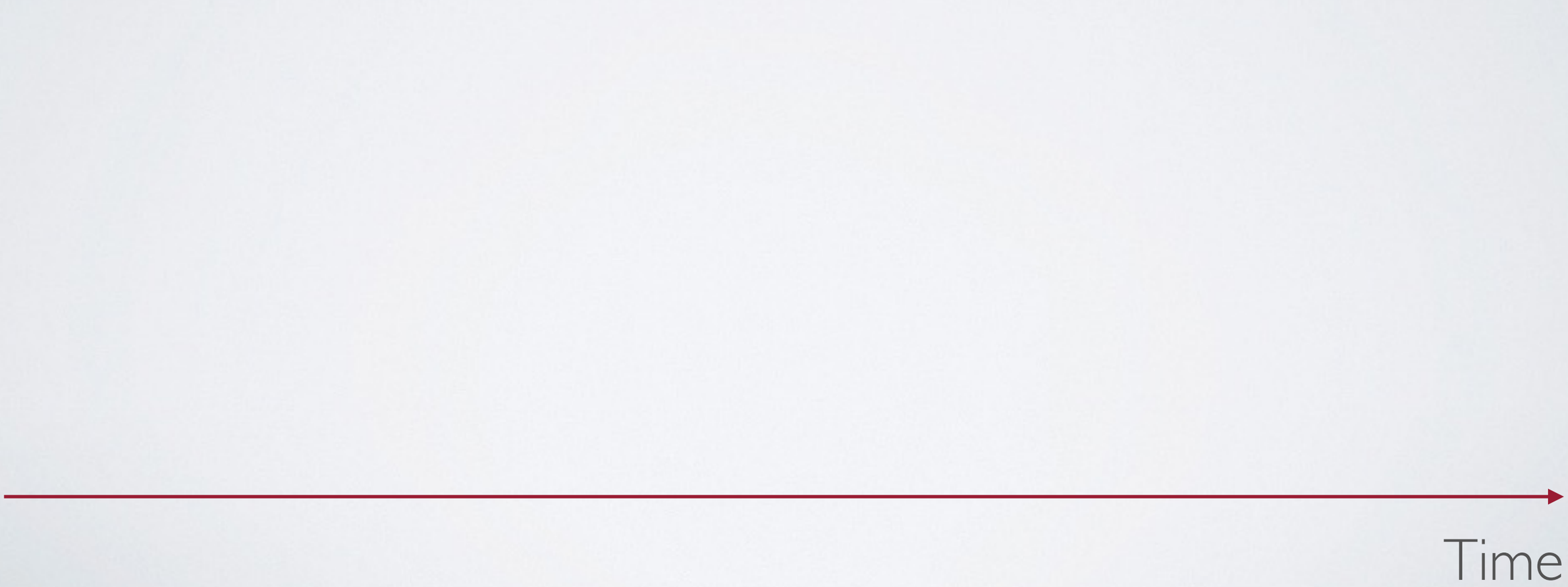
Row 1	Fetch Image	Display Image	Row 2	Fetch Image	Display Image

Can't display Row 2 until Row 1 finishes

💡 Can we at least display the text first and images laters?

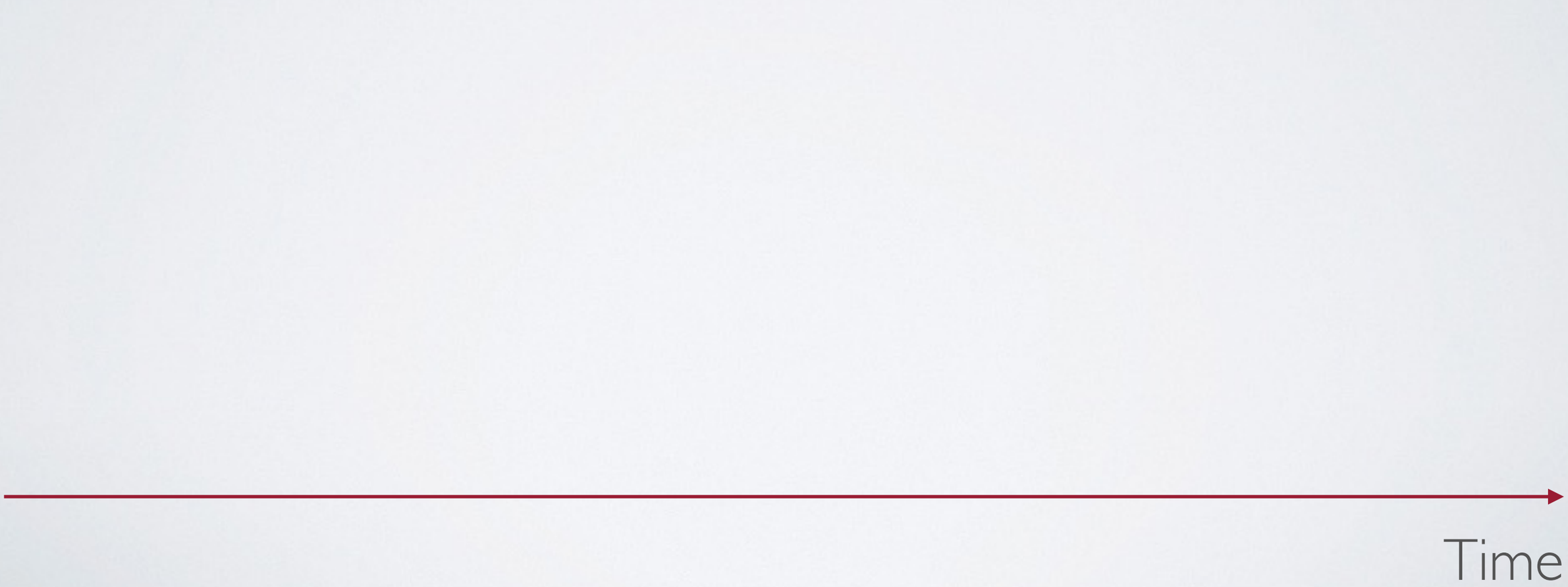
Time

BONUS (GCD)



BONUS (GCD)

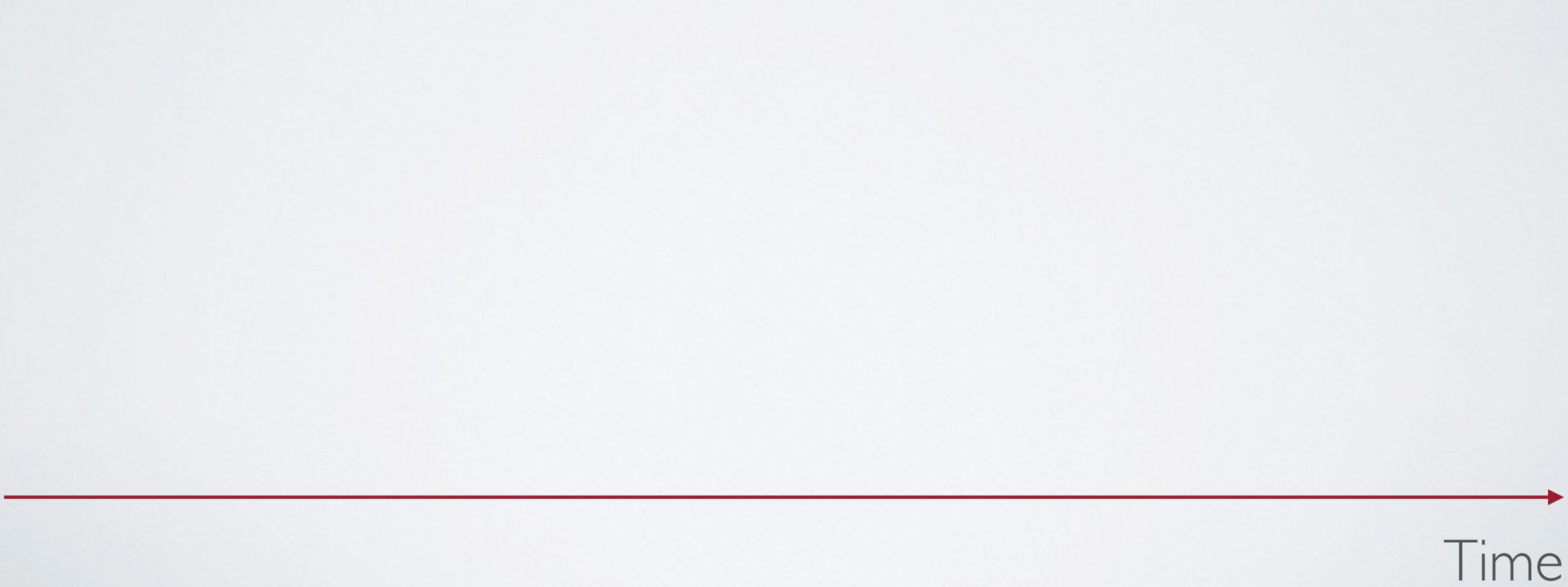
Row 1



BONUS (GCD)

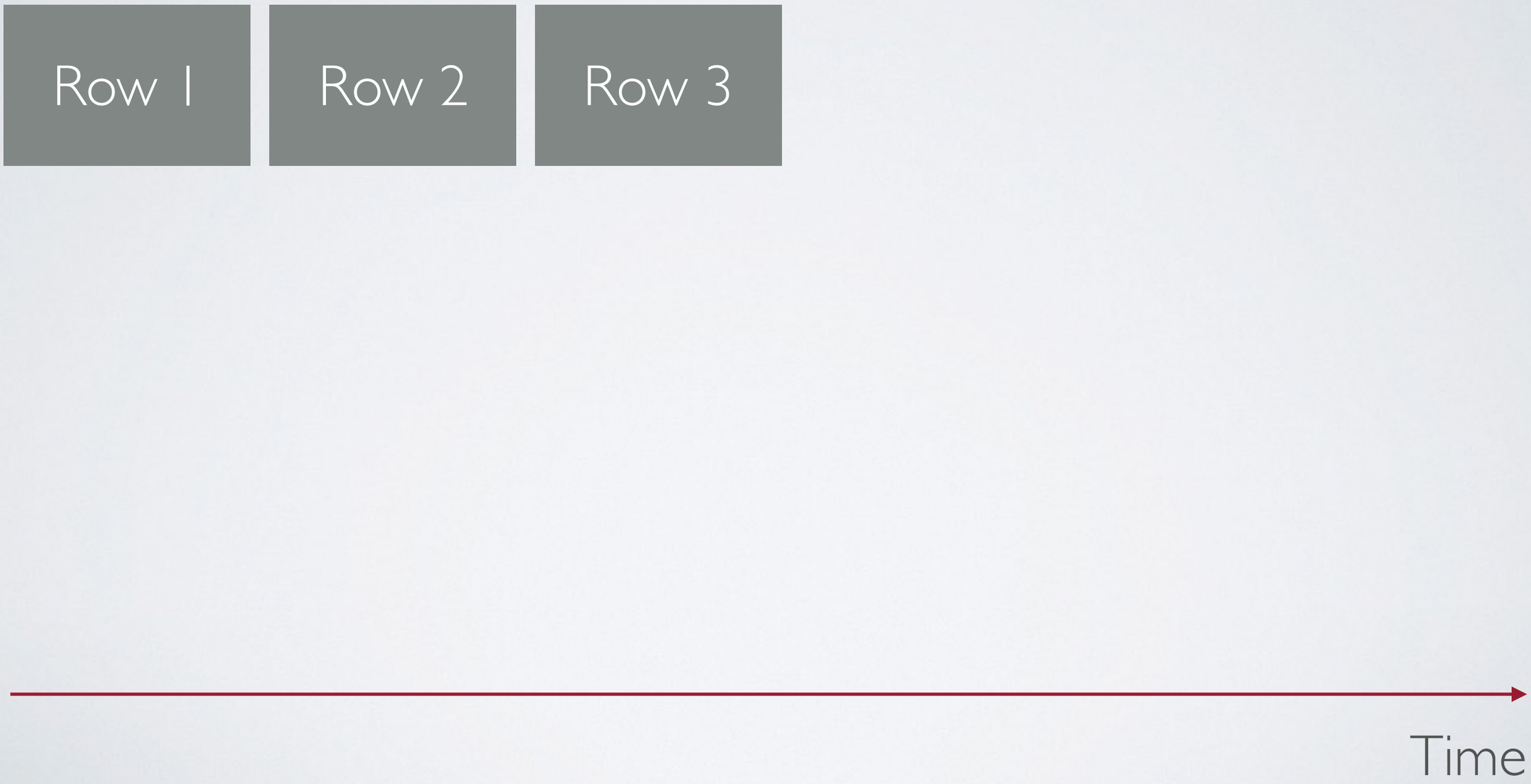
Row 1

Row 2



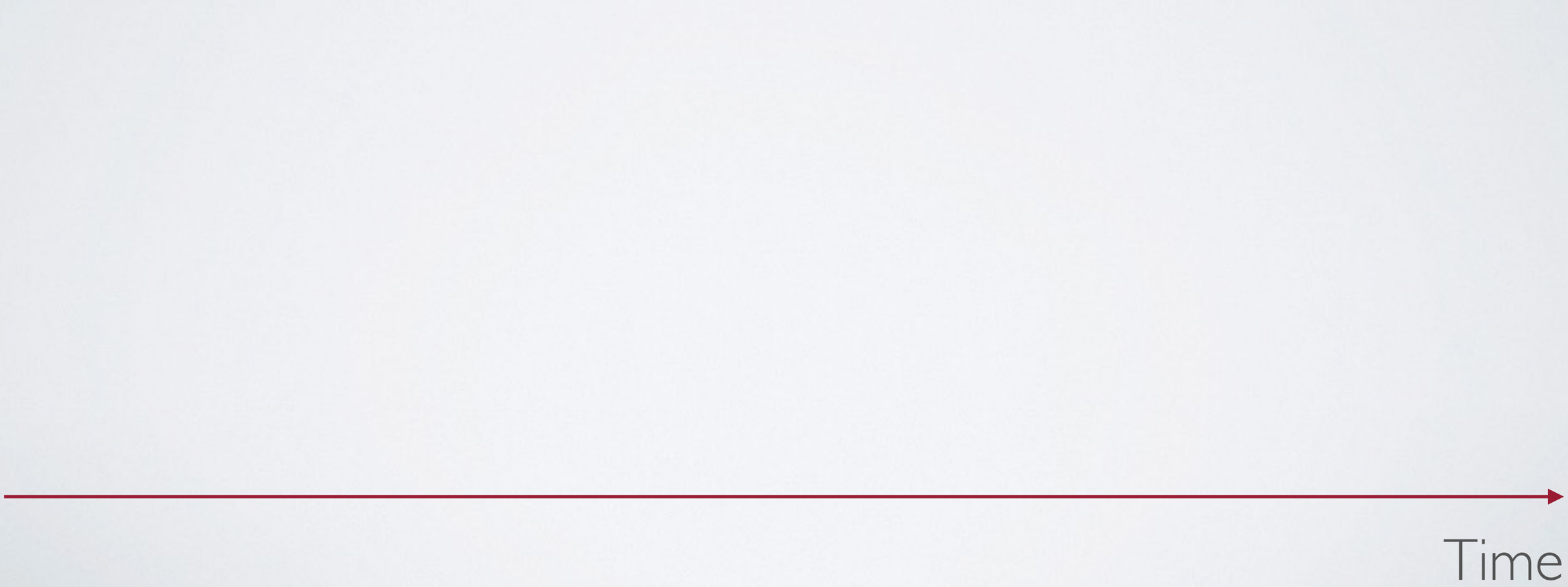
BONUS (GCD)

Row 1 | Row 2 | Row 3



BONUS (GCD)

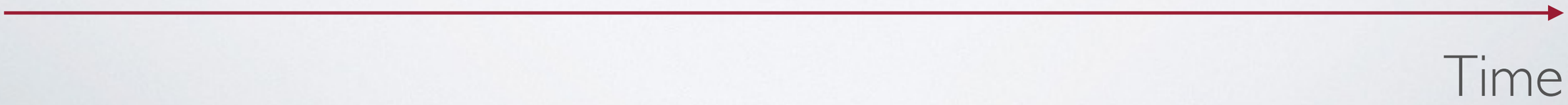
Row 1 | Row 2 | Row 3 | ...



BONUS (GCD)

Row 1 | Row 2 | Row 3 | …

Some time later

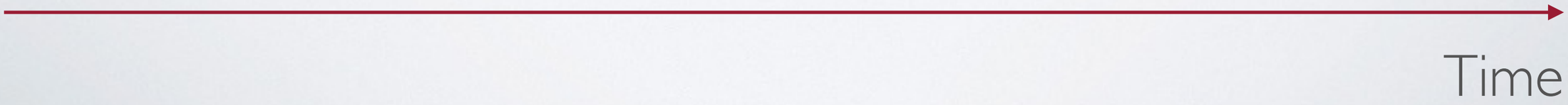


BONUS (GCD)

Row 1 | Row 2 | Row 3 | ...

Some time later

Fetch Image #1 | Display Image #1



BONUS (GCD)

Row 1 | Row 2 | Row 3 | ...

Some time later

Fetch Image #1 | Display Image #1 | Fetch Image #2

Time

BONUS (GCD)

Row 1 | Row 2 | Row 3 | ...

Some time later

Fetch Image #1 | Display Image #1 | Fetch Image #2

Time

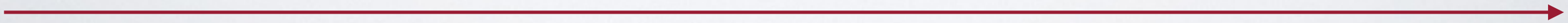
BONUS (GCD)

Row 1 | Row 2 | Row 3 | ...

Some time later

Fetch Image #1 | Display Image #1

Fetch Image #2

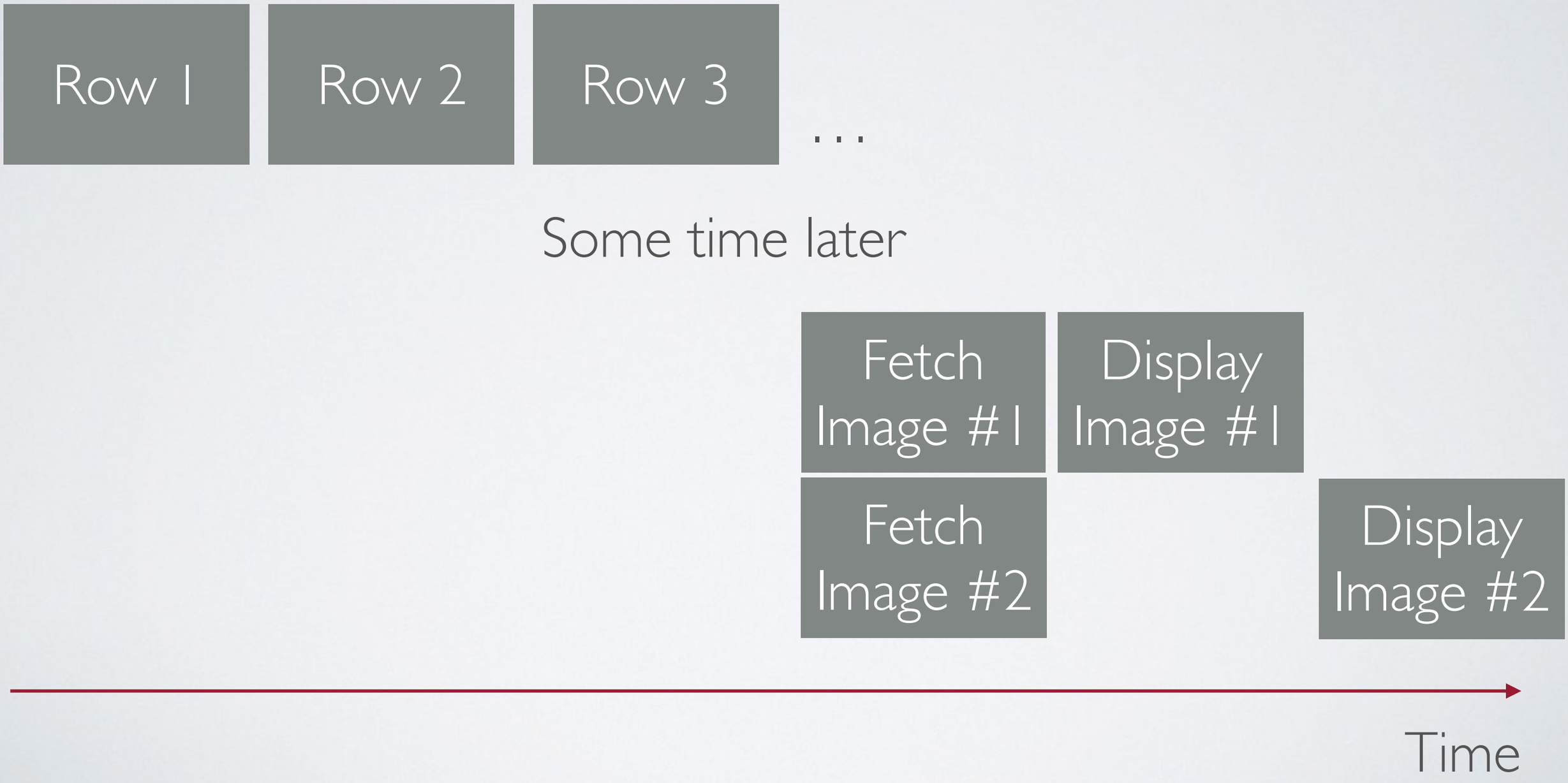


Time

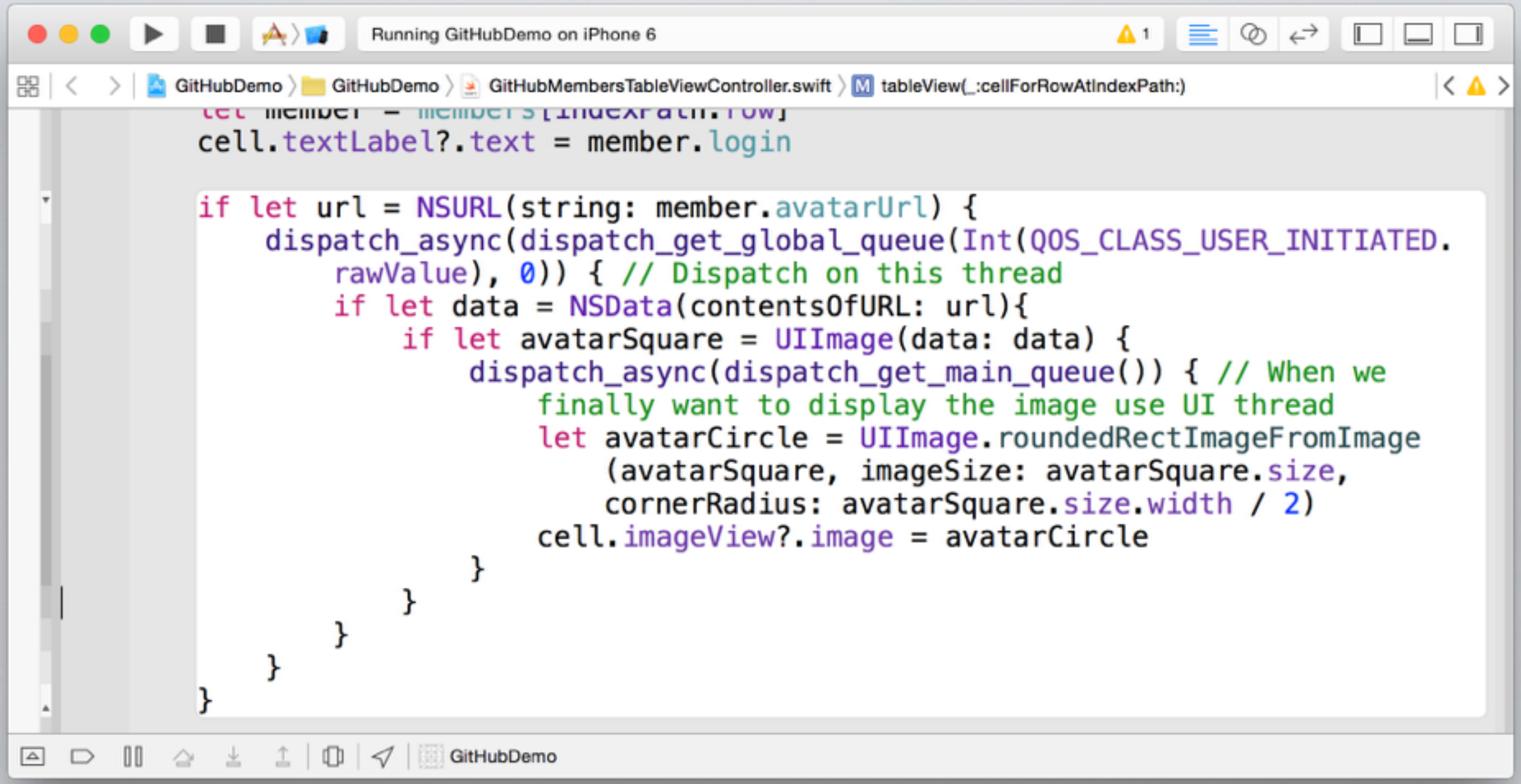
BONUS (GCD)

Row 1 | Row 2 | Row 3 | ...

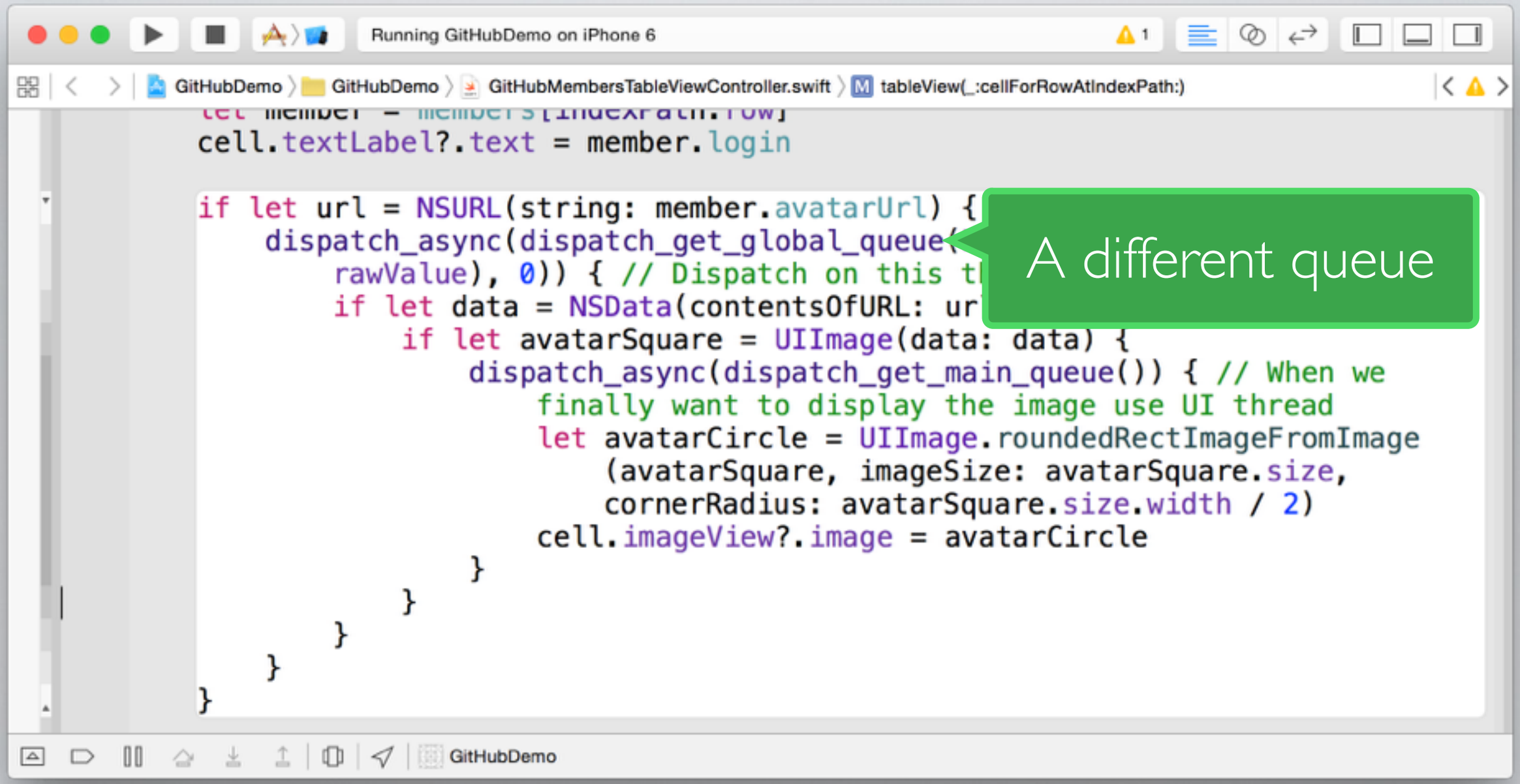
Some time later



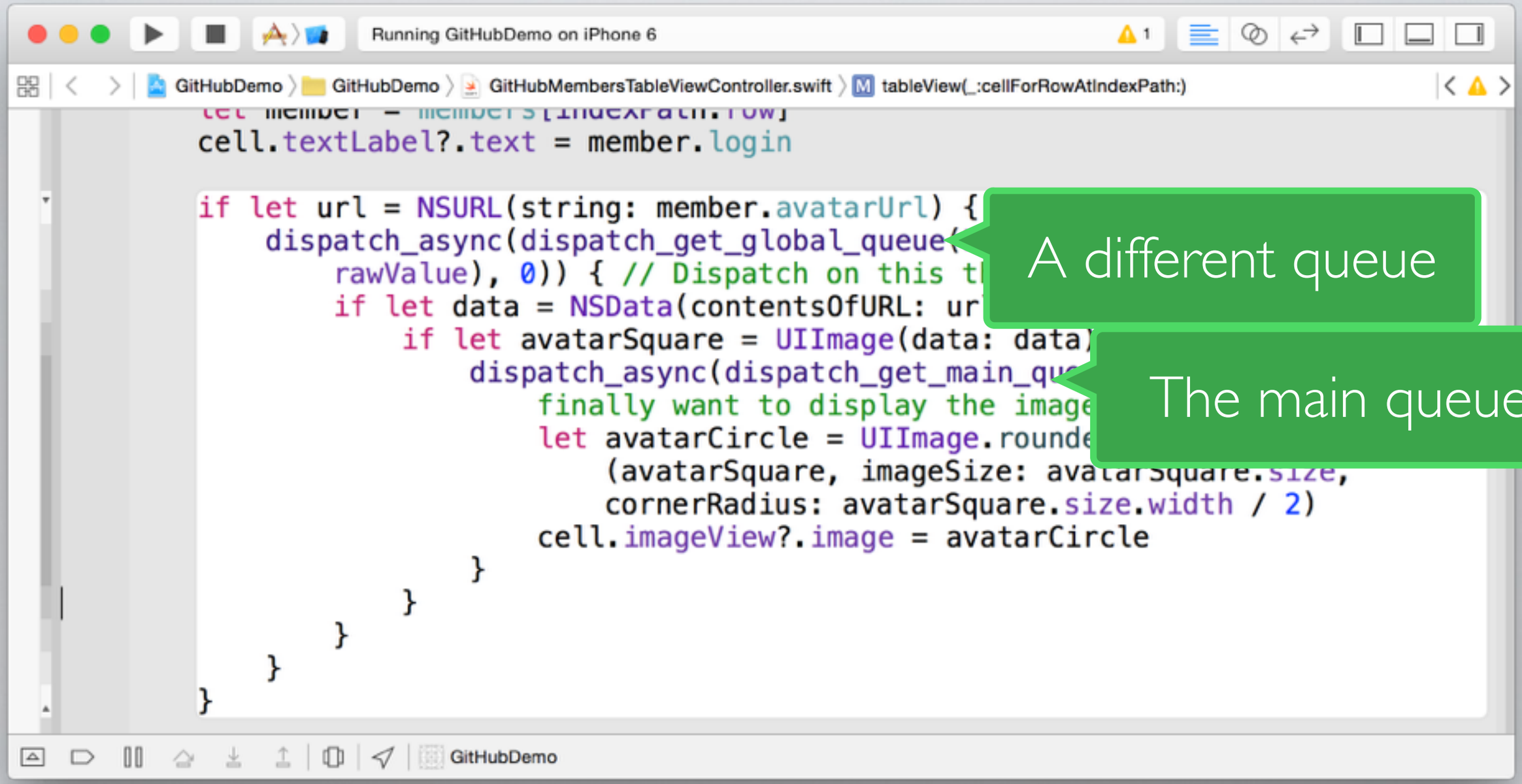
BONUS (GCD)


Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
GitHubMembersTableViewController.swift
tableView(_:cellForRowAtIndexPath:)
cell.textLabel?.text = member.login
if let url = NSURL(string: member.avatarUrl) {
dispatch_async(dispatch_get_global_queue(Int(QOS_CLASS_USER_INITIATED.
rawValue), 0)) { // Dispatch on this thread
if let data = NSData(contentsOfURL: url){
if let avatarSquare = UIImage(data: data) {
dispatch_async(dispatch_get_main_queue()) { // When we
finally want to display the image use UI thread
let avatarCircle = UIImage.roundedRectImageFromImage
(avatarSquare, imageSize: avatarSquare.size,
cornerRadius: avatarSquare.size.width / 2)
cell.imageView?.image = avatarCircle
}
}
}
}
}
GitHubDemo

BONUS (GCD)


Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
GitHubMembersTableViewController.swift
tableView(_:cellForRowAtIndexPath:)
cell.textLabel?.text = member.login
if let url = NSURL(string: member.avatarUrl) {
dispatch_async(dispatch_get_global_queue(
rawValue), 0)) { // Dispatch on this t
if let data = NSData(contentsOfURL: ur
if let avatarSquare = UIImage(data: data) {
dispatch_async(dispatch_get_main_queue()) { // When we
finally want to display the image use UI thread
let avatarCircle = UIImage.roundedRectImageFromImage
(avatarSquare, imageSize: avatarSquare.size,
cornerRadius: avatarSquare.size.width / 2)
cell.imageView?.image = avatarCircle
A different queue
GitHubDemo

BONUS (GCD)


Running GitHubDemo on iPhone 6
GitHubDemo
GitHubDemo
GitHubMembersTableViewController.swift
tableView(_:cellForRowAtIndexPath:)
cell.textLabel?.text = member.login
if let url = NSURL(string: member.avatarUrl) {
dispatch_async(dispatch_get_global_queue(
rawValue), 0)) { // Dispatch on this t
if let data = NSData(contentsOfURL: ur
if let avatarSquare = UIImage(data: data)
dispatch_async(dispatch_get_main_qu
finally want to display the image
let avatarCircle = UIImage.rounde
(avatarSquare, imageSize: avatarSquare.size,
cornerRadius: avatarSquare.size.width / 2)
cell.imageView?.image = avatarCircle
A different queue
The main queue
GitHubDemo

NEXT TIME

- Revisit auto-layouts to see some other examples

- How to persist data/edit data in table views