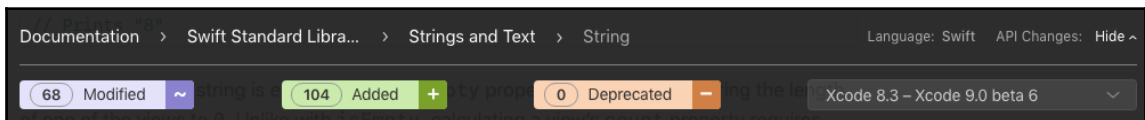
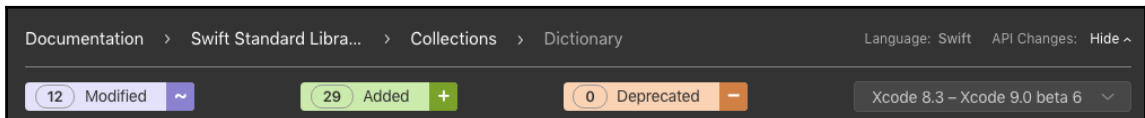
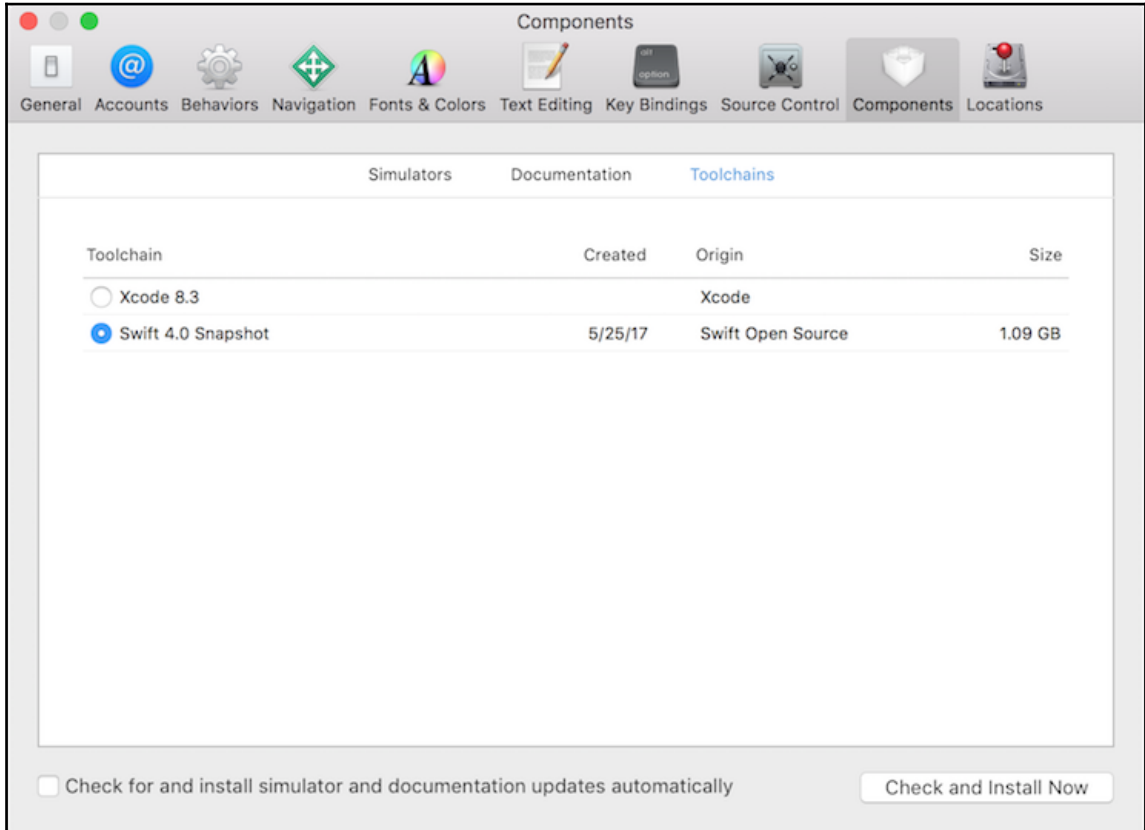
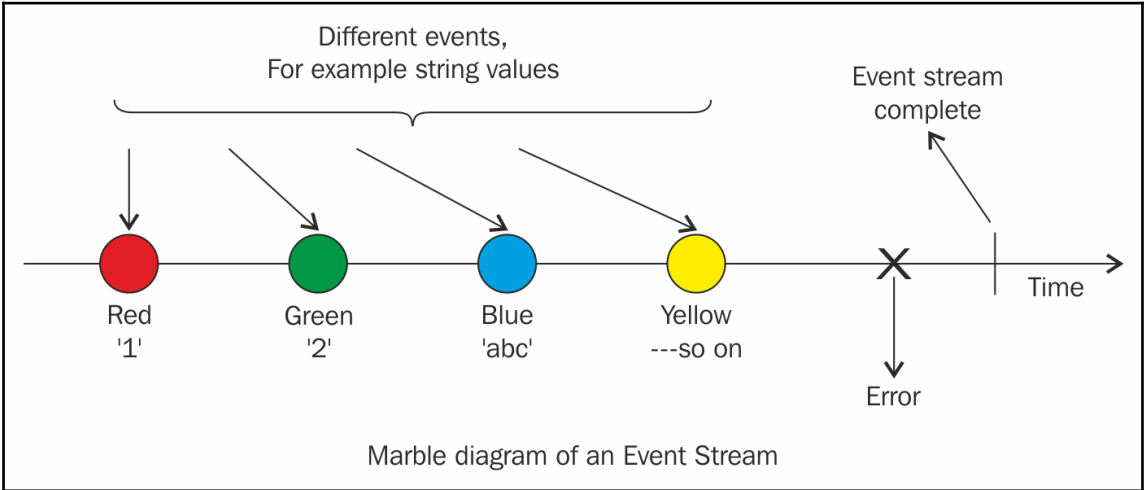


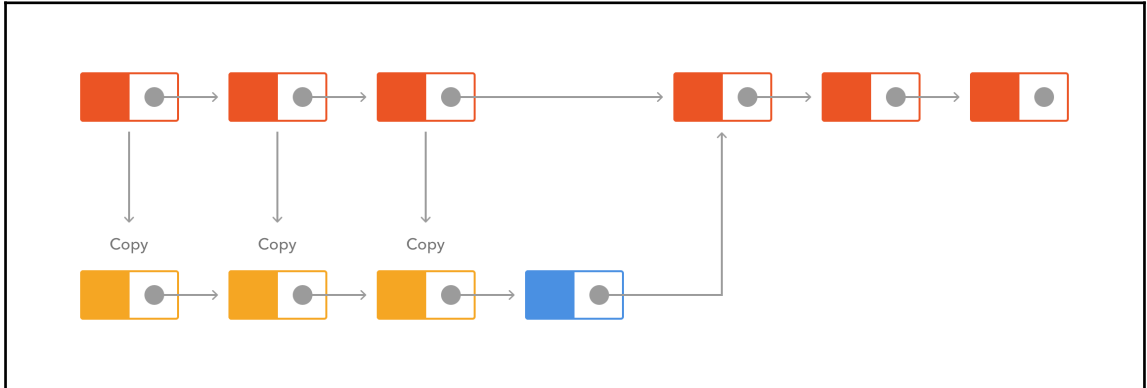
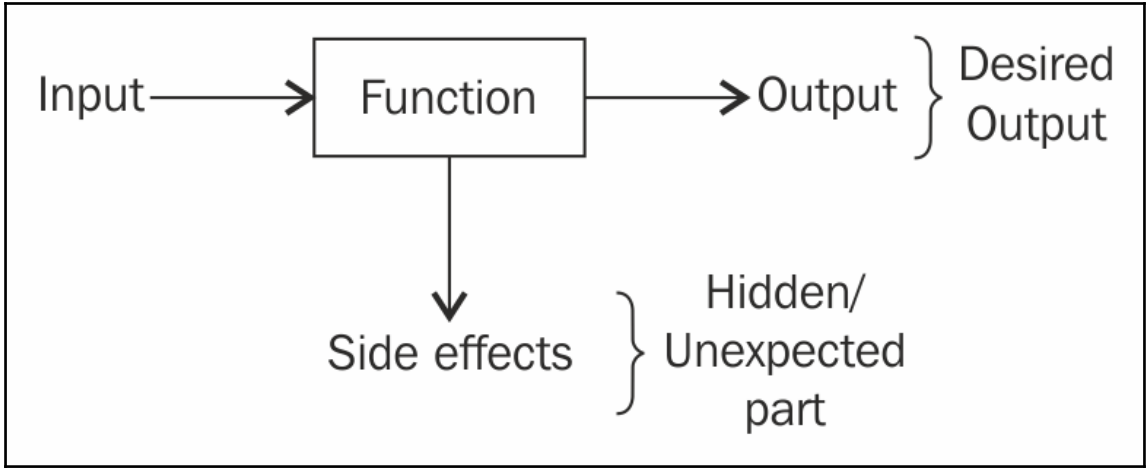
Chapter 1: Migrating from Swift 3 to Swift 4

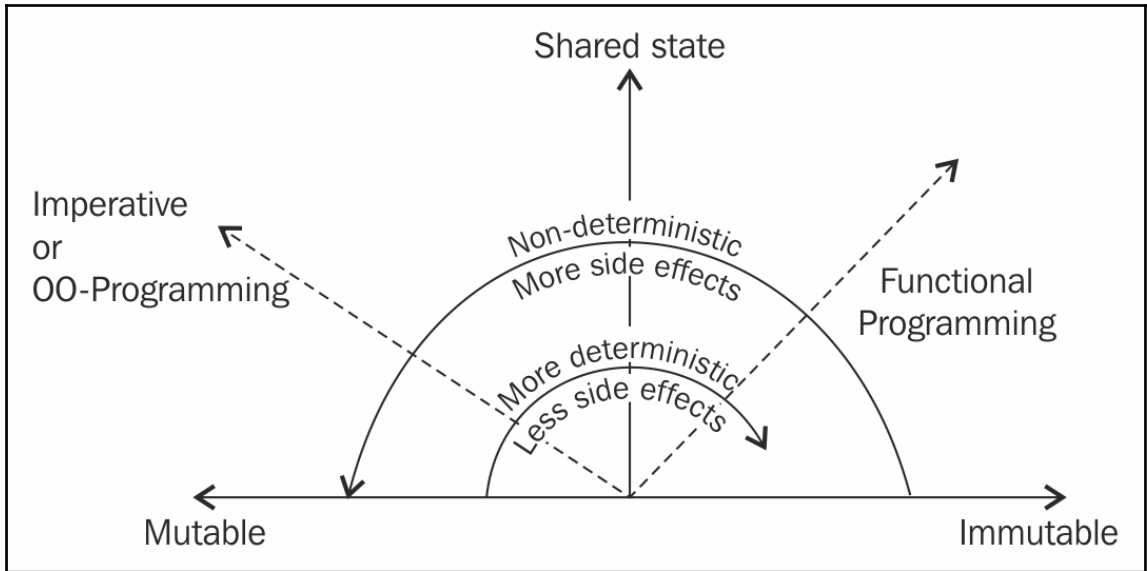


Chapter 2: FRP Fundamentals, Terminology, and Basic Building Blocks



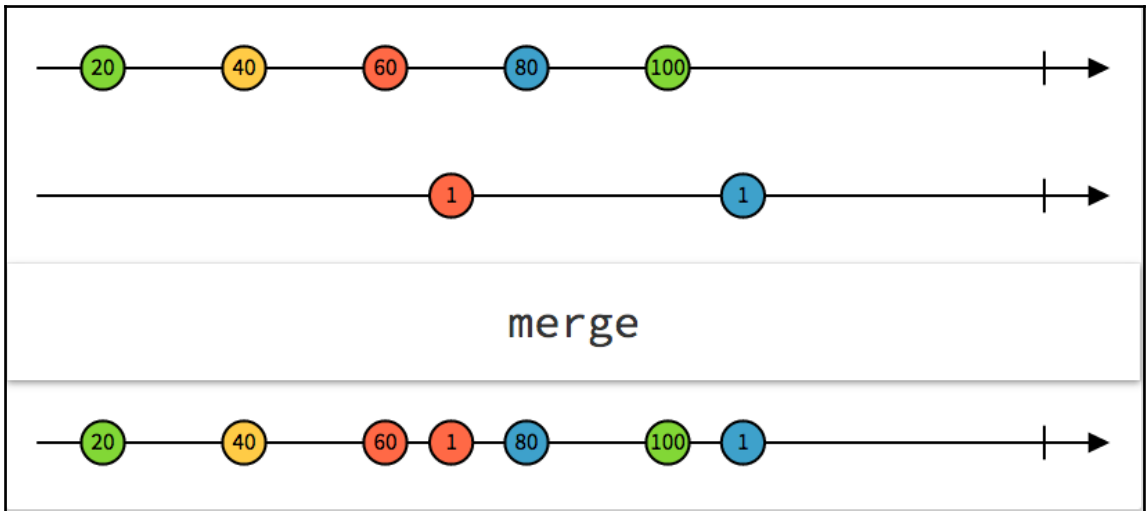
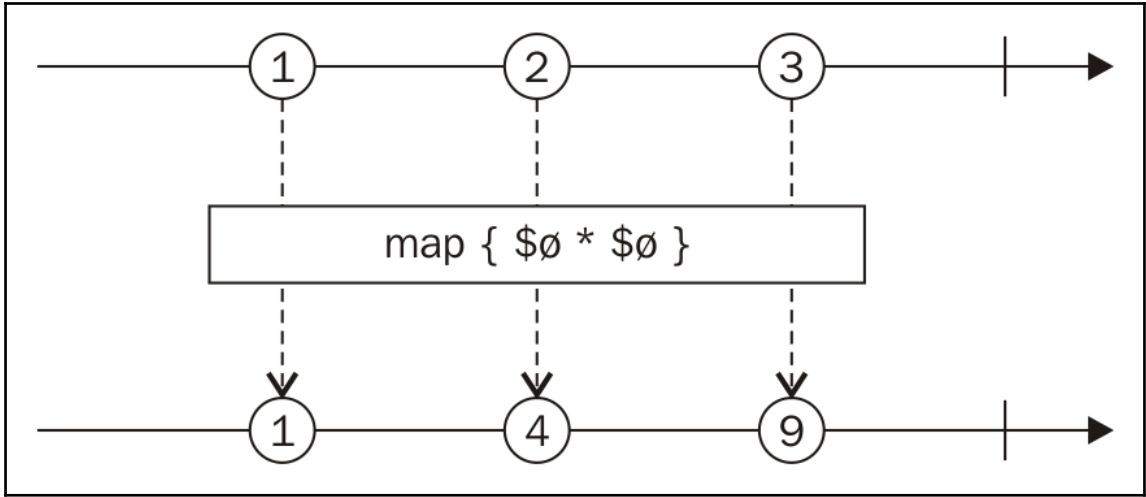
| USER ACTION | BACKGROUND ACTION | CAR STATE |
|-------------------|---------------------------------------|-----------|
| INSERT KEY | SPARK PLUGS ignite fuel | STARTED |
| ROTATE CLOCKWISE | Engine Starts | |
| PRESS CLUTCH | SHAFTS SPIN | MOVING |
| SHIFT GEAR | TYRES ROTATE | |
| PRESS ACCELERATOR | | |
| PRESS BRAKES | SHAFT STOPS SPINNING | STOPPED |
| | TYRES STOP ROTATING | |
| | ENGINE BREAKDOWN OR OVERHEATING | BREAKDOWN |

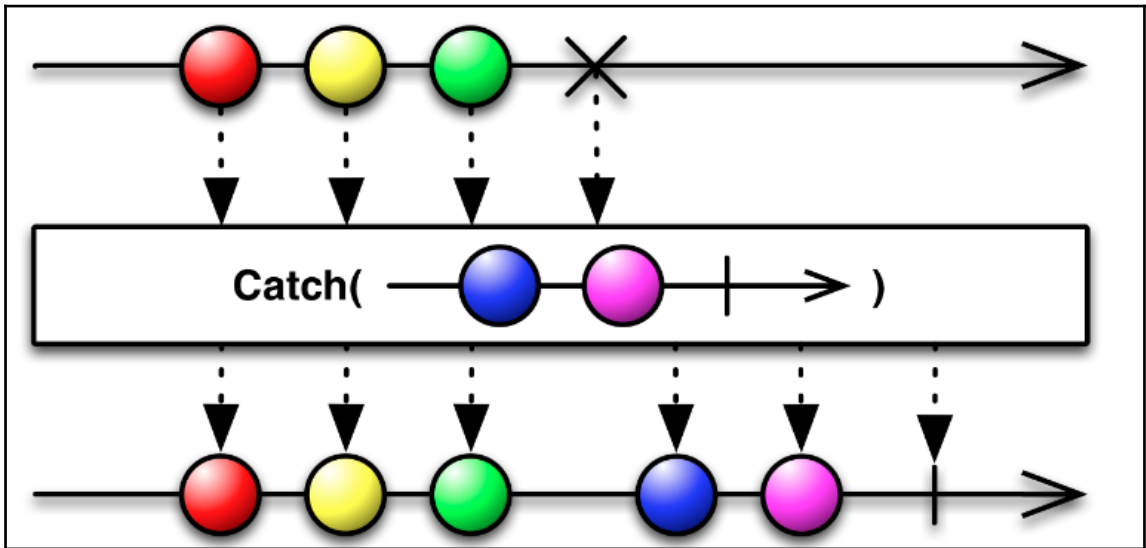
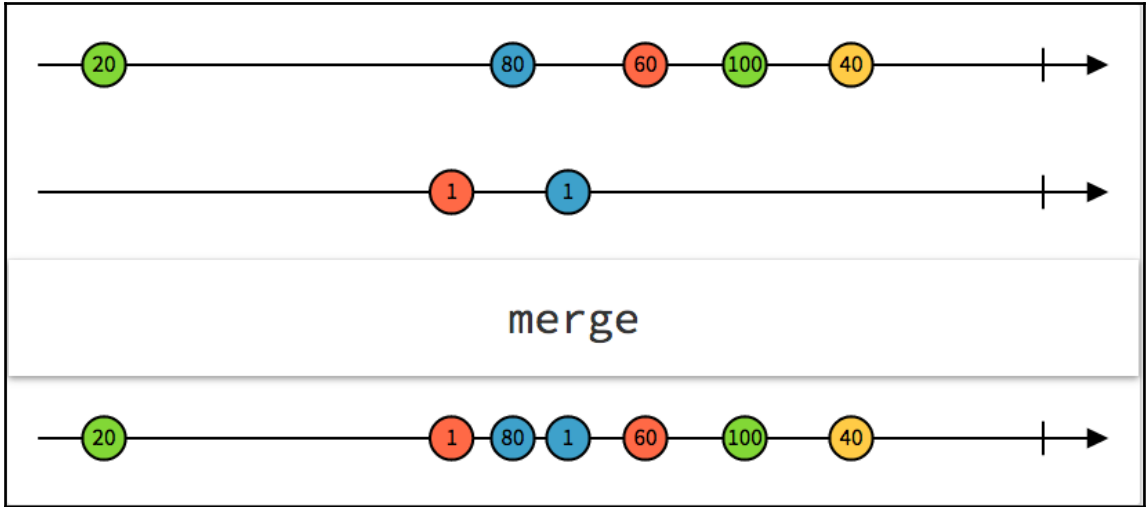


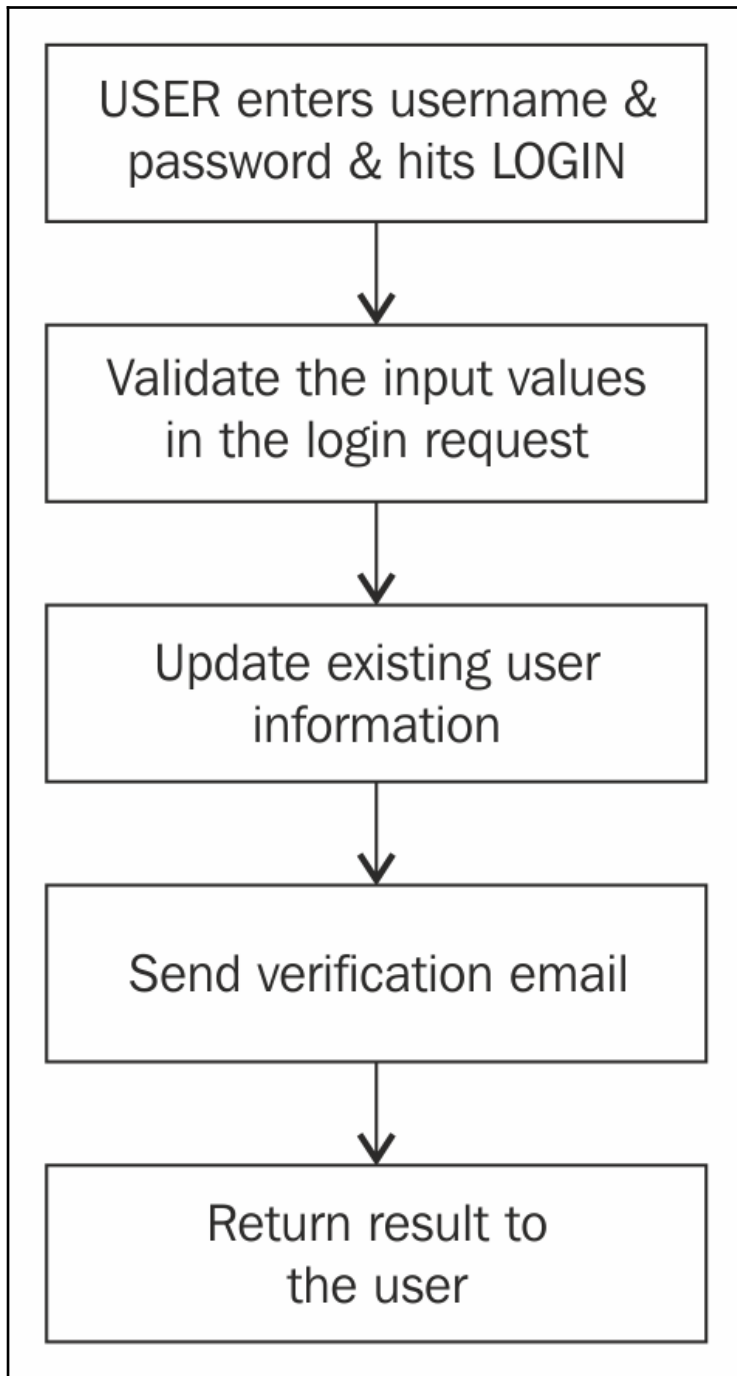


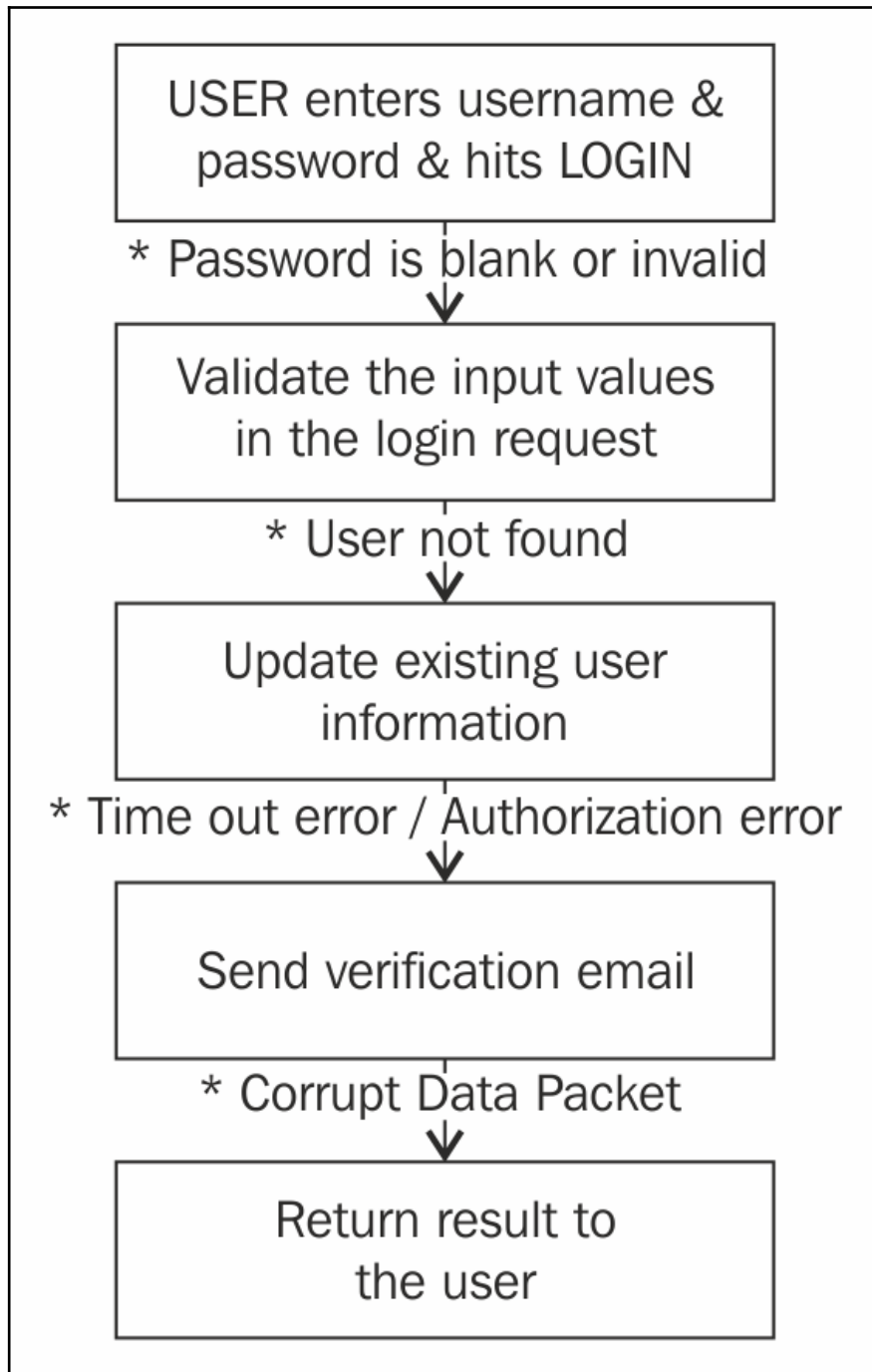
do re mi ...

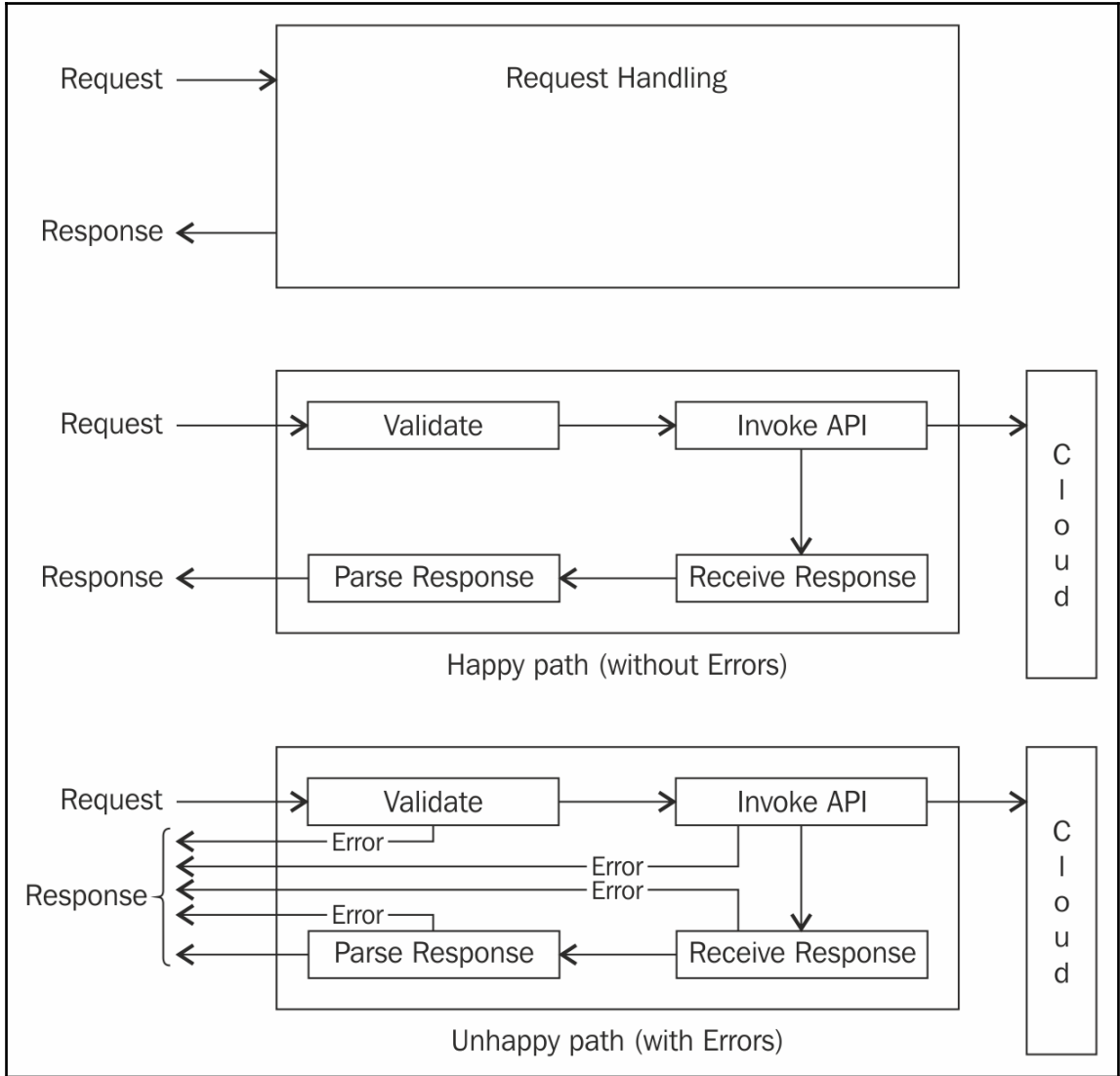
Everything is a Sequence

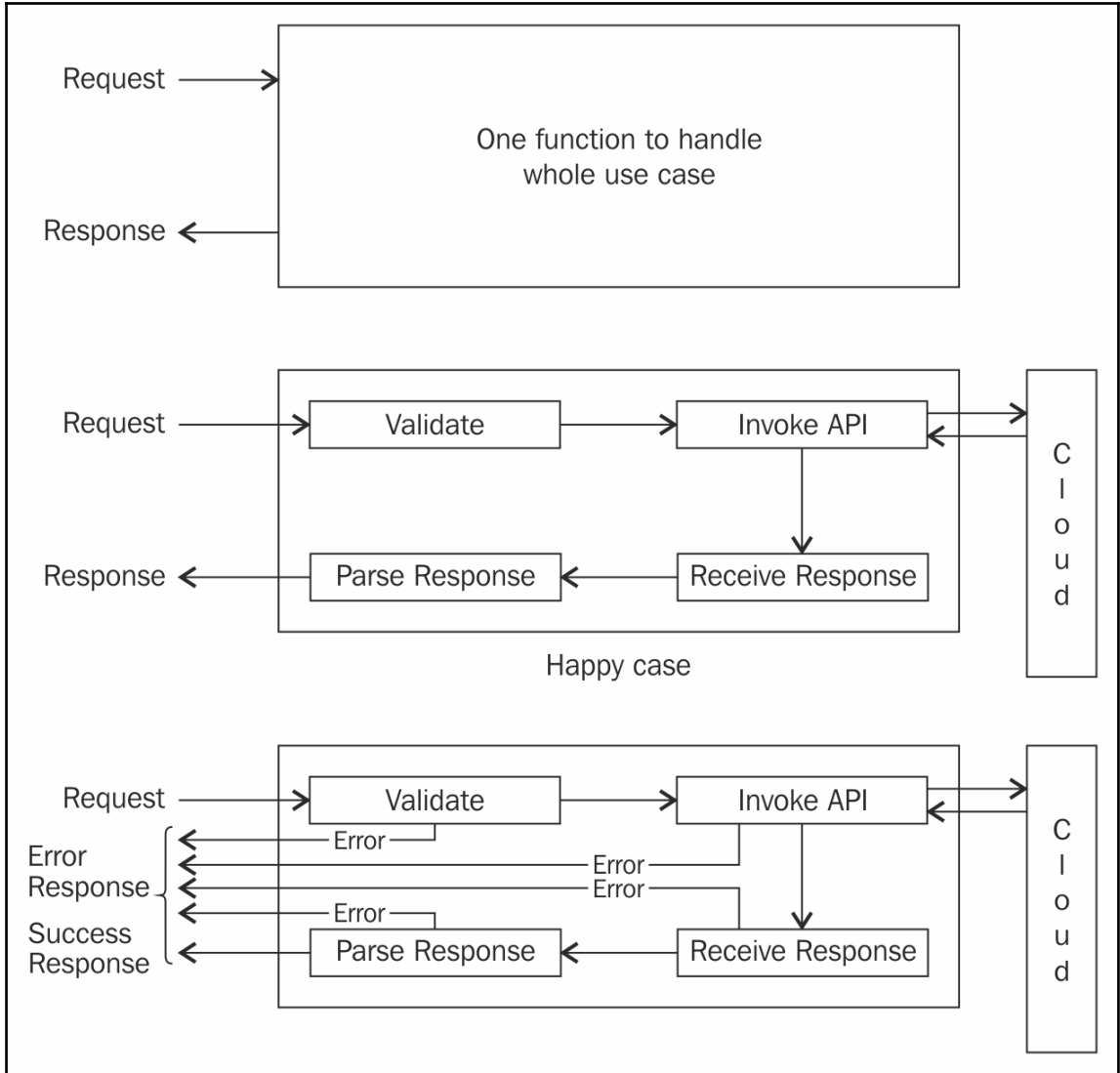


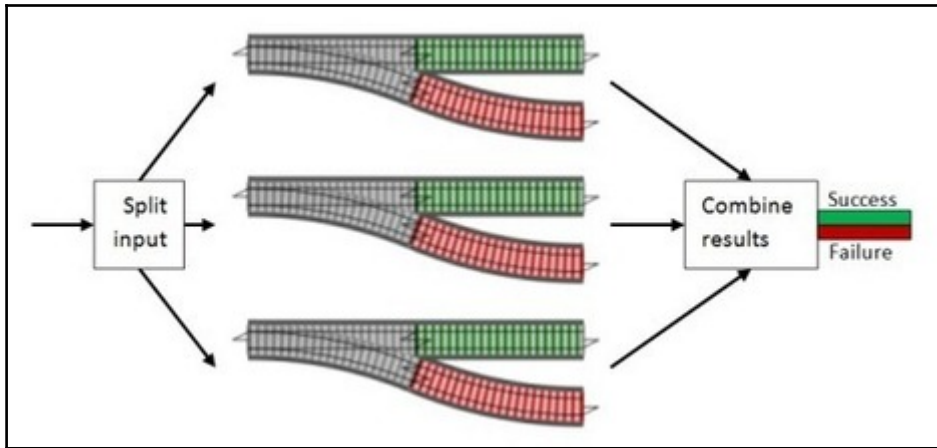




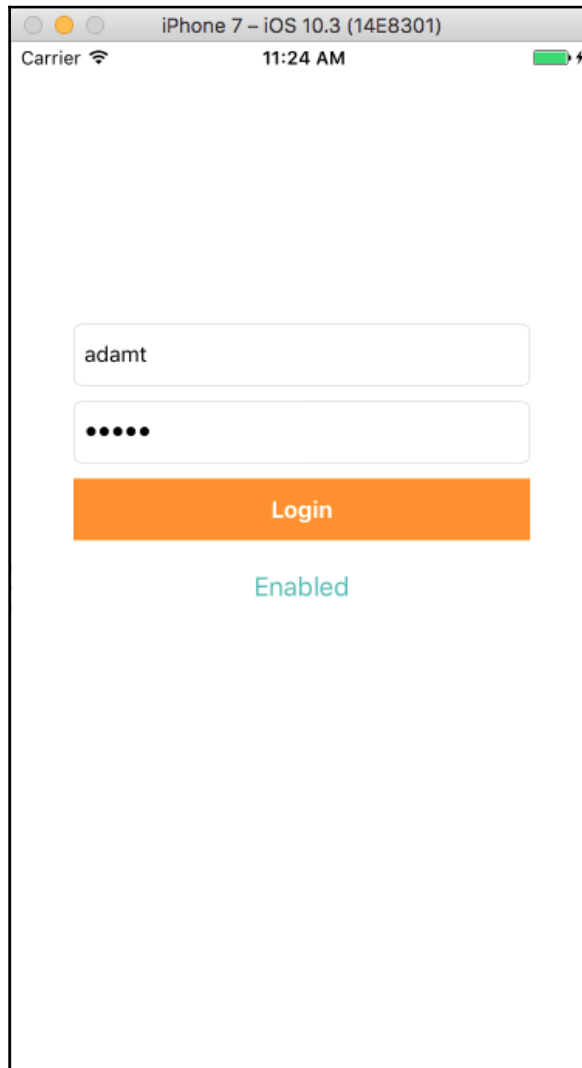


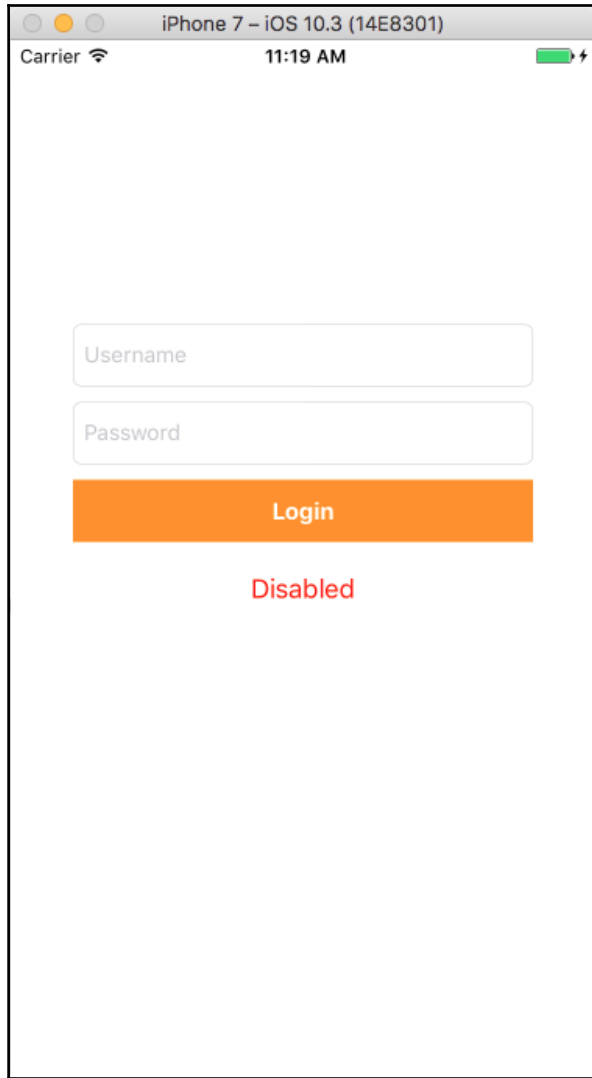






Chapter 3: Set up RxSwift and Convert a Basic Login App to its RxSwift Counterpart





Choose options for your new project:

Product Name: RxSwiftLoginView

Team: ~~Sonar Technologies Australia Pty Ltd~~

Organization Name: Navdeep

Organization Identifier: com.Navdeep

Bundle Identifier: com.Navdeep.RxSwiftLoginView

Language: Swift

Devices: iPhone

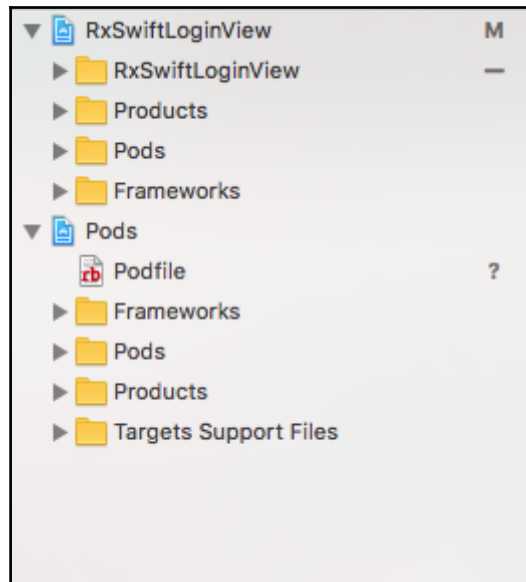
Use Core Data

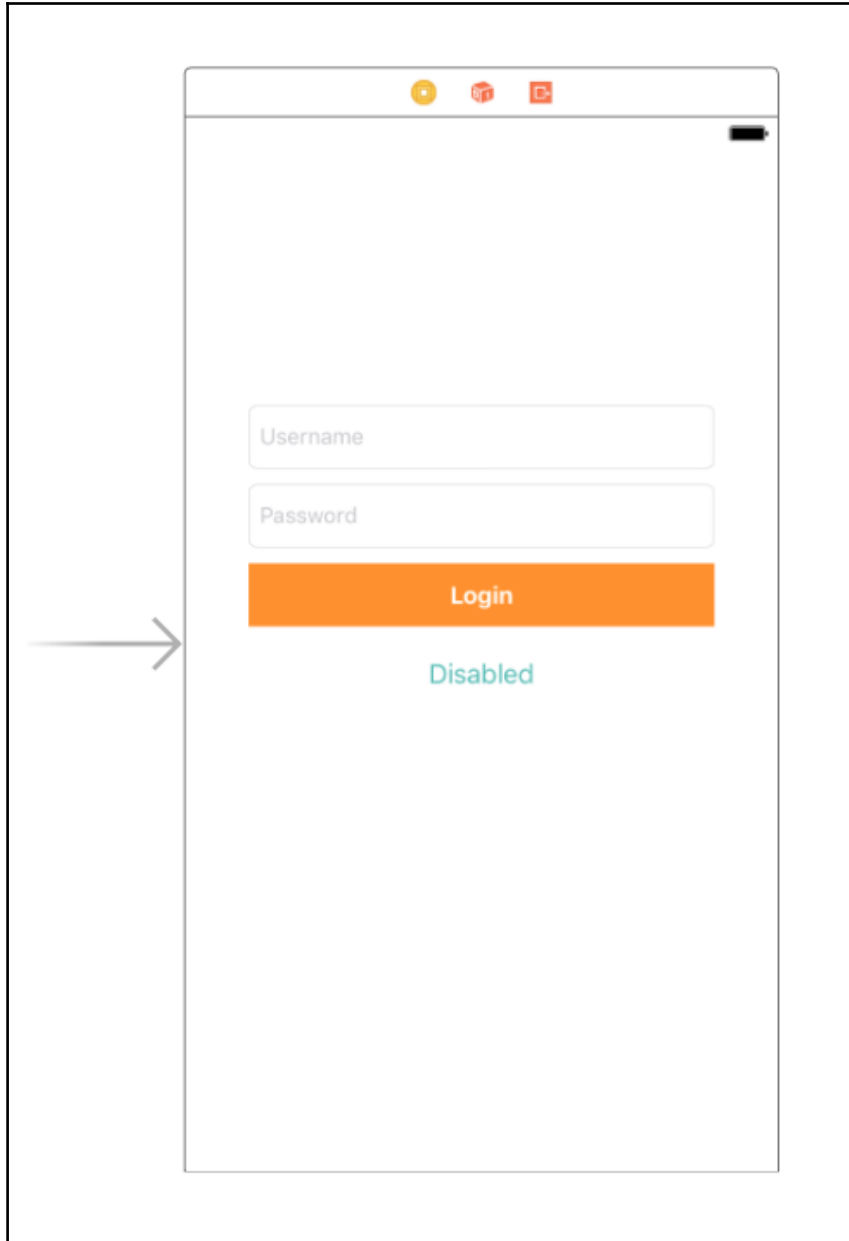
Include Unit Tests

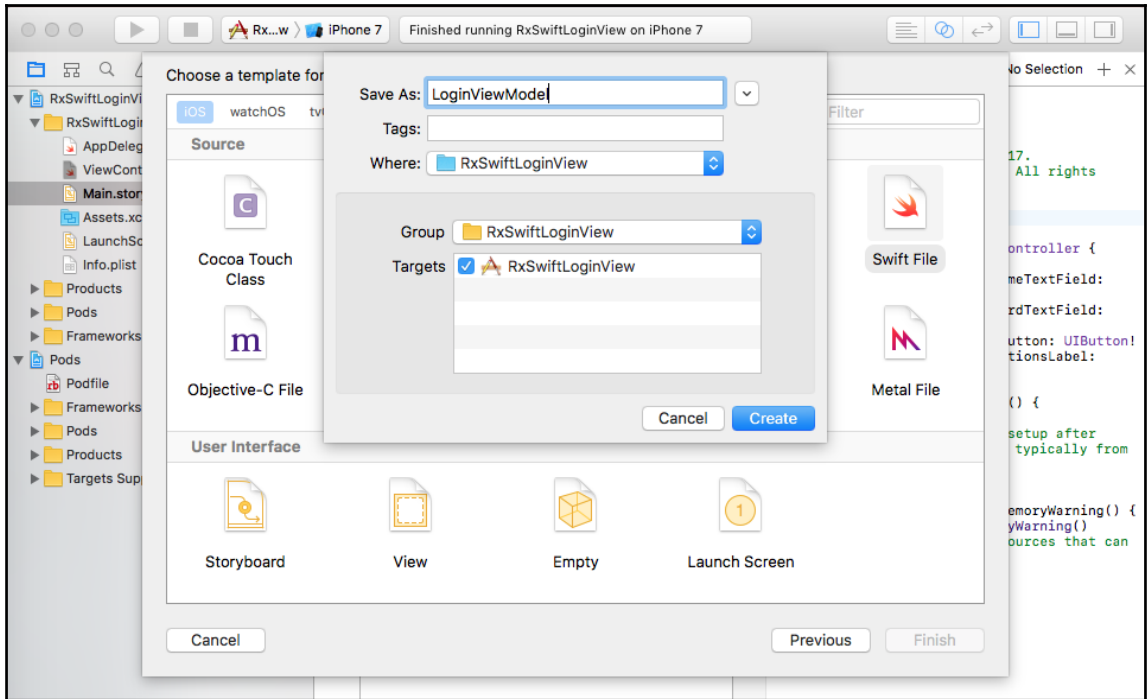
Include UI Tests

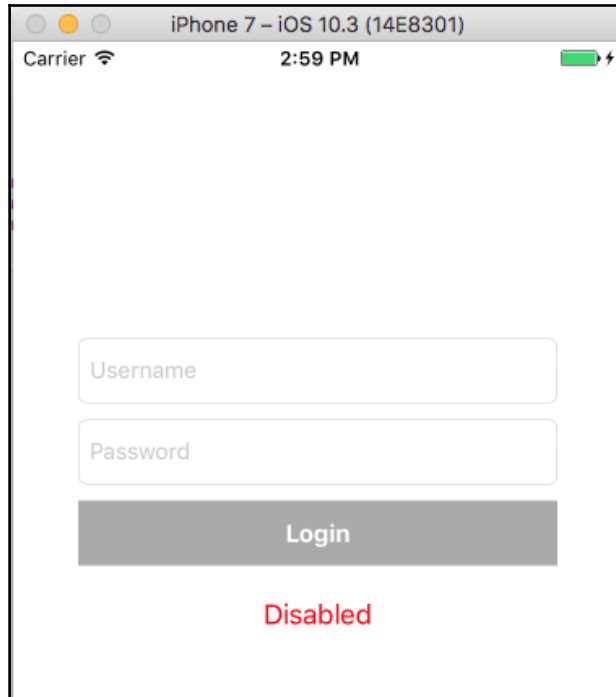
Cancel Previous Next

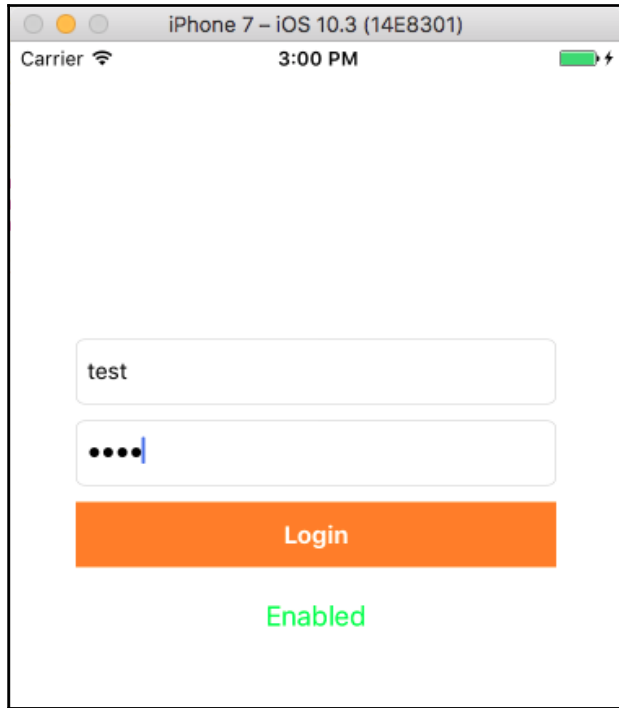
```
RxSwiftLoginView — -bash — 80x24
Last login: Wed Sep 27 14:32:46 on ttys002
Sonar-MBP:~ sonar$ cd /Users/sonar/Documents/XcodeWorkspace/test/RxSwiftLoginView
[Sonar-MBP:RxSwiftLoginView sonar$ pod init
Sonar-MBP:RxSwiftLoginView sonar$ █
```



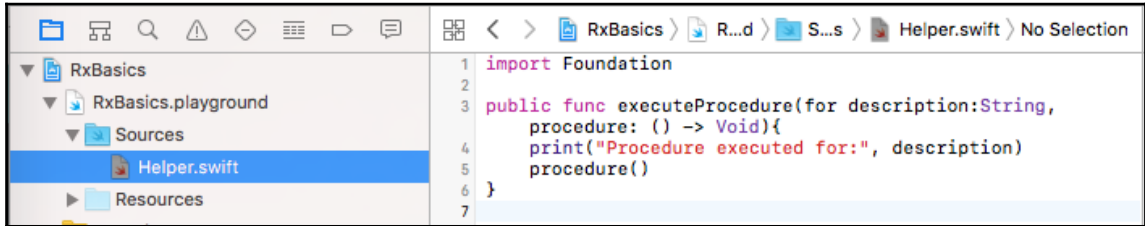




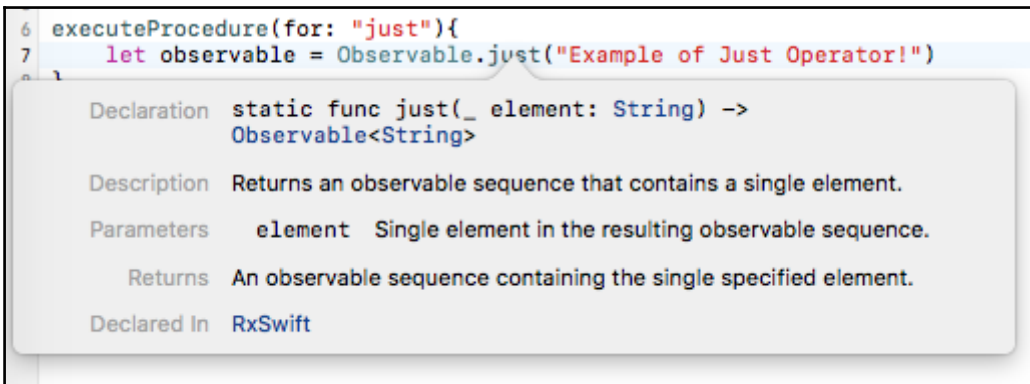




Chapter 4: When to Become Reactive?



```
1 import Foundation
2
3 public func executeProcedure(for description:String,
4                             procedure: () -> Void){
5     print("Procedure executed for:", description)
6     procedure()
7 }
```



```
6 executeProcedure(for: "just"){
7     let observable = Observable.just("Example of Just Operator!")
8 }
```

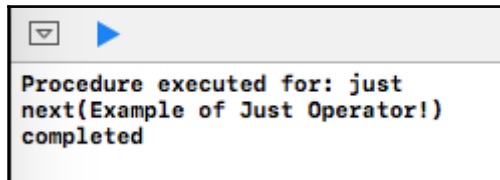
Declaration `static func just(_ element: String) -> Observable<String>`

Description Returns an observable sequence that contains a single element.

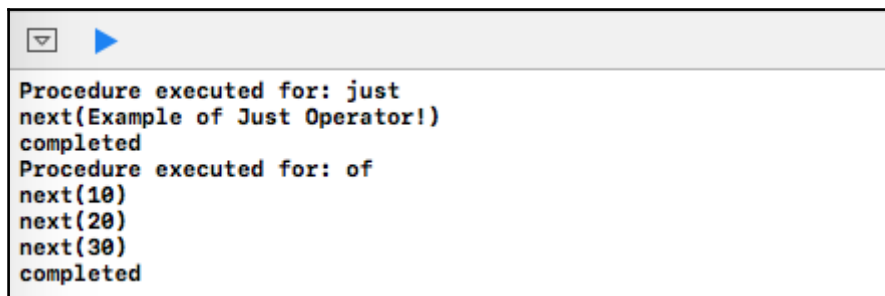
Parameters `element` Single element in the resulting observable sequence.

Returns An observable sequence containing the single specified element.

Declared In RxSwift



```
Procedure executed for: just
next(Example of Just Operator!)
completed
```



```
Procedure executed for: just
next(Example of Just Operator!)
completed
Procedure executed for: of
next(10)
next(20)
next(30)
completed
```

```
▼ ▶
Procedure executed for: from
next(10)
next(20)
next(30)
completed
```

```
/**
 * Subscribes an event handler to an observable sequence.
 *
 * - parameter on: Action to invoke for each event in the observable sequence.
 * - returns: Subscription object used to unsubscribe from the observable sequence.
 */
public func subscribe(_ on: @escaping (RxSwift.Event<Self.E>) -> Swift.Void) -> Disposable
```

```
/// Represents a disposable resource.
public protocol Disposable {
    /// Dispose resource.
    func dispose()
}
```

```
25 executeProcedure(for: "from"){
26     let subscribed = Observable.from([10, 20,30])
27     .subscribe{
    Declaration let subscribed: Disposable
    Declared In RxBasics.playground
```

```
▼ ▶
Procedure executed for: from
10
20
30
```

```
▼ ▶  
Procedure executed for: from  
10  
20  
30  
1  
2  
3  
Completed the events  
Sequence terminated hence Disposed
```

```
▼ ▶  
Procedure executed for: error  
defaultError
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
next(Second Event)
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
completed
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
error(defaultError)
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
next(Second Event)
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
next(Second Event)  
next(I am New!)  
New Subscription I am New!
```

```
▼ ▶  
Procedure executed for: PublishSubject  
next(First Event)  
next(Second Event)  
next(I am New!)  
New Subscription I am New!  
next(Fourth Event)
```

```
▼ ▶  
Procedure executed for: BehaviorSubject  
Line number is 90 and value is Test
```

```
▼ ▶  
Procedure executed for: BehaviorSubject  
Line number is 90 and value is Test  
Line number is 90 and value is Second Event
```

```
▼ ▶  
Procedure executed for: BehaviorSubject  
Line number is 90 and value is Test  
Line number is 90 and value is Second Event  
Line number is 95 and value is Second Event
```

```
▼ ▶  
Procedure executed for: ReplaySubject  
First  
Second  
Third
```





```
▼ ▶  
Procedure executed for: ReplaySubject  
Second  
Third  
Fourth
```

```
▼ ▶  
Procedure executed for: ReplaySubject  
Second  
Third  
Fourth  
Fifth
```

```
▼ ▶  
Procedure executed for: ReplaySubject  
Second  
Third  
Fourth  
Fifth  
New Subscription: Third  
New Subscription: Fourth  
New Subscription: Fifth
```

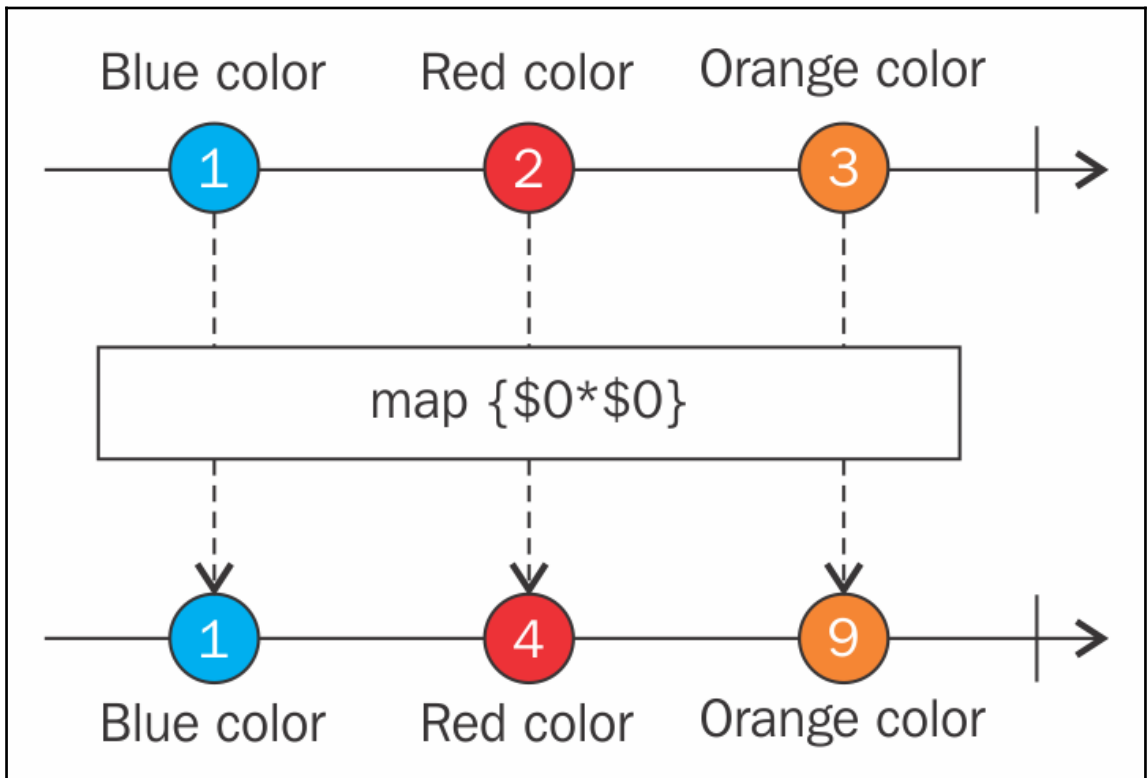
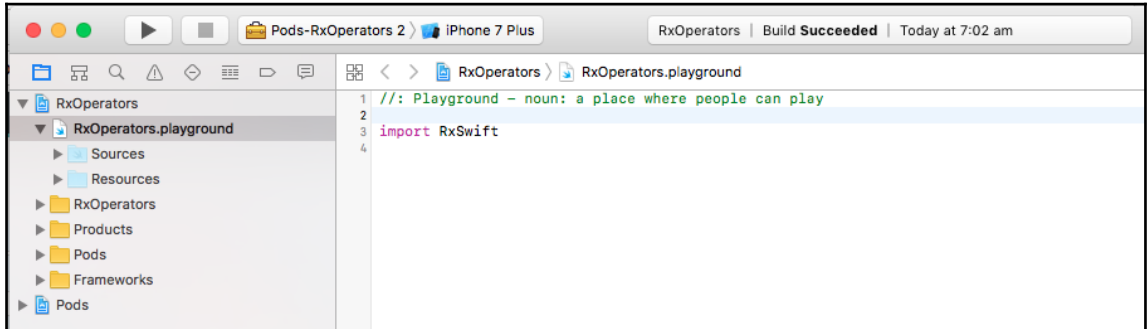
```
▼ ▶  
Procedure executed for: Variable  
next(1)  
completed
```

```
executeProcedure(for: "Variable") {  
    let disposeBag = DisposeBag()  
  
    let variable = Variable(1)  
    variable.asObservable()  
        .subscribe{  
            print($0)  
        }  
        .disposed(by: disposeBag)  
    variable.value  
  
    1  
  
}
```

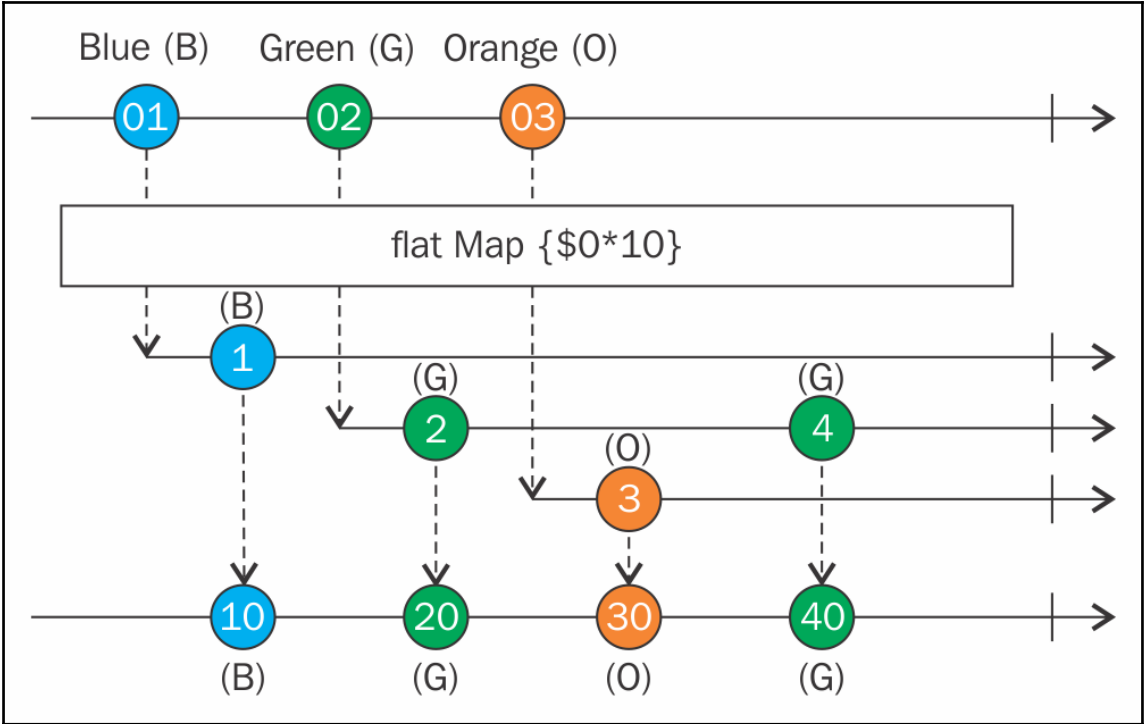
 

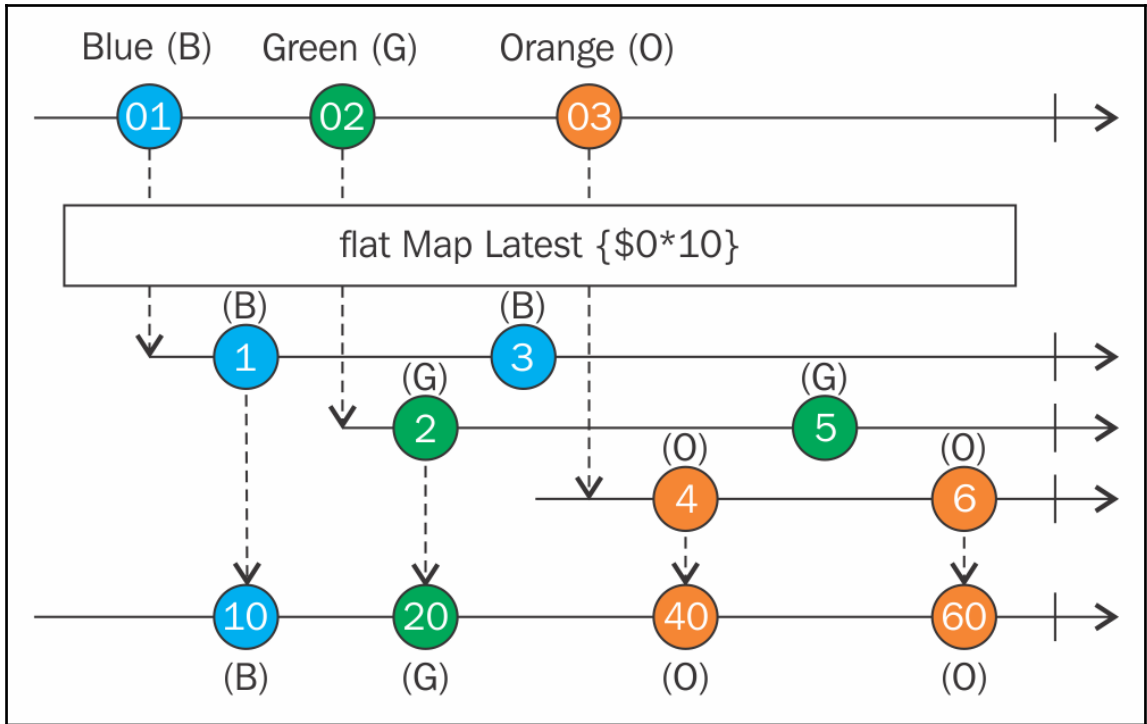
Procedure executed for: Variable
next(1)
next(2)
completed

Chapter 5: Filter, Transform, and Simplify



```
Procedure executed for: map  
100  
400  
900
```





```

25
26 var current*Player = Variable(alex)

```

Declaration var currentPlayer: Variable<GamePlayer>
 Declared In RxOperators.playground

Procedure executed for: flatMap and flatMapLatest
 70

Procedure executed for: flatMap and flatMapLatest
 70
 90

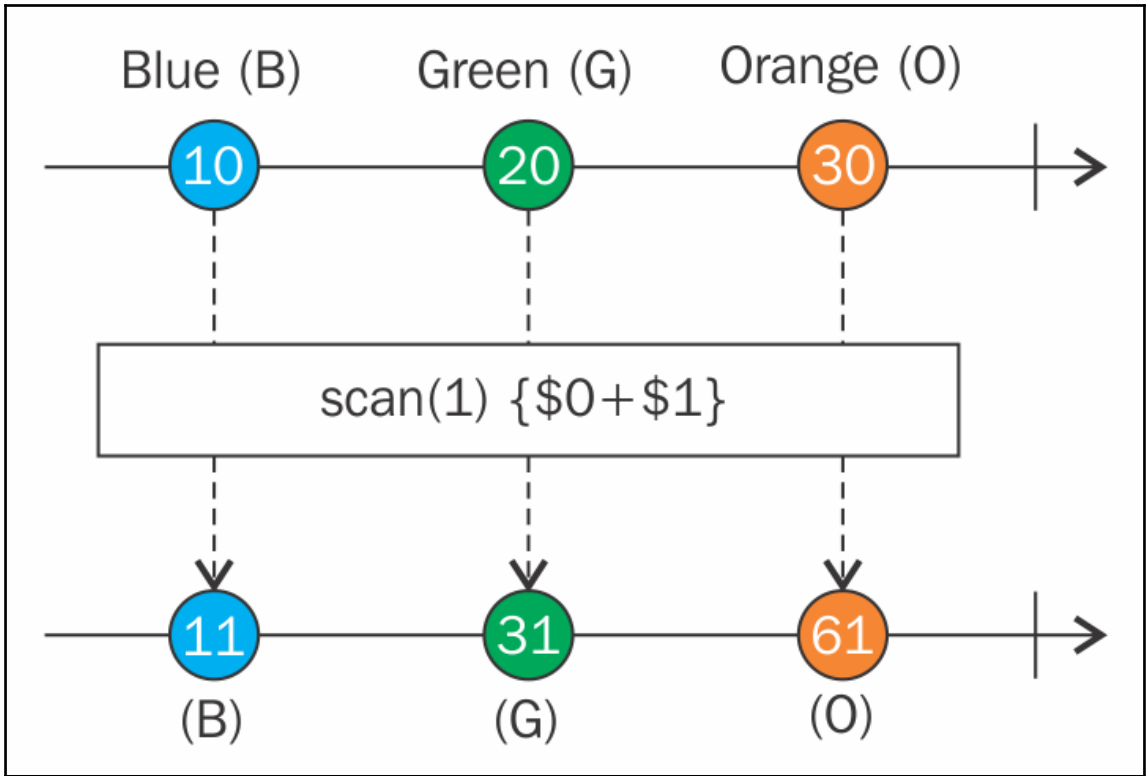
```
Procedure executed for: flatMap and flatMapLatest
70
90
95
```

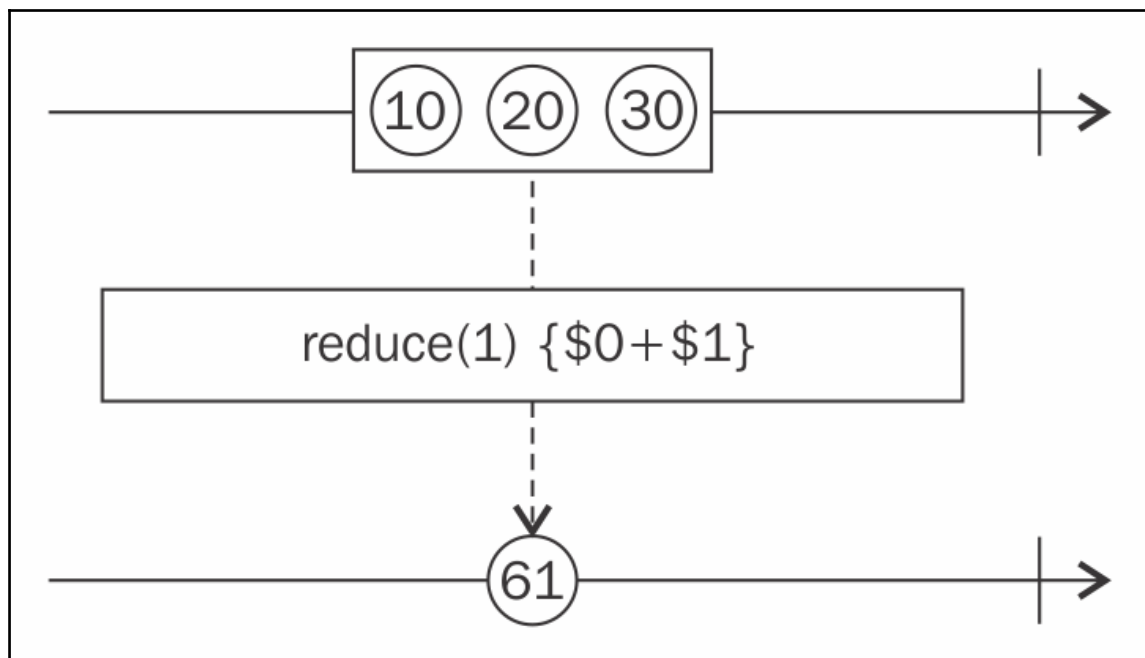
```
Procedure executed for: flatMap and flatMapLatest
70
90
95
85
```

```
Procedure executed for: flatMap and flatMapLatest
70
90
95
85
96
```

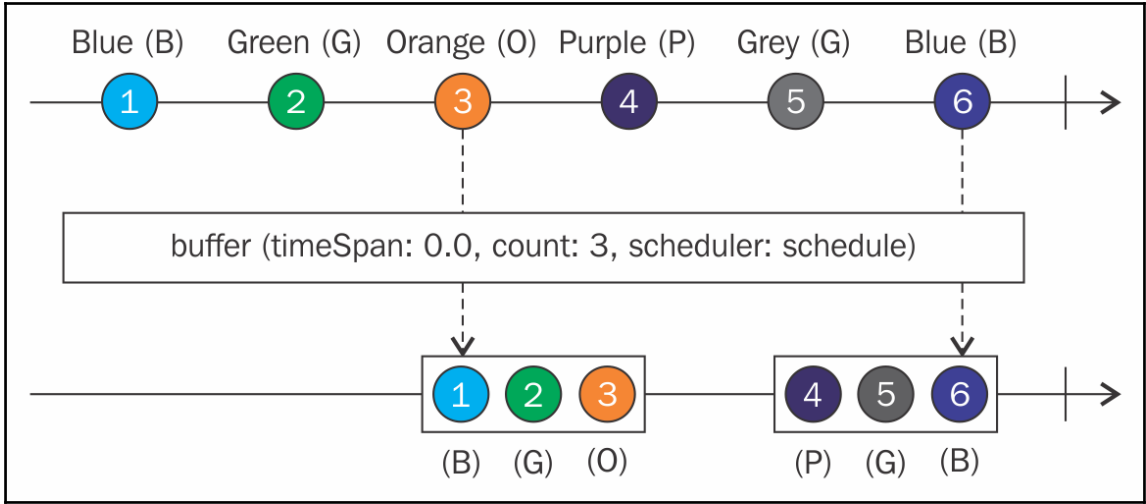
```
Procedure executed for: flatMap and flatMapLatest
70
90
95
85
96
```

```
Procedure executed for: flatMap and flatMapLatest
70
90
95
85
```





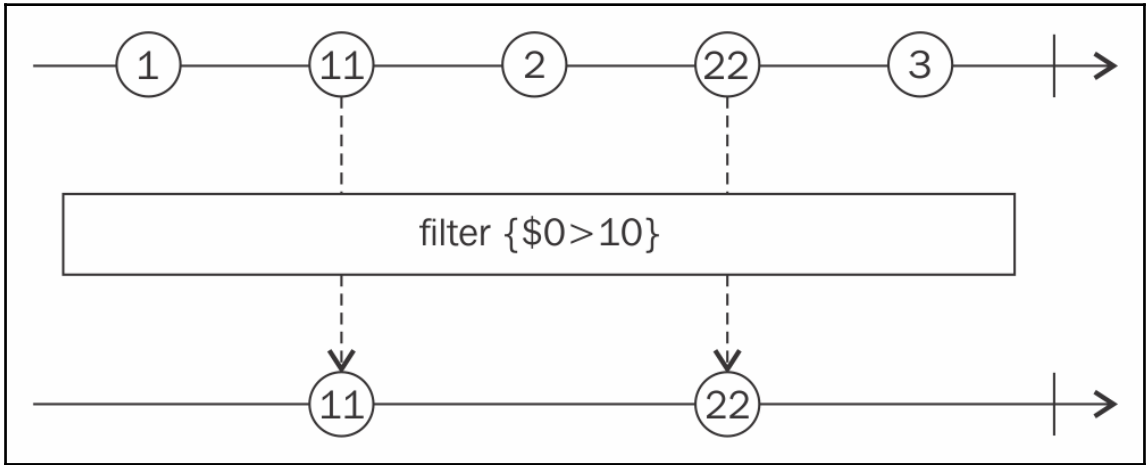
```
▼ ▶  
Procedure executed for: scan and buffer  
441  
428  
378
```

```

▼ ▶
Procedure executed for: scan and buffer
[60, 13, 50] --> 378

```



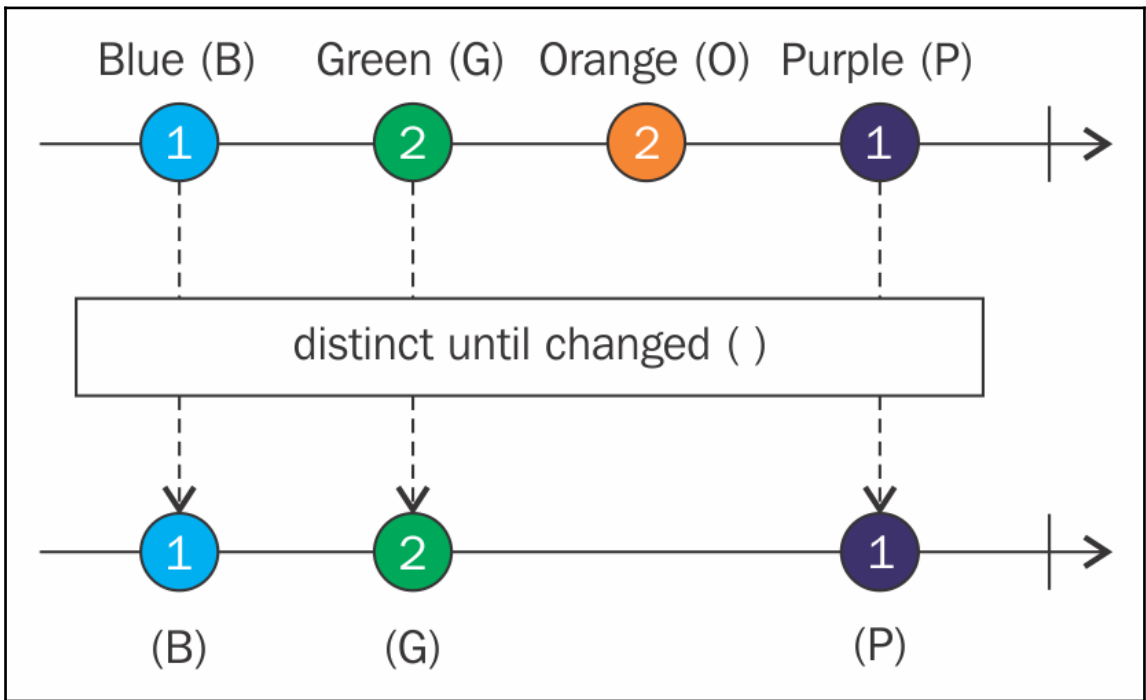
```

▼ ▶
Procedure executed for: filter

```

```
Procedure executed for: filter
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
```

```
Procedure executed for: filter
next([2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97])
completed
```

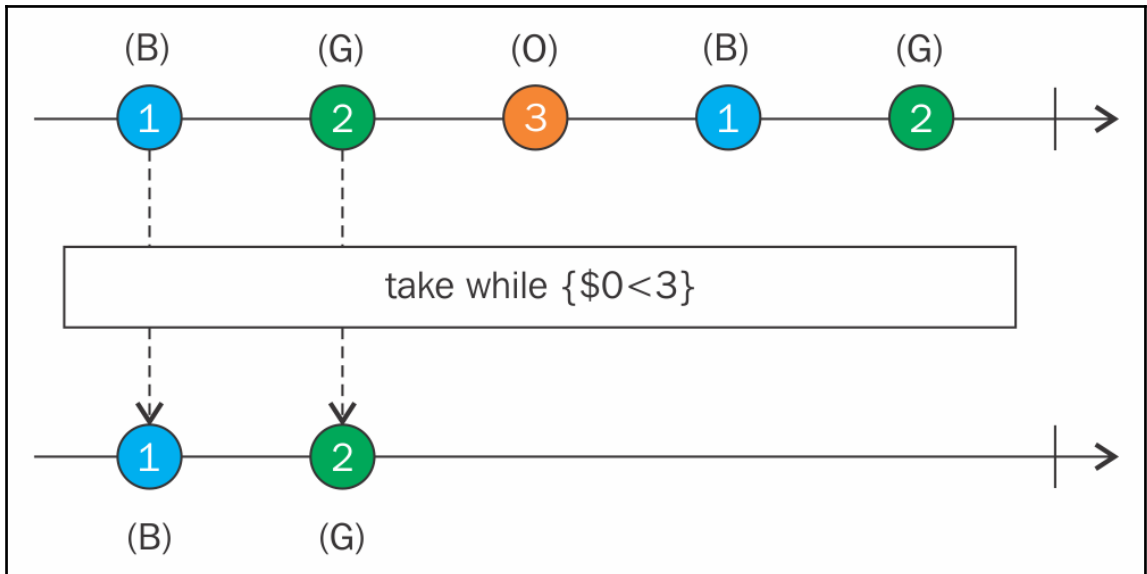


```
Procedure executed for: distinctUntilChanged
next()
completed
```

```
Procedure executed for: distinctUntilChanged
next()
next(tintin)
completed
```



```
Procedure executed for: distinctUntilChanged
next()
next(tintin)
next(noddy)
completed
```

```
Procedure executed for: distinctUntilChanged
next()
next(tintin)
next(noddy)
next(tintin)
completed
```



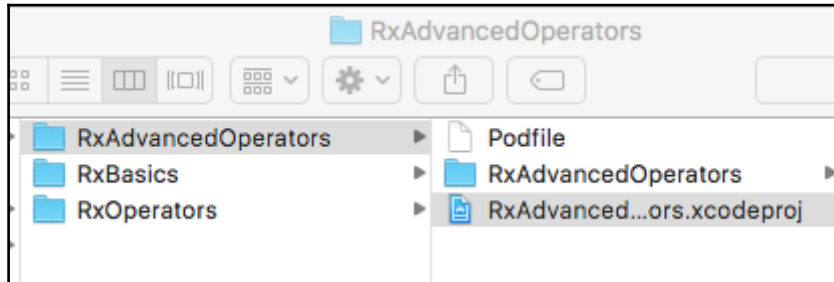
```
116 let integers = Observable.of(10, 20, 30, 40, 30, 20, 10)
117 integers
118 .take|
```

- M Observable<Int> **take**(count: Int)
- M Observable<Int> takeLast(count: Int)
- M Observable<Int> takeUntil(other: ObservableType)
- M Observable<Int> takeWhile(predicate: (Int) throws -> Bool)
- M Observable<Int> take(duration: RxTimeInterval, scheduler: SchedulerType)
- M Observable<Int> takeWhileWithIndex(predicate: (Int, Int) throws -> Bool)
- M ConnectableObservable<SubjectType.E> multicast(makeSubject: () -> SubjectType)

Procedure executed for: takeWhile
10
20
30

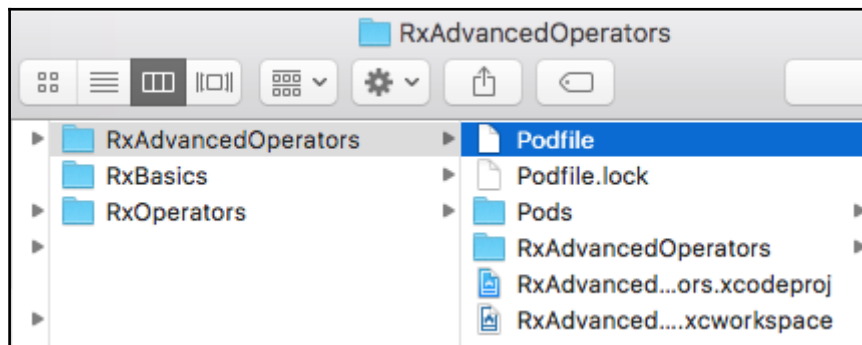
Chapter 6: Reduce by Combining and Filtering and Common Trade Offs

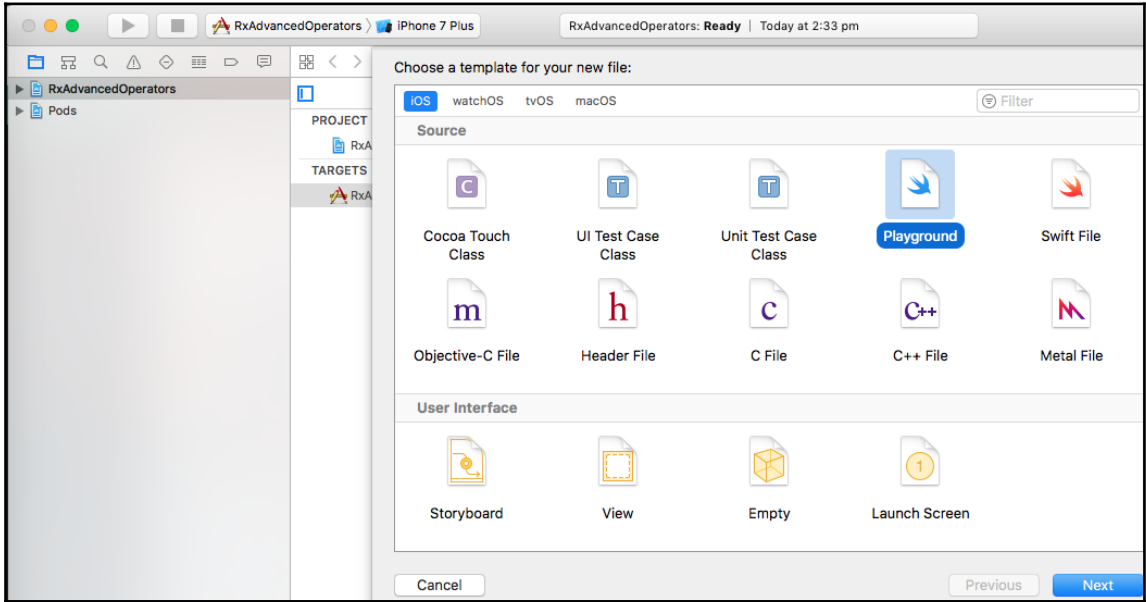
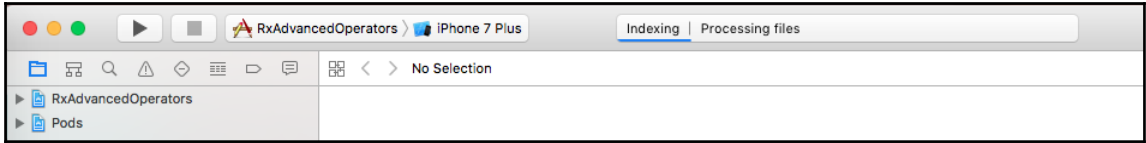


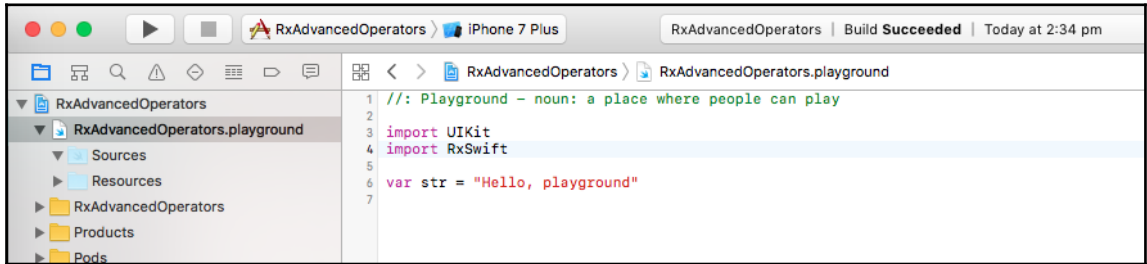
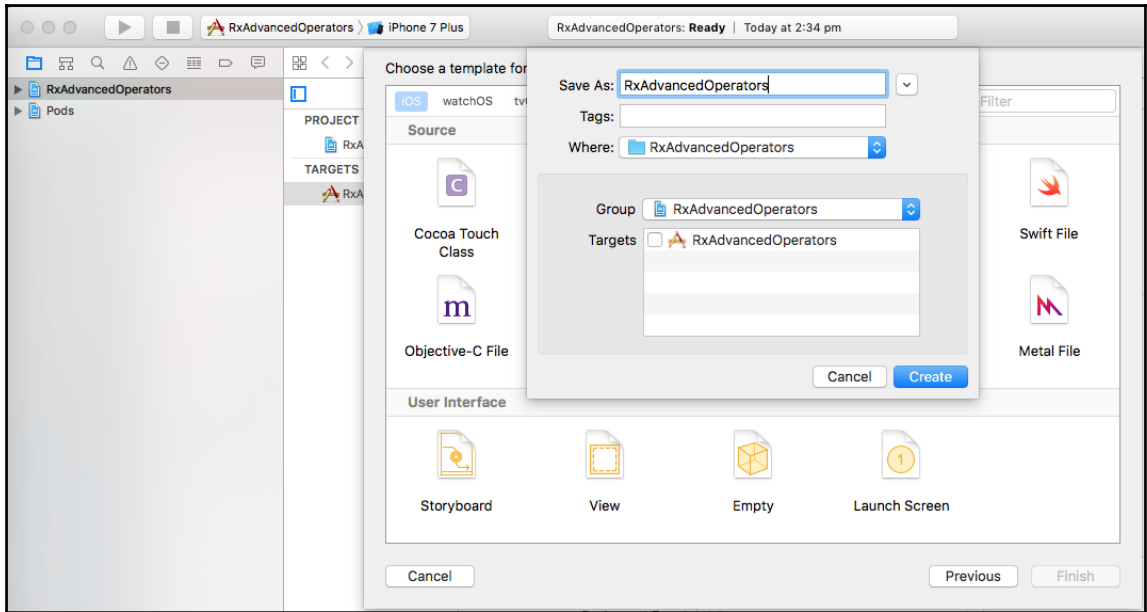
```
Sonar-MBP:RxAdvancedOperators sonar$ pod install
Analyzing dependencies
Downloading dependencies
Installing RxSwift (3.6.1)
Generating Pods project
Integrating client project

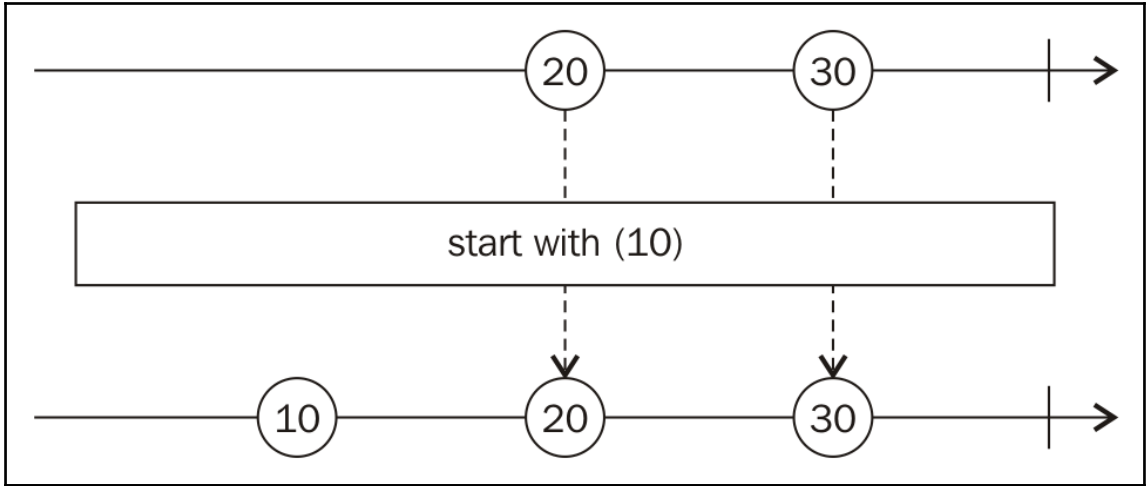
[!] Please close any current Xcode sessions and use `RxAdvancedOperators.xcworkspace` for this project from now on.
Sending stats
Pod installation complete! There is 1 dependency from the Podfile and 1 total pod installed.

[!] Automatically assigning platform ios with version 10.3 on target RxAdvancedOperators because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.
Sonar-MBP:RxAdvancedOperators sonar$
```



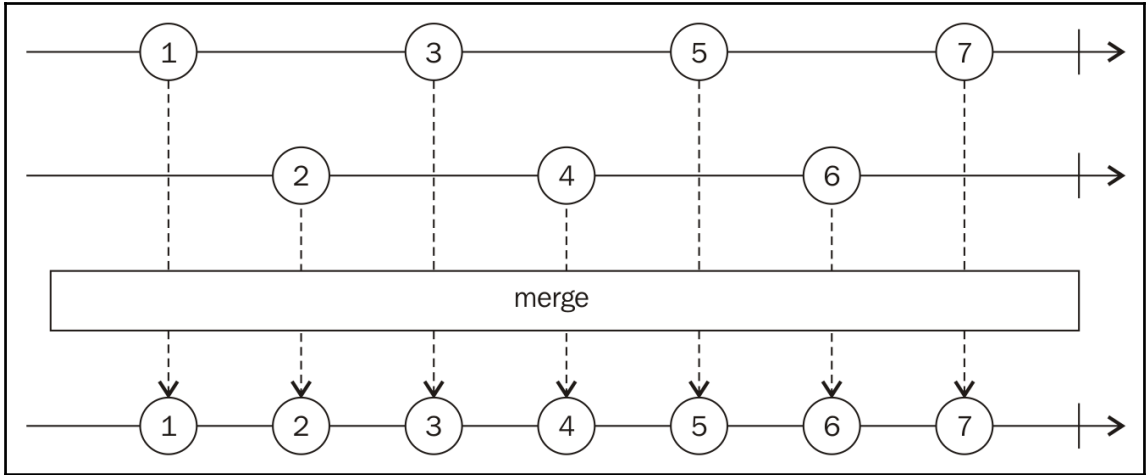






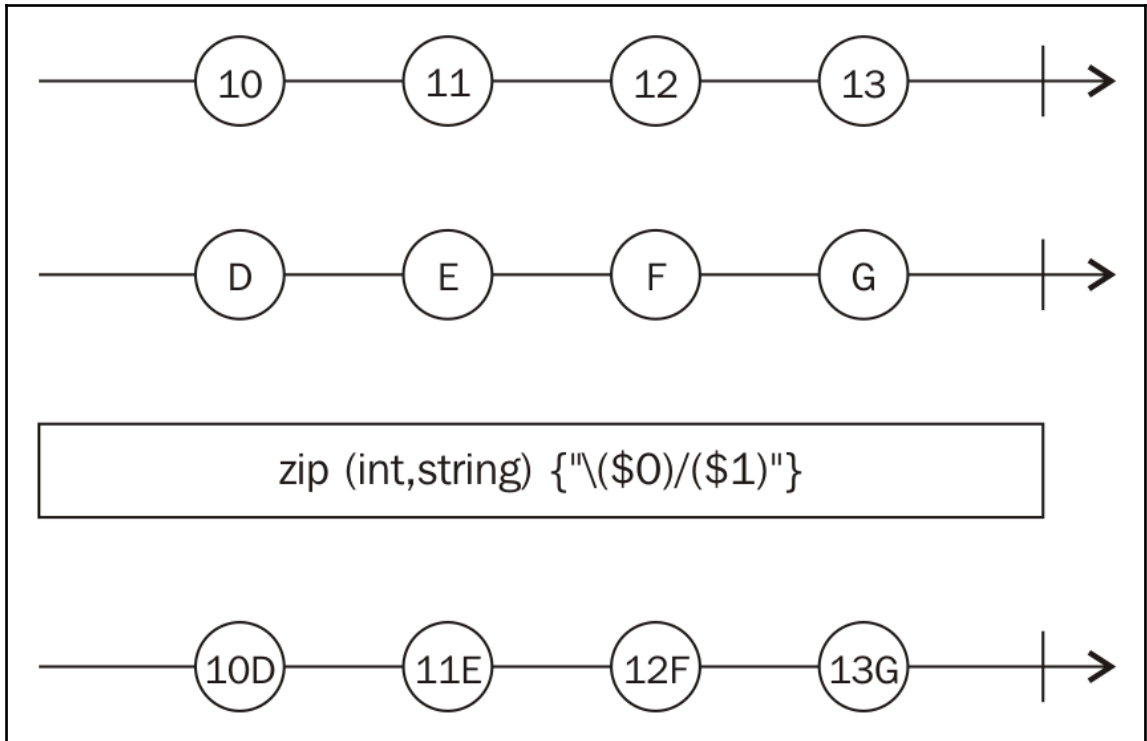
```
1 import Foundation
2
3 public func executeProcedure(for description:String, procedure: () -> Void){
4     print("Procedure executed for:", description)
5     procedure()
6 }
7
```

```
Procedure executed for: startWith
String -2
String -1
String 0
String 1
String 2
String 3
String 4
```

```
▼ ▶  
Procedure executed for: merge  
First Element from Subject 1  
First Element from Subject 2  
First Element from Subject 3  
Second Element from Subject 1  
Second Element from Subject 3  
Second Element from Subject 2
```

```
▼ ▶  
Playground execution failed: error: RxAdvancedOperators.playground:42:24: error: cannot convert value of type 'Int'  
to expected argument type 'String'  
pubSubject3.onNext(3)  
^
```

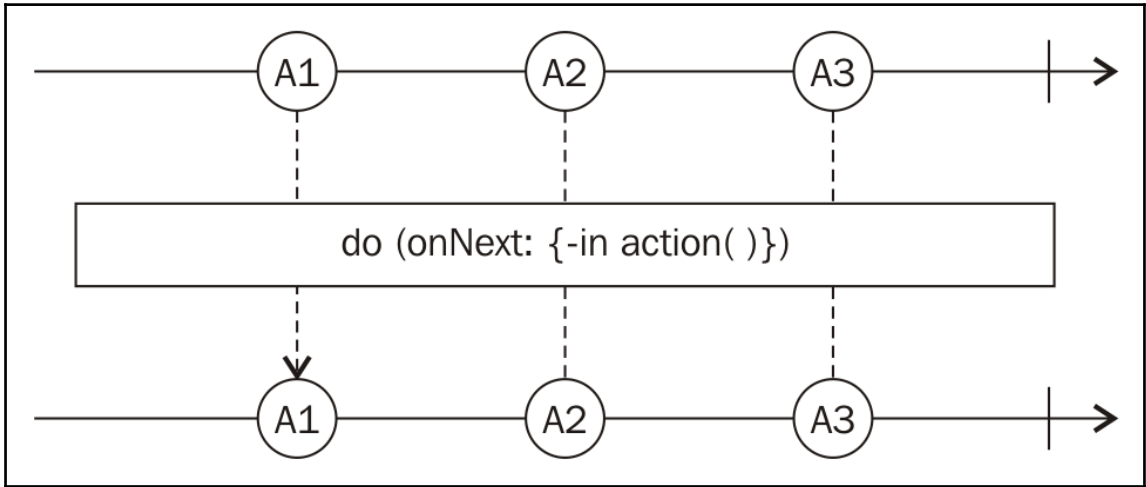


▼ ▶
Procedure executed for: zip

▼ ▶
Procedure executed for: zip
1 : is the first String element on stringPubSubject1 AND 3 : is the first String element on stringPubSubject2

▼ ▶
Procedure executed for: zip
1 : is the first String element on stringPubSubject1 AND 3 : is the first String element on stringPubSubject2
2 : is the second String element on stringPubSubject1 AND 4 : is the second String element on stringPubSubject2

```
Procedure executed for: zip
1 : is the first String element on stringPubSubject1 AND 3 : is the first String element on stringPubSubject2
2 : is the second String element on stringPubSubject1 AND 4 : is the second String element on stringPubSubject2
5 : is the third String element on stringPubSubject1 AND 3 : is the third String element on stringPubSubject2
```



```
RxAdvancedOperators.playground
RxAdvancedOperators > RxAdvancedOperators.playground

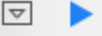
69
70 stringPubSubject1.onNext("is the second String element on stringPubSubject1")
71 stringPubSubject2.onNext("is the first String element on stringPubSubject2")
72 stringPubSubject2.onNext("is the second String element on stringPubSubject2")
73
74
75
76 intPubSubject1.onNext(1)
77
78
79
80
81
82
83
84
85 string element on stringPubSubject1")
86 string element on stringPubSubject2")
87
88
89 }
90
91 */
92
93 exec
94
95
96
97
98
99
100
101
102
103
104 .do(onNext: {
105     print("\(S0)")
106 })
107 }
108
109
110
111
112
113
114
```

Search: degree

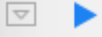
°C °F °

Result of operator "" is unused

```
Procedure executed for: do(on....:)
Subscribed to sequence
-40°F = -40.0°C
0°F = -17.8°C
37°F = 2.8°C
Sequence Disposed
```



```
Procedure executed for: do(on....:)  
Subscribed to sequence  
-40°F = -40.0°C  
0°F = -17.8°C  
37°F = 2.8°C
```



```
Procedure executed for: do(on....:)  
Subscribed to sequence  
-40°F = -40.0°C  
0°F = -17.8°C  
37°F = 2.8°C  
Completed the sequence  
Sequence Disposed
```

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

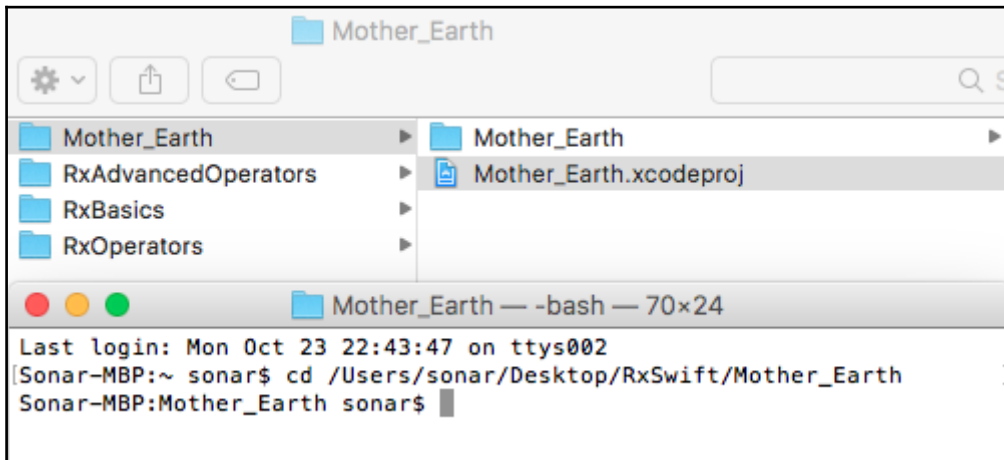
Language:

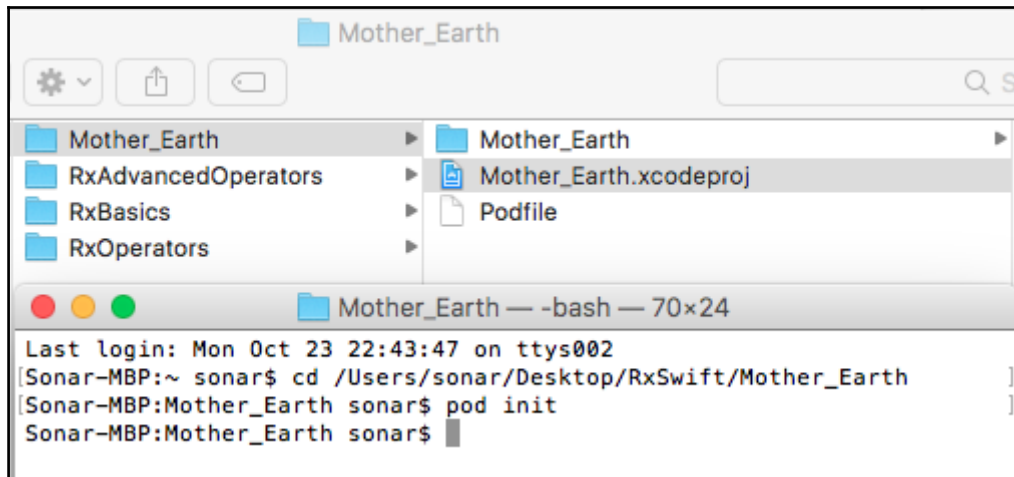
Devices:

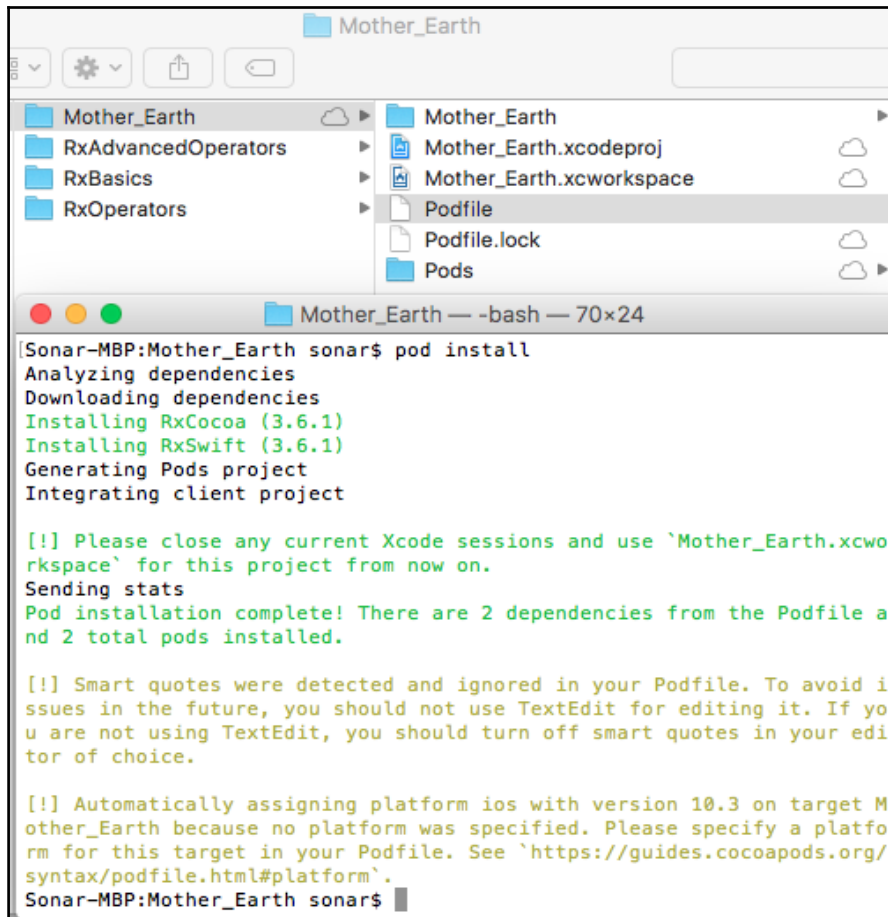
Use Core Data

Include Unit Tests

Include UI Tests







The image shows a macOS file manager window for a project named "Mother_Earth". The project's contents are listed as follows:

- Mother_Earth (folder)
- RxAdvancedOperators (folder)
- RxBasics (folder)
- RxOperators (folder)
- Mother_Earth.xcodeproj (file)
- Mother_Earth.xcworkspace (file)
- Podfile (file)
- Podfile.lock (file)
- Pods (folder)

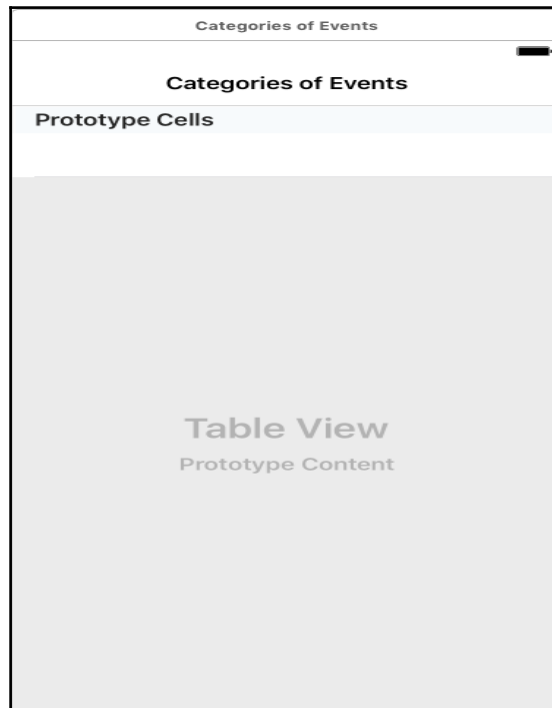
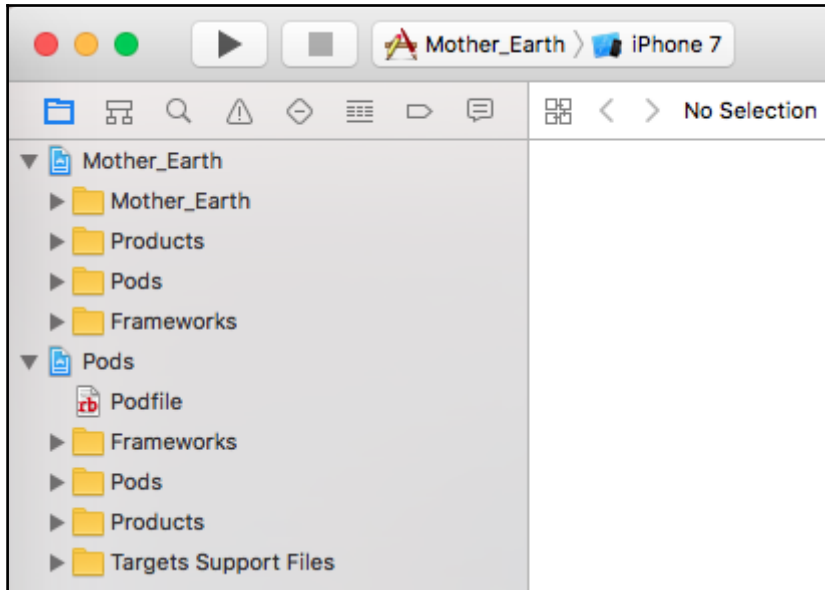
Below the file manager is a terminal window titled "Mother_Earth" with a shell prompt of "-bash" and a window size of "70x24". The terminal shows the execution of the "pod install" command and its output:

```
Sonar-MBP:Mother_Earth sonar$ pod install
Analyzing dependencies
Downloading dependencies
Installing RxCocoa (3.6.1)
Installing RxSwift (3.6.1)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `Mother_Earth.xcworkspace` for this project from now on.
Sending stats
Pod installation complete! There are 2 dependencies from the Podfile and 2 total pods installed.

[!] Smart quotes were detected and ignored in your Podfile. To avoid issues in the future, you should not use TextEdit for editing it. If you are not using TextEdit, you should turn off smart quotes in your editor of choice.

[!] Automatically assigning platform ios with version 10.3 on target Mother_Earth because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.
Sonar-MBP:Mother_Earth sonar$
```

Choose options for your new file:

Class:

Subclass of:

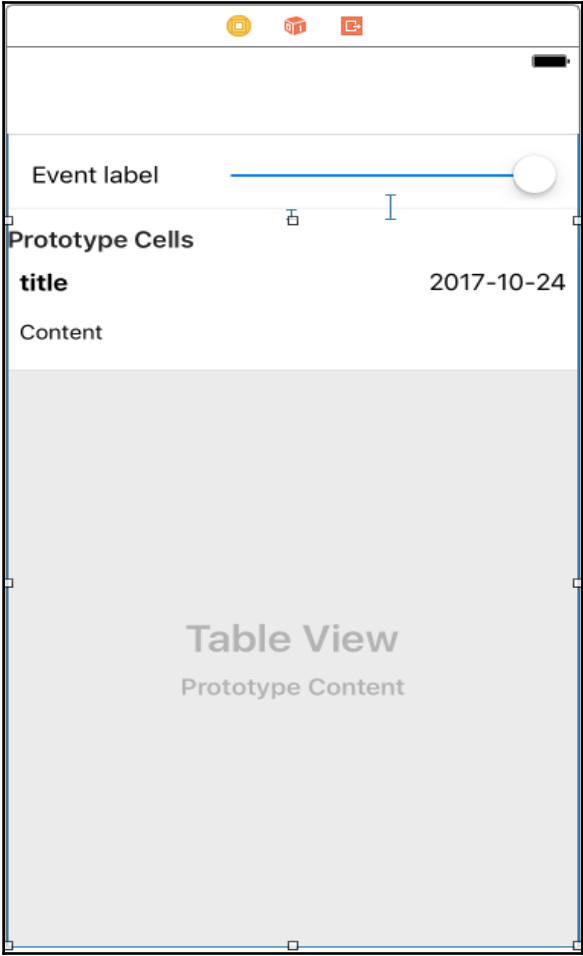
Also create XIB file

Language:

Cancel

Previous

Next



Event label

Prototype Cells

| | |
|----------------|------------|
| title | 2017-10-24 |
| Content | |

Table View

Prototype Content

Custom Class

Class:

Module:

Inherit From Target

Identity

Storyboard ID:

Restoration ID:

Use Storyboard ID

User Defined Runtime Attributes

| Key Path | Type | Value |
|----------|------|-------|
| | | |

Document

Label:

Object ID:

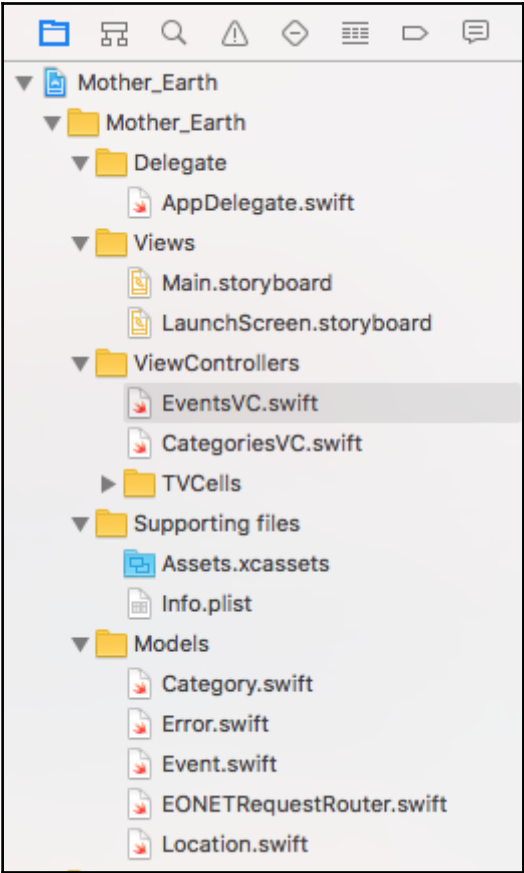
Lock:

Notes:

📄 { } ⌚ 📄

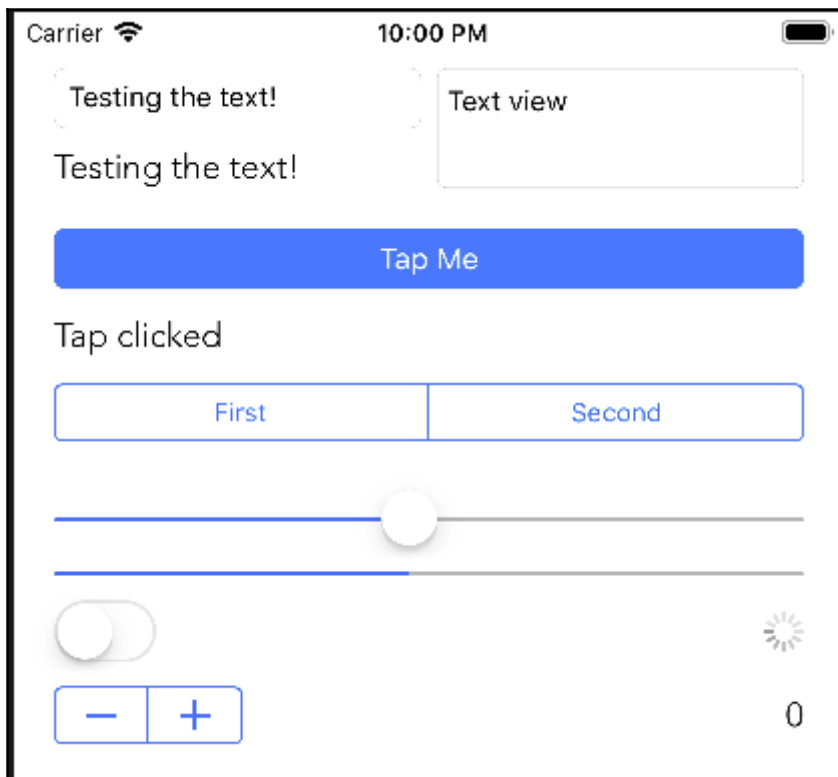
View Controller - A controller that manages a view.

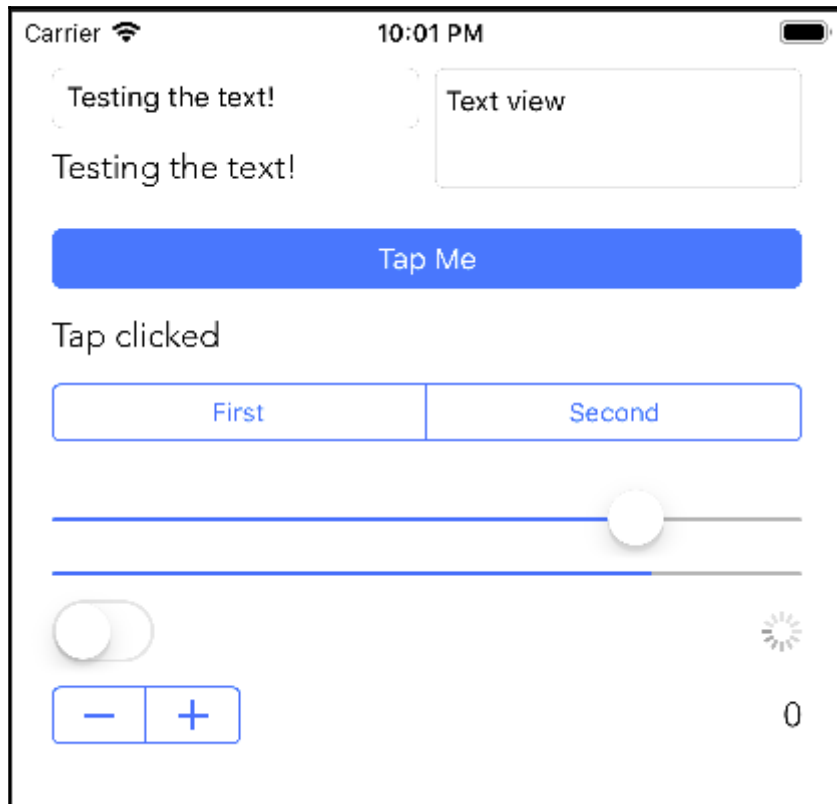
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.



Chapter 7: React to UI Events – Start Subscribing

```
public func asDriver() -> Driver<E> {
    return self.asDriver { (error) -> Driver<E> in
        #if DEBUG
            rxFatalError("Somehow driver received error from a
                source that shouldn't fail.")
        #else
            return Driver.empty()
        #endif
    }
}
```





```
.map { !$0 }  
.bind(to: activityIndicator.rx.isHidden)  
.disposed(by: disposeE
```

Declaration: `var isHidden: UIBindingObserver<UIActivityIndicatorView, Bool> { get }`

Description: Bindable sink for hidden property.

Declared In: RxCocoa

10:01



Text field Text view

Char count: 9

Tap Me

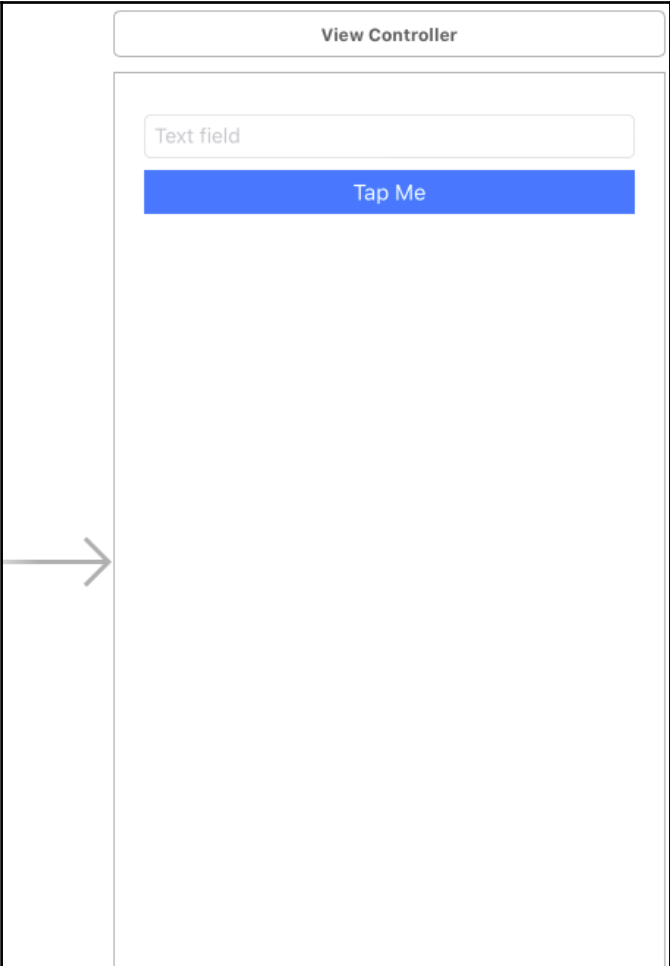
First Second

Slider control

- + 8

| | | | |
|--------------|-----------|-----------|-----------|
| Mon Nov 13 | 9 | 11 | |
| Tue Nov 14 | 10 | 12 | |
| Wed Nov 15 | 11 | 13 | |
| Today | 12 | 14 | AM |
| Fri Nov 17 | 1 | 15 | PM |
| Sat Nov 18 | 2 | 16 | |
| Sun Nov 19 | 3 | 17 | |

Date picked: Nov 16, 2017 at 12:14 AM





Krunoslav Zaher

kzaher



Yury Korolev

yury



Serg Dort

sergdort



Mo Ramezanpoor

mohsenr



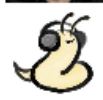
Carlos García

carlosypunto



Scott Gardner

scotteg



Nobuo Saito

tarunon



Junior B.

bontoJR



Jesse Farless

solidcell



Jamie Pinkham

jamiepinkham

```
Bind Table Views

You selected: Serg Dort: github.com/sergdort
You selected: Yury Korolev: github.com/yury
You selected: Mo Ramezanpoor: github.com/mohsenr
You selected: Krunoslav Zaher: github.com/kzaher
```

```
31 let data = Observable.of( [
    bID: "kzaher"),
    : "yury"),
    sergdort"),
    ID: "mohsenr"),
```

Declaration let data: Observable<[Developer]>
Declared In ViewController.swift

```
31 let data = Observable.of( [
    bID: "kzaher"),
    : "yury"),
    sergdort"),
    ID: "mohsenr"),
```

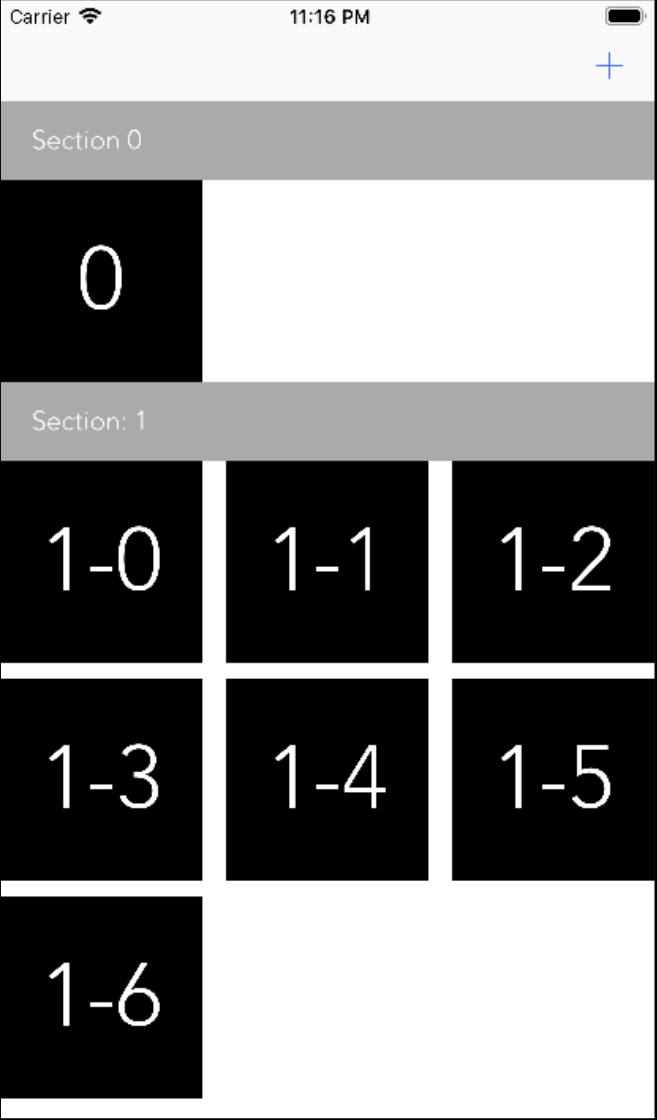
Declaration let data: Observable<[Developer]>
Declared In ViewController.swift

```
dataSource.supplementaryViewFactory ! Expression
```

Declaration var supplementaryViewFactory:
(CollectionViewSectionedDataSource<SectionModel>,
UICollectionView, String, IndexPath) ->
UICollectionViewReusableView { get set }

Declared In RxDataSources





Chapter 8: RxTest and Custom Rx Extensions – Testing with Rx

| Product | Debug | Source |
|----------------|-------|--------|
| Run | | ⌘R |
| Test | | ⌘U |
| Profile | | ⌘I |
| Analyze | | ⇧⌘B |
| Archive | | |
| Build For | | ▶ |
| Perform Action | | ▶ |

```
✓ func testMapOperator(){
28     let localObserver = scheduler.createObserver(Int.self)
29     let observableSeq = scheduler.createHotObservable([
30         next(100,10),
31         next(200,20),
32         next(300,30)
33     ])
34
35     let observableForMap = observableSeq.map {$0 * 2}
36     scheduler.scheduleAt(0){
37         self.subscription = observableForMap.subscribe(localObserver)
38     }
39     scheduler.start()
}
```



```
❌ func testMapOperator(){
28     let localObserver = scheduler.createObserver(Int.self)
29     let observableSeq = scheduler.createHotObservable([
30         next(100,10),
31         next(200,20),
32         next(300,30)
33     ])
34
35     let observableForMap = observableSeq.map {S0 * 2}
36     scheduler.scheduleAt(0){
37         self.subscription = observableForMap.subscribe(localObserver)
38     }
39     scheduler.start()
40
41     let resultsFromMapOperation = localObserver.events.map{
42         S0.value.element!
43     }
44
45     XCTAssertEqual(resultsFromMapOperation, [10,20,30]) ❌ XCTAssertEqual failed: ("[20, 40, 60]") is not equal to ("[10, 20, 30]") -
46 }
```

```
let result = observableToTest.toBlocking() ⚠️ Initialization of immutable value
```

Declaration `func toBlocking(timeout: RxTimeInterval? = default) -> BlockingObservable<Int>`

Description Converts an Observable into a BlockingObservable (an Observable with blocking operators).

Parameters `timeout` Maximal time interval BlockingObservable can block without throwing `RxError.timeout`.

Returns BlockingObservable version of self

Declared in `RxBlocking`

```
next(20)
next(40)
next(60)
completed
Test Case '-[TestingTests.TestingTests testAsynchronousToArray]' passed (0.359 seconds).
```

Chapter 9: Testing Your RxCode – Testing Asynchronous Code

```
7 func exampleWithPulish(){
8     let intervalSeq = Observable<Int>.interval(1, scheduler:
          MainScheduler.instance)
9     .publish()
10
11     intervalSeq
12         .subscribe { print($0) }
13 }
14
15 exampleWithPulish()
```

```
7 func exampleWithPulish(){
8     let intervalSeq = Observable<Int>.interval(1, scheduler:
          MainScheduler.instance)
9     .publish()
10
11     intervalSeq
12         .subscribe { print($0) }
13
14     delayInExecution(2) {
15         _ = intervalSeq.connect()
16     }
17 }
```

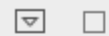
RxSwift.(ConnectableObserver) (7 times)

RxSwift.SubscriptionDisposable (7 times)

()

```
next(0)
next(1)
next(2)
next(3)
next(4)
next(5)
next(6)
```

```
10
11   intervalSeq
12     .subscribe { print($0) }
13
14   delayInExecution(2) {
15     _ = intervalSeq.connect()
16   }
17
18   delayInExecution(4) {
19     _ = intervalSeq
20       .subscribe { print($0) }
21   }
22 }
23
24 exampleWithPulish()
25
```



```
next(0)
next(1)
next(1)
next(2)
next(2)
next(3)
next(3)
next(4)
next(4)
next(5)
next(5)
next(6)
next(6)
```

```
11 intervalSeq
12     .debug()
13     .subscribe { print($0) }
14
15     delayInExecution(2) {
16         _ = intervalSeq.connect()
17     }
18
19     delayInExecution(4) {
20         _ = intervalSeq
21         Stop Playground.subscribe { print($0) }
22     }
23 }
24
25 exampleWithPulish()
```

RxSwift.(SinkDisposer in _B...

(7 times)

()

()

(6 times)

```
2018-01-16 23:18:23.221: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> subscribed
2018-01-16 23:18:26.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(0)
next(0)
2018-01-16 23:18:27.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(1)
next(1)
next(1)
2018-01-16 23:18:28.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(2)
next(2)
next(2)
2018-01-16 23:18:29.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(3)
next(3)
next(3)
2018-01-16 23:18:30.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(4)
next(4)
next(4)
2018-01-16 23:18:31.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(5)
next(5)
next(5)
2018-01-16 23:18:32.416: RxSwift_Chapter9.playground:12 (exampleWithPulish()) -> Event next(6)
next(6)
next(6)
```

```

11     intervalSeq
12         .debug("First")
13         .subscribe { print($0) }
14
15     delayInExecution(2) {
16         _ = intervalSeq.connect()
17     }
18
19     delayInExecution(4) {
20         _ = intervalSeq
21             .debug("Second")
22             .subscribe { print($0) }
23     }
24 }
25
26 exampleWithPulish()
27

```

RxSwift.(SinkDisposer in _B...

(4 times)

()

()

(3 times)

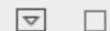


```

2018-01-16 23:26:07.882: First -> subscribed
2018-01-16 23:26:10.890: First -> Event next(0)
next(0)
2018-01-16 23:26:11.888: Second -> subscribed
2018-01-16 23:26:11.889: First -> Event next(1)
next(1)
2018-01-16 23:26:11.889: Second -> Event next(1)
next(1)
2018-01-16 23:26:12.889: First -> Event next(2)
next(2)
2018-01-16 23:26:12.890: Second -> Event next(2)
next(2)
2018-01-16 23:26:13.889: First -> Event next(3)
next(3)
2018-01-16 23:26:13.890: Second -> Event next(3)
next(3)

```

```
11 intervalSeq
12     .debug("First")
13     .subscribe { _ in }
14
15 delayInExecution(2) {
16     _ = intervalSeq.connect()
17 }
18
19 delayInExecution(4) {
20     intervalSeq
21         .debug("Second")
22         .subscribe { _ in }
23         .dispose()
24 }
25 }
26
27 exampleWithPulish()
```



```
2018-01-16 23:32:31.592: First -> subscribed
2018-01-16 23:32:34.778: First -> Event next(0)
2018-01-16 23:32:35.715: Second -> subscribed
2018-01-16 23:32:35.715: Second -> isDisposed
2018-01-16 23:32:35.778: First -> Event next(1)
2018-01-16 23:32:36.778: First -> Event next(2)
2018-01-16 23:32:37.778: First -> Event next(3)
2018-01-16 23:32:38.778: First -> Event next(4)
2018-01-16 23:32:39.778: First -> Event next(5)
2018-01-16 23:32:40.778: First -> Event next(6)
2018-01-16 23:32:41.778: First -> Event next(7)
2018-01-16 23:32:42.778: First -> Event next(8)
```

```
4 exampleOf(description: "total") {
5     print(RxSwift.Resources.total)
6
7     let object = NSObject()
8     var disposeBag = DisposeBag()
9     print(RxSwift.Resources.total)
10
11     let stringSequence = Observable.just("I am a string")
12     print(RxSwift.Resources.total)
13
```



===> Example of: total ===>

0

2

3

```
10
11     let stringSequence = Observable.just("I am a string")
12     print(RxSwift.Resources.total)
13
14     stringSequence
15         .subscribe(onNext: { _ in
16             print(RxSwift.Resources.total)
17         })
18         .disposed(by: disposeBag)
```



===> Example of: total ===>

0

2

3

4

```
11 let stringSequence = Observable.just("I am a string")
12 print(RxSwift.Resources.total)
13
14 stringSequence
15     .subscribe(onNext: { _ in
16         print(RxSwift.Resources.total)
17     })
18     .disposed(by: disposeBag)
19
20 disposeBag = DisposeBag()
21 print(RxSwift.Resources.total)
22 }
23 print(RxSwift.Resources.total)
```

24



===> Example of: total ===>

0

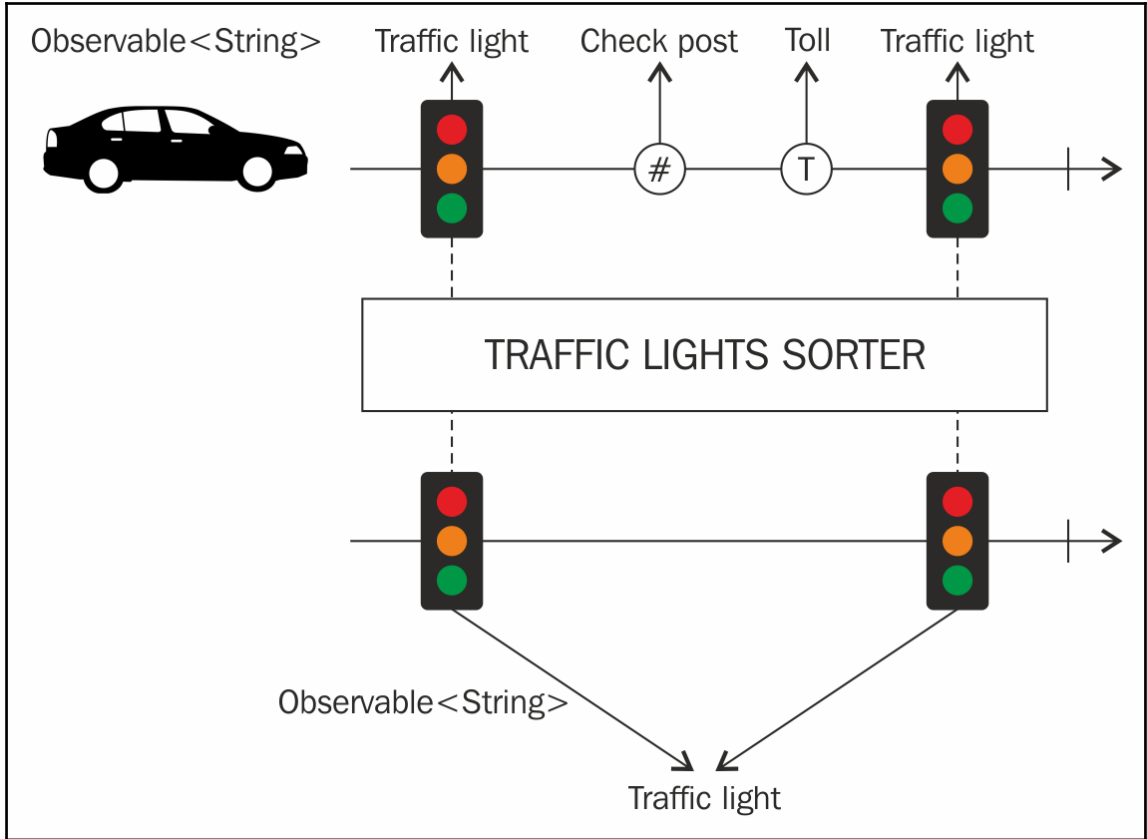
2

3

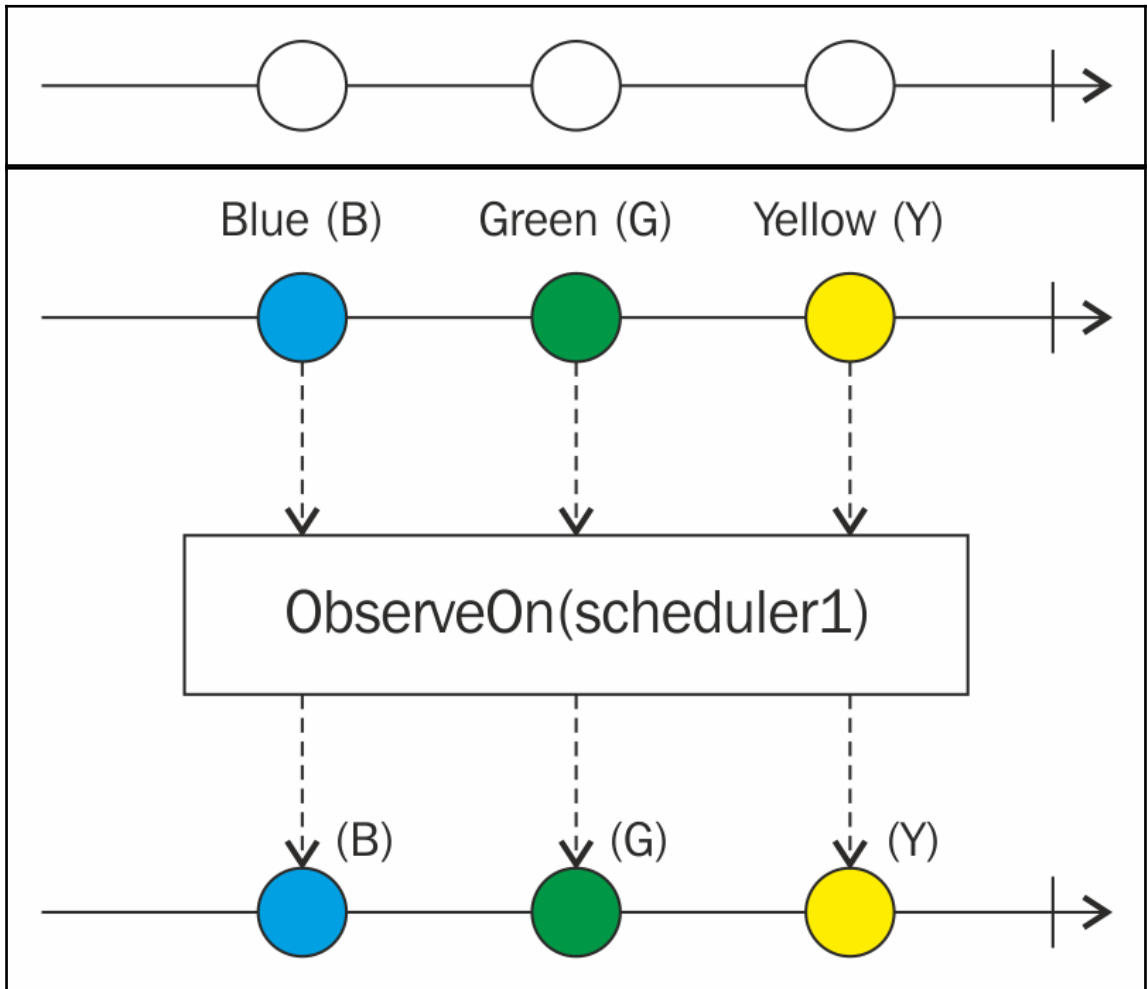
4

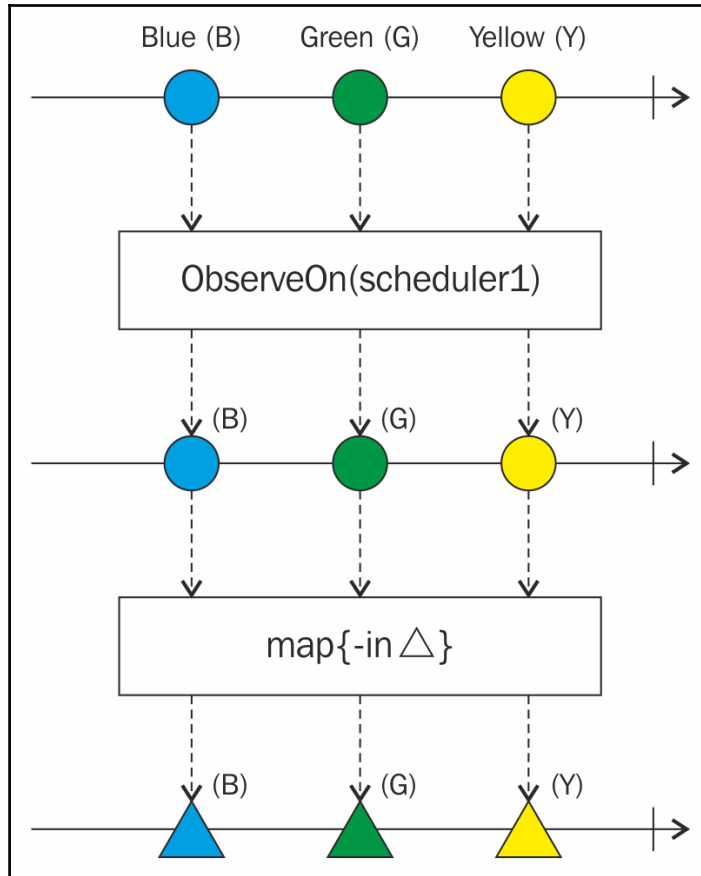
3

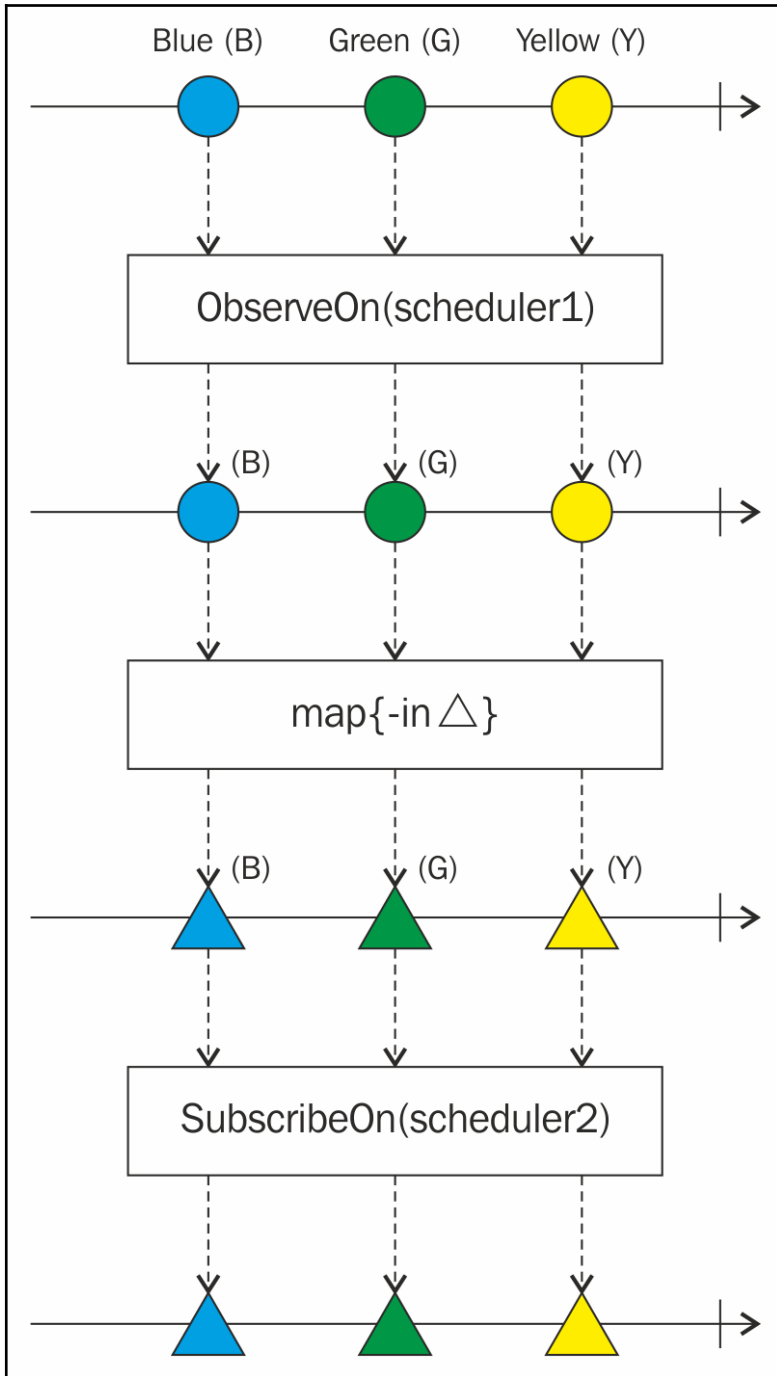
0

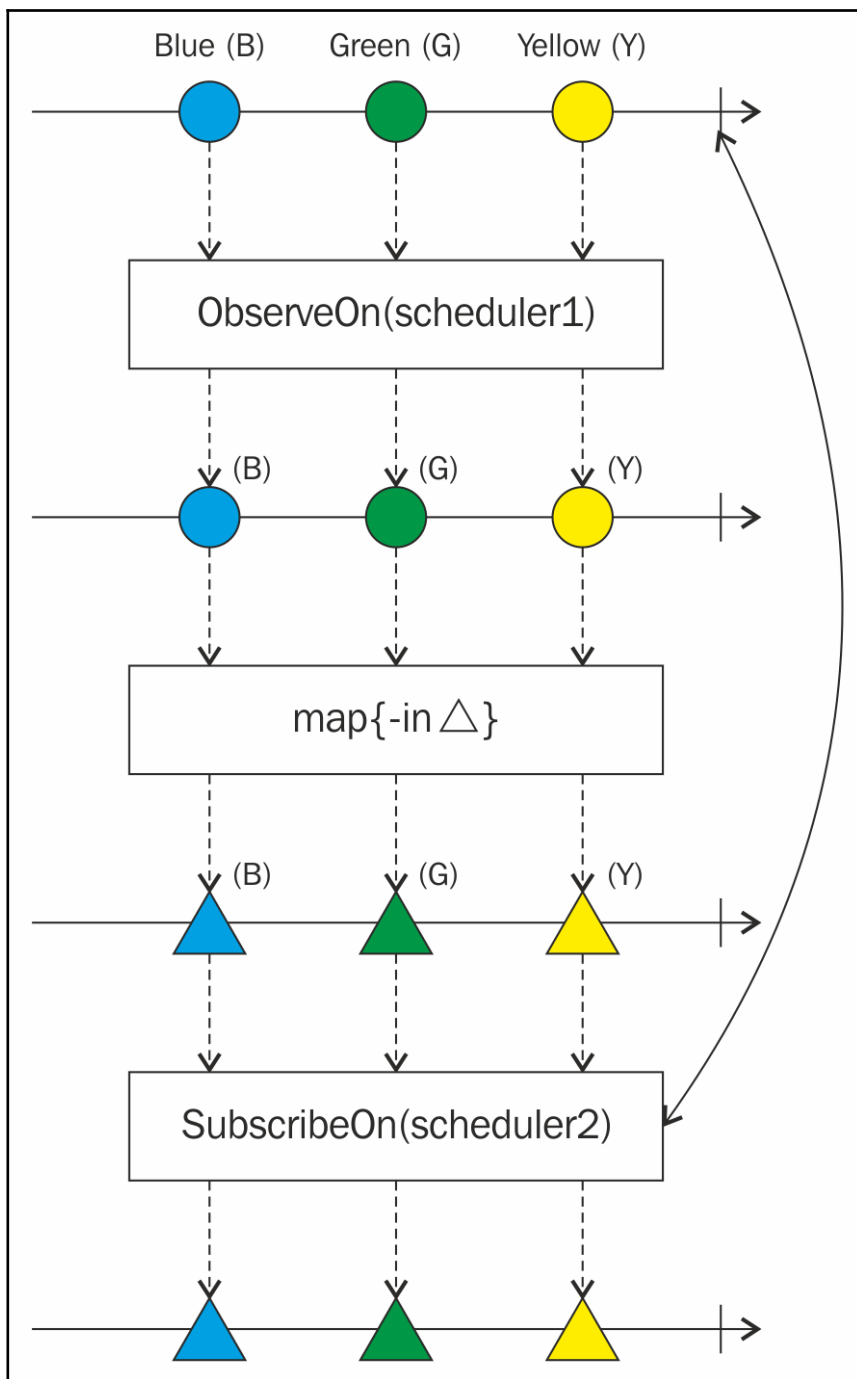


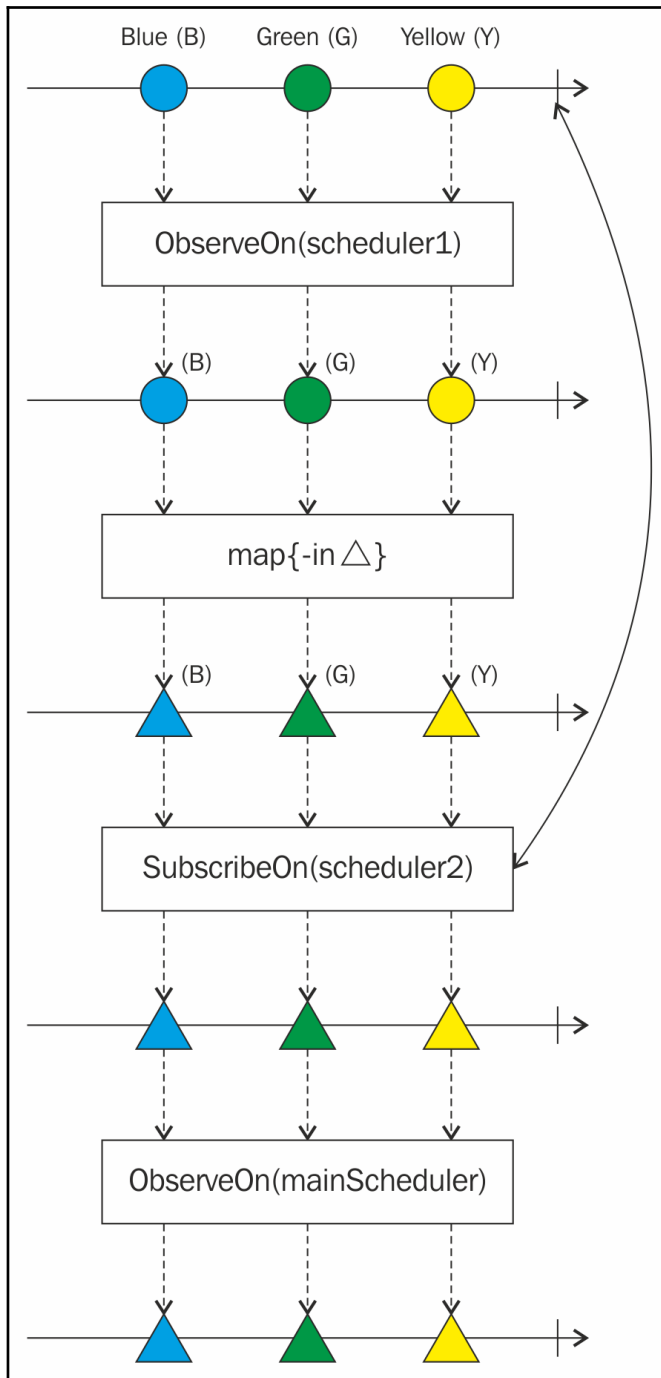
Chapter 10: Schedule Your Tasks, Don't Queue!

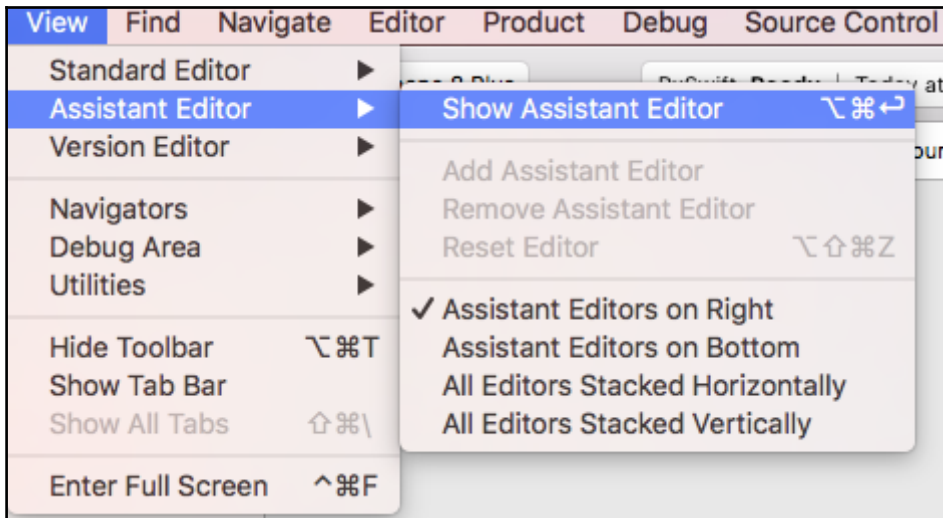













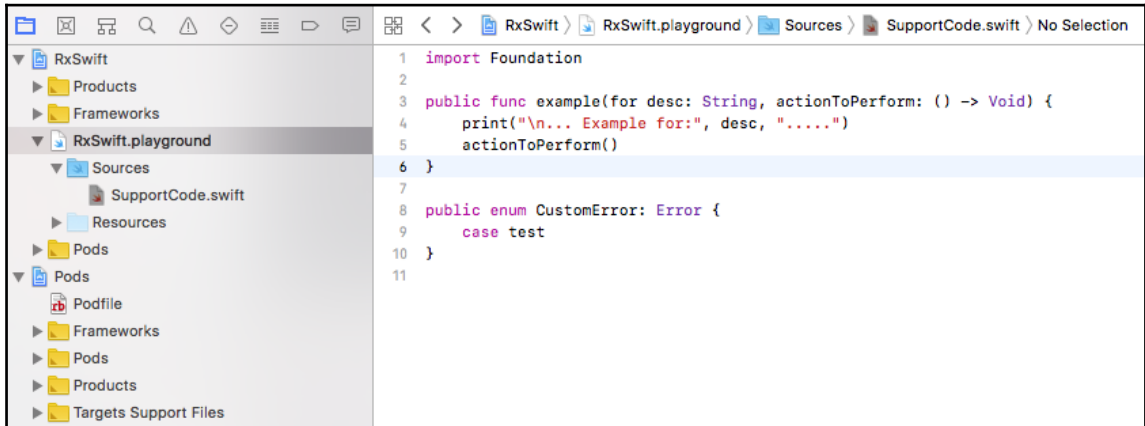




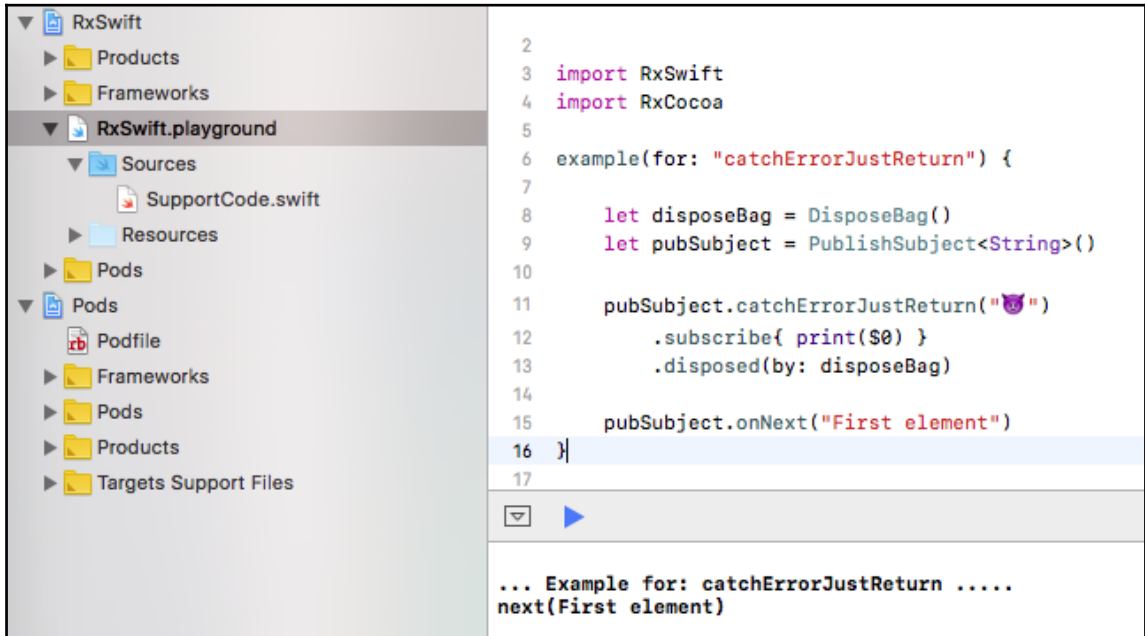
| | | |
|--|---|---|
| <pre> imageData .observeOn(concurrentScheduler) .map { UIImage(data: \$0) } </pre> | (2 times) |   |
|  | | |
| <pre> .observeOn(MainScheduler.instance) .subscribe(onNext: { imageView.image = \$0 }) .disposed(by: disposeBag) </pre> | (2 times) |  |
| <pre>imageData.onNext(swiftImageData)</pre> | RxSwift.PublishSubject<Foundation.Data> |  |
| <pre>imageData.onNext(rxImageData)</pre> | RxSwift.PublishSubject<Foundation.Data> |  |

17 It is extremely important that this scheduler is serial, because
18 certain operator perform optimizations that rely on that
property.
22 will be assumed, and internal serial proxy dispatch queue will be
created.

Chapter 11: Subscribe to Errors and Save Your App



```
1 import Foundation
2
3 public func example(for desc: String, actionToPerform: () -> Void) {
4     print("\n... Example for:", desc, ".....")
5     actionToPerform()
6 }
7
8 public enum CustomError: Error {
9     case test
10 }
11
```



```
2
3 import RxSwift
4 import RxCocoa
5
6 example(for: "catchErrorJustReturn") {
7
8     let disposeBag = DisposeBag()
9     let pubSubject = PublishSubject<String>()
10
11     pubSubject.catchErrorJustReturn("🙄")
12         .subscribe{ print($0) }
13         .disposed(by: disposeBag)
14
15     pubSubject.onNext("First element")
16 }
17
```

... Example for: catchErrorJustReturn
next(First element)

```
8     let disposeBag = DisposeBag()
9     let pubSubject = PublishSubject<String>()
10
11     pubSubject.catchErrorJustReturn("🐾")
12         .subscribe{ print($0) }
13         .disposed(by: disposeBag)
14
15     pubSubject.onNext("First element")
16     pubSubject.onError(CustomError.test)
17 }
```



```
... Example for: catchErrorJustReturn .....
next(First element)
next(🐾)
completed
```

```

19 example(for: "catchError") {
20     let disposeBag = DisposeBag()
21     let pubSubject = PublishSubject<String>()
22
23     let recoverySeq = PublishSubject<String>()
24
25     pubSubject.catchError{
26         print("Error=", $0)
27         return recoverySeq
28     }
29     .subscribe { print($0) }
30     .disposed(by: disposeBag)
31
32     pubSubject.onNext("First element")
33     pubSubject.onError(CustomError.test)
34 }

```



```

... Example for: catchErrorJustReturn .....
next(First element)
next(🐾)
completed

... Example for: catchError .....
next(First element)
Error= test

```

```

32     pubSubject.onNext("First element")
33     pubSubject.onError(CustomError.test)
34     pubSubject.onNext("Second element")
35 }

```



```

... Example for: catchErrorJustReturn .....
next(First element)
next(🐾)
completed

... Example for: catchError .....
next(First element)
Error= test

```

```
32     pubSubject.onNext("First element")
33     pubSubject.onError(CustomError.test)
34     pubSubject.onNext("Second element")
35
36     recoverySeq.onNext("Third element")
37 }
```



```
... Example for: catchErrorJustReturn .....
next(First element)
next(🐼)
completed

... Example for: catchError .....
next(First element)
Error= test
next(Third element)
```

```
39 example(for: "retry") {
40     let disposeBag = DisposeBag()
41     var shouldEmitError = true
42
43     let observableSeq = Observable<Int>.create { observer in
44         observer.onNext(10)
45         observer.onNext(20)
46
47         if shouldEmitError{
48             observer.onError(CustomError.test)
49             shouldEmitError = false
50         }
51
52         observer.onNext(30)
53         observer.onCompleted()
54
55         return Disposables.create()
56     }
57
58     observableSeq.subscribe{ print($0) }
59     .disposed(by: disposeBag)
60 }
```



next(Third element)

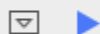
... Example for: retry

next(10)

next(20)

error(test)

```
39 example(for: "retry") {
40     let disposeBag = DisposeBag()
41     var shouldEmitError = true
42
43     let observableSeq = Observable<Int>.create { observer in
44         observer.onNext(10)
45         observer.onNext(20)
46
47         if shouldEmitError{
48             observer.onError(CustomError.test)
49             shouldEmitError = false
50         }
51
52         observer.onNext(30)
53         observer.onCompleted()
54
55         return Disposables.create()
56     }
57
58     observableSeq.retry()
59     .subscribe{ print($0) }
60     .disposed(by: disposeBag)
61 }
```



```
... Example for: retry .....
next(10)
next(20)
next(10)
next(20)
next(30)
completed
```

```
63 example(for: "Driver onErrorJustReturn"){
64
65     let disposeBag = DisposeBag()
66     let pubSubject = PublishSubject<Int>()
67
68     pubSubject.asDriver(onErrorJustReturn: 1000)
69         .drive(onNext: {
70             print ( $0 )
71         })
72     .disposed(by: disposeBag)
73
74     pubSubject.onNext(10)
75     pubSubject.onNext(20)
76 }
```



```
next(30)
completed
```

```
... Example for: Driver onErrorJustReturn .....
```

```
10
```

```
20
```

```
63 example(for: "Driver onErrorJustReturn"){
64
65     let disposeBag = DisposeBag()
66     let pubSubject = PublishSubject<Int>()
67
68     pubSubject.asDriver(onErrorJustReturn: 1000)
69         .drive(onNext: {
70             print ( $0 )
71         })
72     .disposed(by: disposeBag)
73
74     pubSubject.onNext(10)
75     pubSubject.onNext(20)
76     pubSubject.onError(CustomError.test)
77 }
78 }
```



completed

```
... Example for: Driver onErrorJustReturn .....
10
20
1000
```



```
80 example(for: "Driver onErrorDriveWith"){
81
82     let disposeBag = DisposeBag()
83     let pubSubject = PublishSubject<Int>()
84
85     let recoverySubject = PublishSubject<Int>()
86
87     pubSubject.asDriver(onErrorDriveWith: recoverySubject.asDriver(onErrorJustReturn:
88         1000))
89         .drive(onNext: {
90             print ( $0 )
91         })
92         .disposed(by: disposeBag)
93
94     pubSubject.onNext(10)
95     pubSubject.onNext(20)
96
97     pubSubject.onError(CustomError.test)
98 }
```



```
20
1000

... Example for: Driver onErrorDriveWith .....
10
20
```

```
80 example(for: "Driver onErrorDriveWith"){
81
82     let disposeBag = DisposeBag()
83     let pubSubject = PublishSubject<Int>()
84
85     let recoverySubject = PublishSubject<Int>()
86
87     pubSubject.asDriver(onErrorDriveWith: recoverySubject.asDriver(onErrorJustReturn:
88         1000))
89         .drive(onNext: {
90             print ( $0 )
91         })
92         .disposed(by: disposeBag)
93
94     pubSubject.onNext(10)
95     pubSubject.onNext(20)
96
97     pubSubject.onError(CustomError.test)
98     recoverySubject.onNext(100)
99 }
```



1000

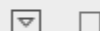
... Example for: Driver onErrorDriveWith

10

20

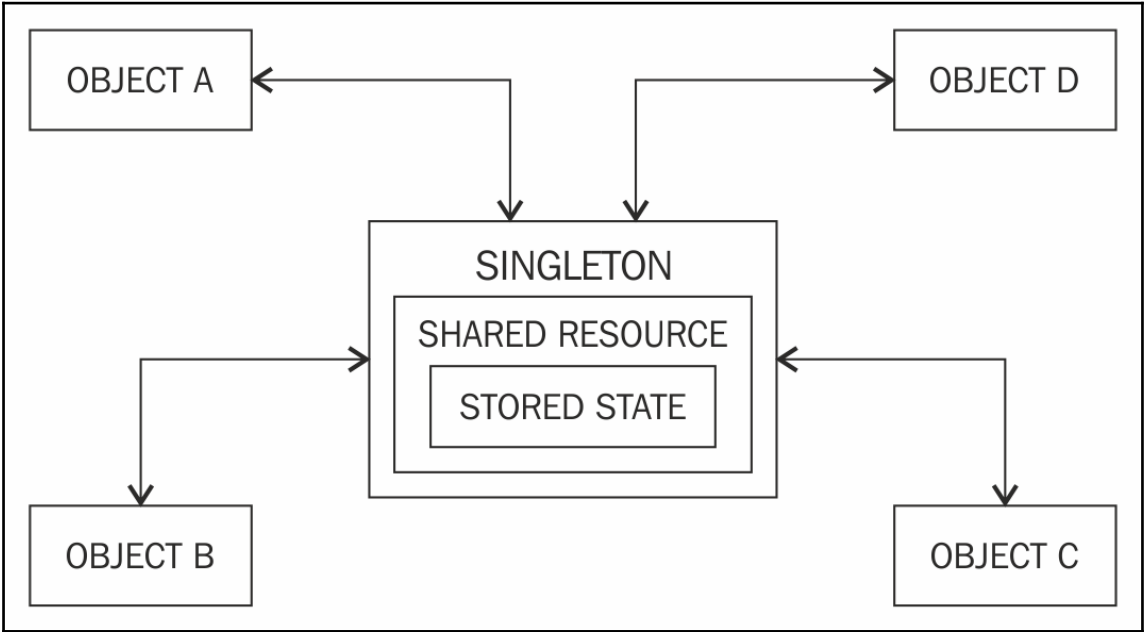
100

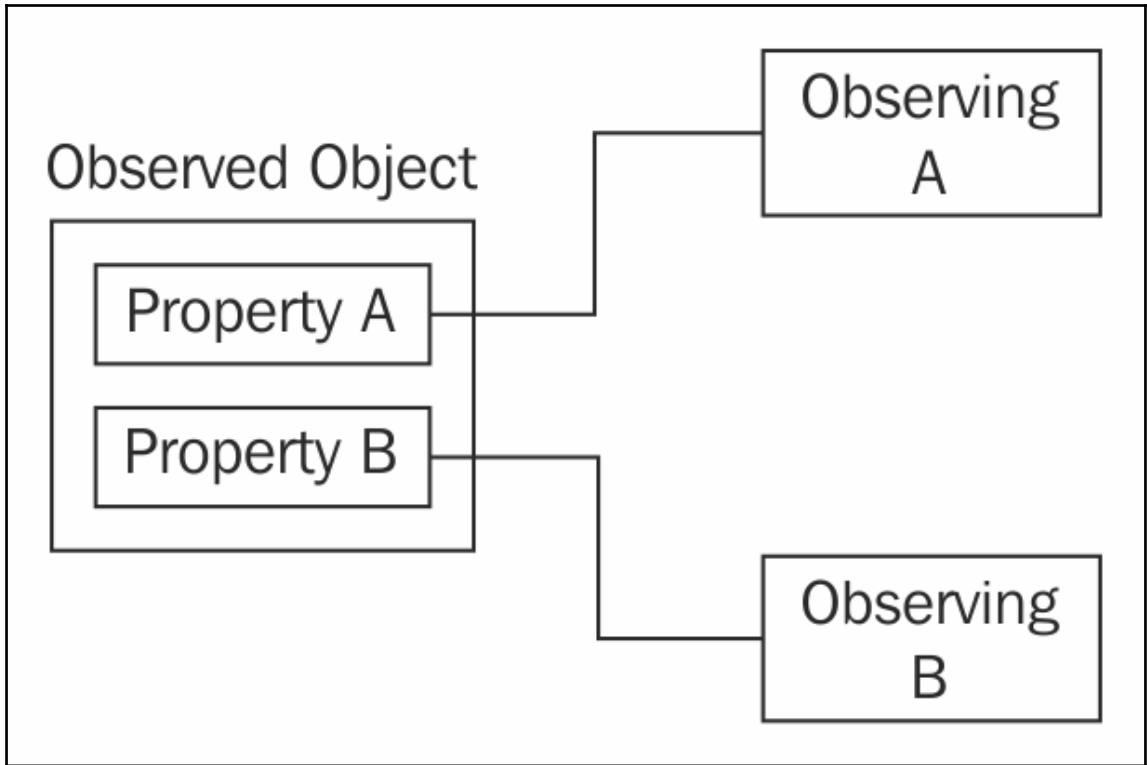
```
104 print(... Example for: Driver onErrorRecover.....")
105 import PlaygroundSupport
106 PlaygroundPage.current.needsIndefiniteExecution = true
107
108 let disposeBag = DisposeBag()
109 let pubSubject = PublishSubject<Int>()
110
111 pubSubject.asDriver{
112     print( "Error:", $0 )
113     return Driver.just(1000)
114 }
115     .drive(onNext: {
116         print ( $0 )
117     })
118     .disposed(by: disposeBag)
119
120 pubSubject.onNext(10)
121 pubSubject.onNext(20)
122
123 pubSubject.onError(CustomError.test)
```

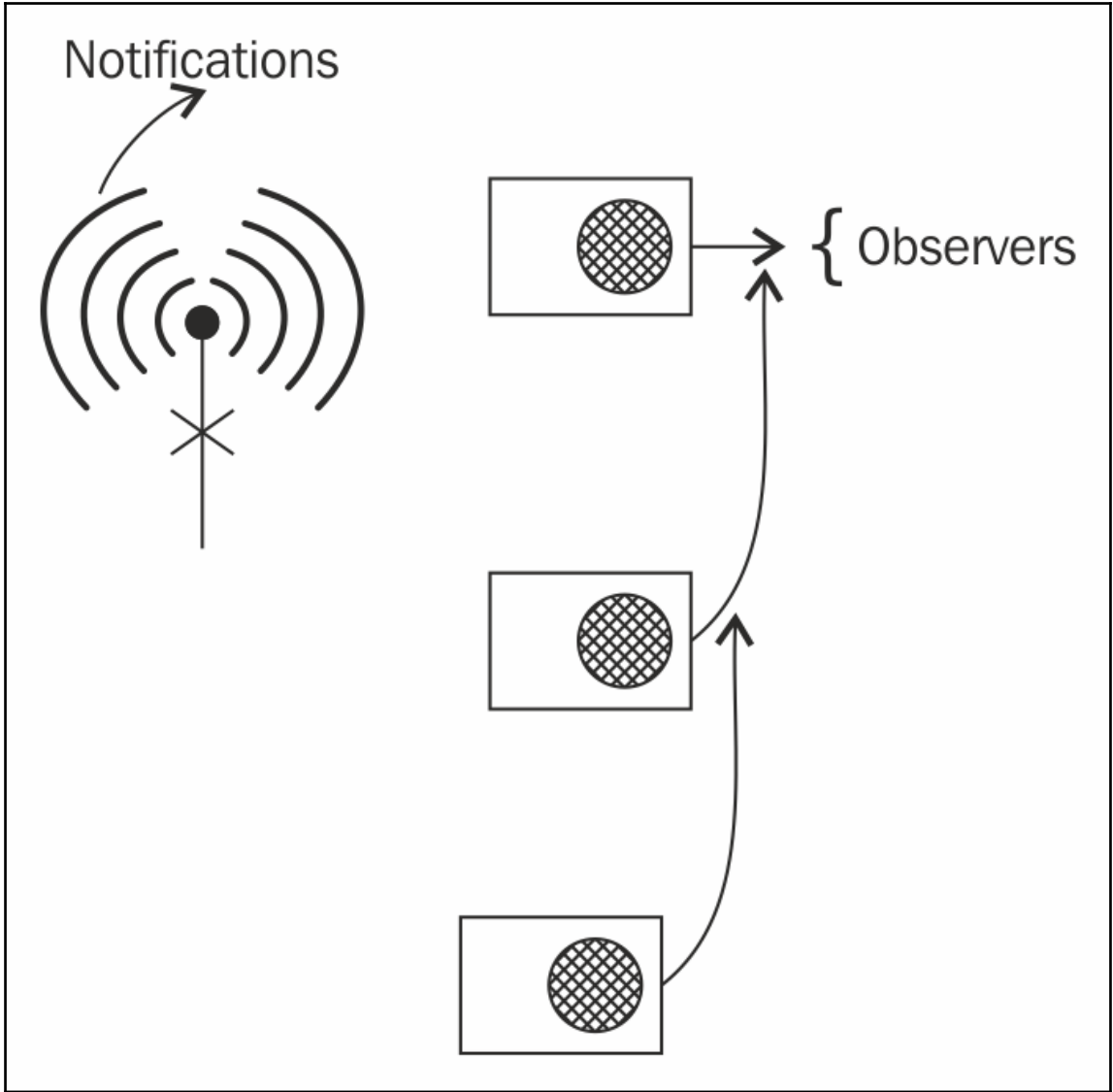


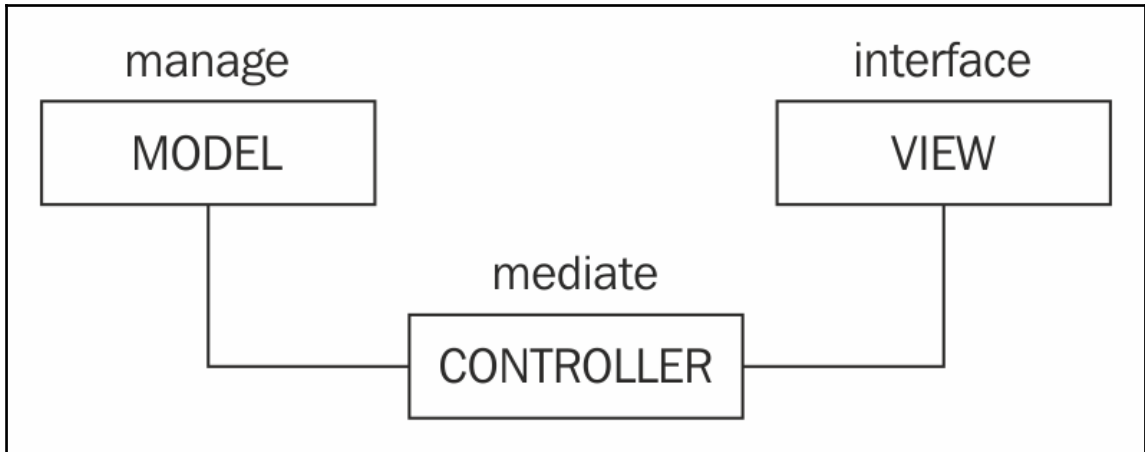
```
100
... Example for: Driver onErrorRecover.....
10
20
Error: test
1000
```

Chapter 12: Functional and Reactive App-architecture









```

1 //
2 // SingletonObject.swift
3 // Singleton
4 //
5 // Created by Mac Mini on 28/12/17.
6 // Copyright © 2017 Mac Mini. All rights reserved.
7 //
8
9 import Cocoa
10
11 class SingletonObject: NSObject {
12
13 }
14
  
```

```

9 import Foundation
10 var kvcObject = KVCObject()
11 print("KVC value = \(kvcObject.value(forKey: "stringToFind"))")
  
```

Thread 1: signal SIGABRT

```

2017-12-30 20:02:18.226144+1100 Singleton[4139:8384178] [DYMTLInitPlatform] platform initialization successful
2017-12-30 20:02:18.280224+1100 Singleton[4139:8384110] *** Terminating app due to uncaught exception 'NSUnknownKeyException', reason:
'[-[Singleton.KVCObject 0x101257650 valueForUndefinedKey:]: this class is not key value coding-compliant for the key stringToFind.'
*** First throw call stack:
(
  0 CoreFoundation                0x00007fff8cf6057b __exceptionPreprocess + 171
  1 libobjc.A.dylib                0x00007fffa21c11da objc_exception_throw + 48
  2 CoreFoundation                0x00007fff8cf604c9 -[NSObject raise] + 9
  3 Foundation                     0x00007fff8bea5bdce -[NSObject(NSKeyValueCoding) valueForKey:] + 226
  4 Foundation                     0x00007fff8e92da7c -[NSObject(NSKeyValueCoding) valueForKey:] + 283
  5 Singleton                      0x0000000100002c40 main + 480
  6 libdyld.dylib                 0x00007fffa2aa2235 start + 1
  7 ???                            0x0000000000000001 0x0 + 1
)
libc++abi.dylib: terminating with uncaught exception of type NSEException
(1ldb)
  
```

>

```

11 class KVCObject: NSObject {
12
13     override func value(forKey key: String) -> Any? {
14         print("we do not have any key for \(key)")
15         return nil
16     }
17 }

```

2018-01-01 16:29:48.438635+1100 Singleton[4872:8654402] [DYMTLInitPlatform] platform initialization successful
we do not have any key for stringToFind
KVC value = nil
Program ended with exit code: 0

2018-01-01 16:51:48.765388+1100 Singleton[4948:8661962] [DYMTLInitPlatform] platform initialization successful
we do not have any key for stringToFind
KVC value = nil
we do not have any key for numberToFind
KVC value of Second Key = nil
Program ended with exit code: 0

```

13 @objc dynamic var stringToFind: String
14
15 public override init() {
16     self.stringToFind = "First value"
17 }
18 override func value(forKey key: String) -> Any? {
19     print("we do not have any key for \(key)")
20     return "nil"
21 }
22

```

2018-01-02 22:42:03.343969+1100 Singleton[5490:8690606] [DYMTLInitPlatform] platform initialization successful
KVC value = First value
we do not have any key for numberToFind
KVC value of Second Key = nil
Program ended with exit code: 0


```

9 import UIKit
10
11 @UIApplicationMain
12 class AppDelegate: UIResponder, UIApplicationDelegate {
13
14     var window: UIWindow?
15     func application(_ application: UIApplication, didFinishLaunchingWithOptions
        launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
16         // Override point for customization after application launch.
17         let notificationCenter = NotificationCenter.default
18         notificationCenter.addObserver(forName: nil, object: nil, queue: nil) {notification
            in
19             print("Notification captured \(notification.name)")
20         }
21         return true
22     }
23 }

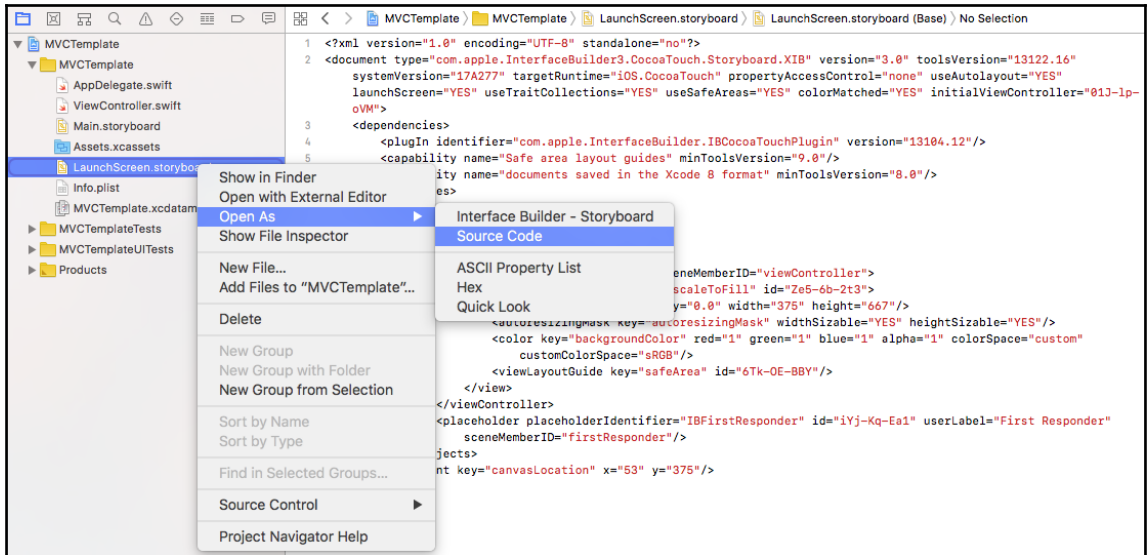
```

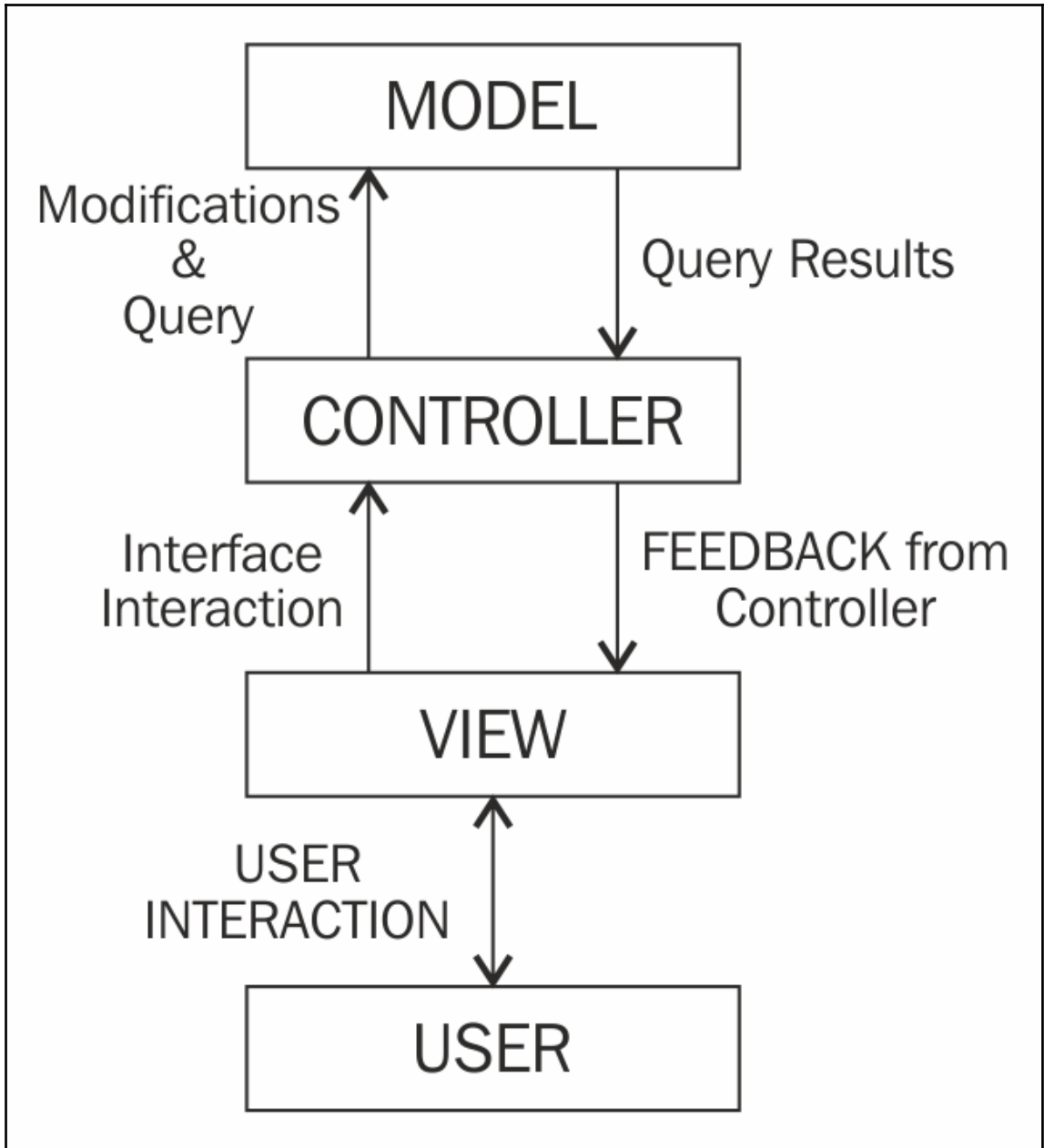
NotificationExample

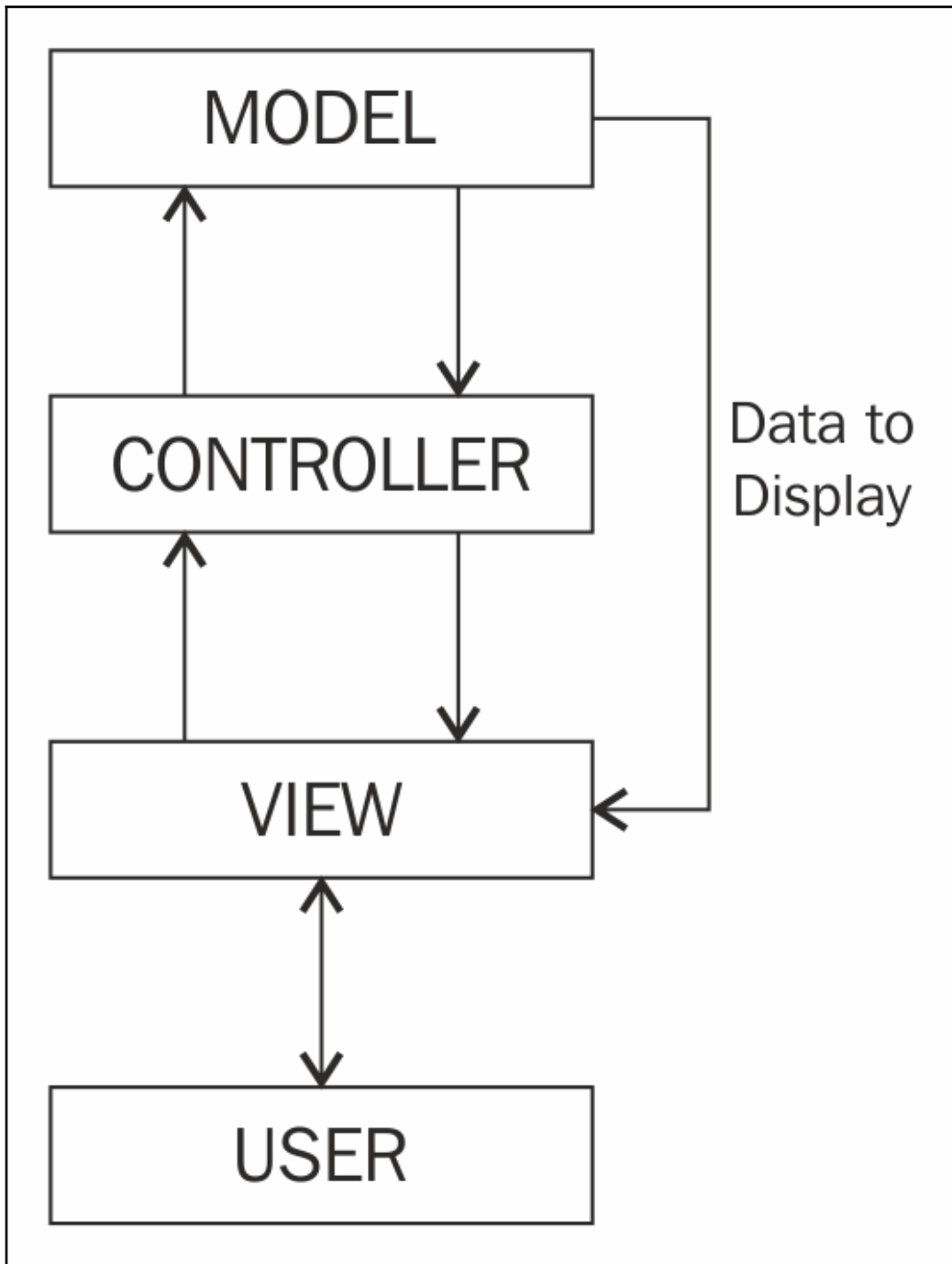
```

Notification captured Name(_rawValue: _UIApplicationStatusBarHiddenStateChangedNotification)
Notification captured Name(_rawValue: UIWindowDidBecomeVisibleNotification)
Notification captured Name(_rawValue: UIWindowDidBecomeKeyNotification)
Notification captured Name(_rawValue: UIApplicationDidFinishLaunchingNotification)
Notification captured Name(_rawValue: UIViewAnimationDidCommitNotification)
Notification captured Name(_rawValue: UIViewAnimationDidStopNotification)
Notification captured Name(_rawValue: _UIWindowContentWillRotateNotification)
Notification captured Name(_rawValue: _UIWindowContentWillRotateNotification)
Notification captured Name(_rawValue: UIDeviceOrientationDidChangeNotification)
Notification captured Name(_rawValue: _UIApplicationDidRemoveDeactivationReasonNotification)
Notification captured Name(_rawValue: UIDeviceOrientationDidChangeNotification)
Notification captured Name(_rawValue: UIDeviceOrientationDidChangeNotification)
Notification captured Name(_rawValue: _UIApplicationDidRemoveDeactivationReasonNotification)
Notification captured Name(_rawValue: UIApplicationDidBecomeActiveNotification)
Notification captured Name(_rawValue:
UIStatusBarItemViewShouldEndDisablingRasterizationNotification)
Notification captured Name(_rawValue: UIStatusBarTimeItemViewDidMoveNotification)
Notification captured Name(_rawValue: UIStatusBarTimeItemViewDidMoveNotification)

```







Chapter 13: Finish a Real-World Application

```
← → ↻ 🔒 Secure | https://api.github.com/users/NavdeepSinghh/repos ☆ ⋮
{
  {
    "id": 21881534,
    "name": "book-notes",
    "full_name": "NavdeepSinghh/book-notes",
    "owner": {
      "login": "NavdeepSinghh",
      "id": 8130835,
      "avatar_url": "https://avatars0.githubusercontent.com/u/8130835?v=4",
      "gravatar_id": "",
      "url": "https://api.github.com/users/NavdeepSinghh",
      "html_url": "https://github.com/NavdeepSinghh",
      "followers_url": "https://api.github.com/users/NavdeepSinghh/followers",
      "following_url": "https://api.github.com/users/NavdeepSinghh/following{/other_user}",
      "gists_url": "https://api.github.com/users/NavdeepSinghh/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/NavdeepSinghh/starred{/owner}/{repo}",
      "subscriptions_url": "https://api.github.com/users/NavdeepSinghh/subscriptions",
      "organizations_url": "https://api.github.com/users/NavdeepSinghh/orgs",
      "repos_url": "https://api.github.com/users/NavdeepSinghh/repos",
      "events_url": "https://api.github.com/users/NavdeepSinghh/events{/privacy}",
      "received_events_url": "https://api.github.com/users/NavdeepSinghh/received_events",
      "type": "User",
      "site_admin": false
    },
    "private": false,
    "html_url": "https://github.com/NavdeepSinghh/book-notes",
    "description": "Notes from books I've read. Most of which are technical or entrepreneurial.",
    "fork": true,
    "url": "https://api.github.com/repos/NavdeepSinghh/book-notes",
    "forks_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/forks",
    "keys_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/keys{/key_id}",
    "collaborators_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/collaborators{/collaborator}",
    "teams_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/teams",
    "hooks_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/hooks",
    "issue_events_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/issues/events{/number}",
    "events_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/events",
    "assignees_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/assignees{/user}",
    "branches_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/branches{/branch}",
    "tags_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/tags",
    "blobs_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/git/blobs{/sha}",
    "git_tags_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/git/tags{/sha}",
    "git_refs_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/git/refs{/sha}",
    "trees_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/git/trees{/sha}",
    "statuses_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/statuses{/sha}",
    "languages_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/languages",
    "stargazers_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/stargazers",
    "contributors_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/contributors",
    "subscribers_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/subscribers",
    "subscription_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/subscription",
    "commits_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/commits{/sha}",
    "git_commits_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/git/commits{/sha}",
    "comments_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/comments{/number}",
    "issue_comment_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/issues/comments{/number}",
    "contents_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/contents/{+path}",
    "compare_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/compare/{base}...{head}",
    "merges_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/merges",
    "archive_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/{archive_format}/{ref}",
    "downloads_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/downloads",
    "issues_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/issues{/number}",
    "pulls_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/pulls{/number}",
    "milestones_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/milestones{/number}",
    "notifications_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/notifications?since=all",
    "labels_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/labels{/name}",
    "releases_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/releases{/id}",
    "deployments_url": "https://api.github.com/repos/NavdeepSinghh/book-notes/deployments",
    "created_at": "2014-07-16T01:05:12Z",
    "updated_at": "2014-07-09T04:08:12Z",
    "pushed_at": "2014-05-26T18:14:13Z",
    "git_url": "git://github.com/NavdeepSinghh/book-notes.git",
    "ssh_url": "git@github.com:NavdeepSinghh/book-notes.git",
  }
}
```

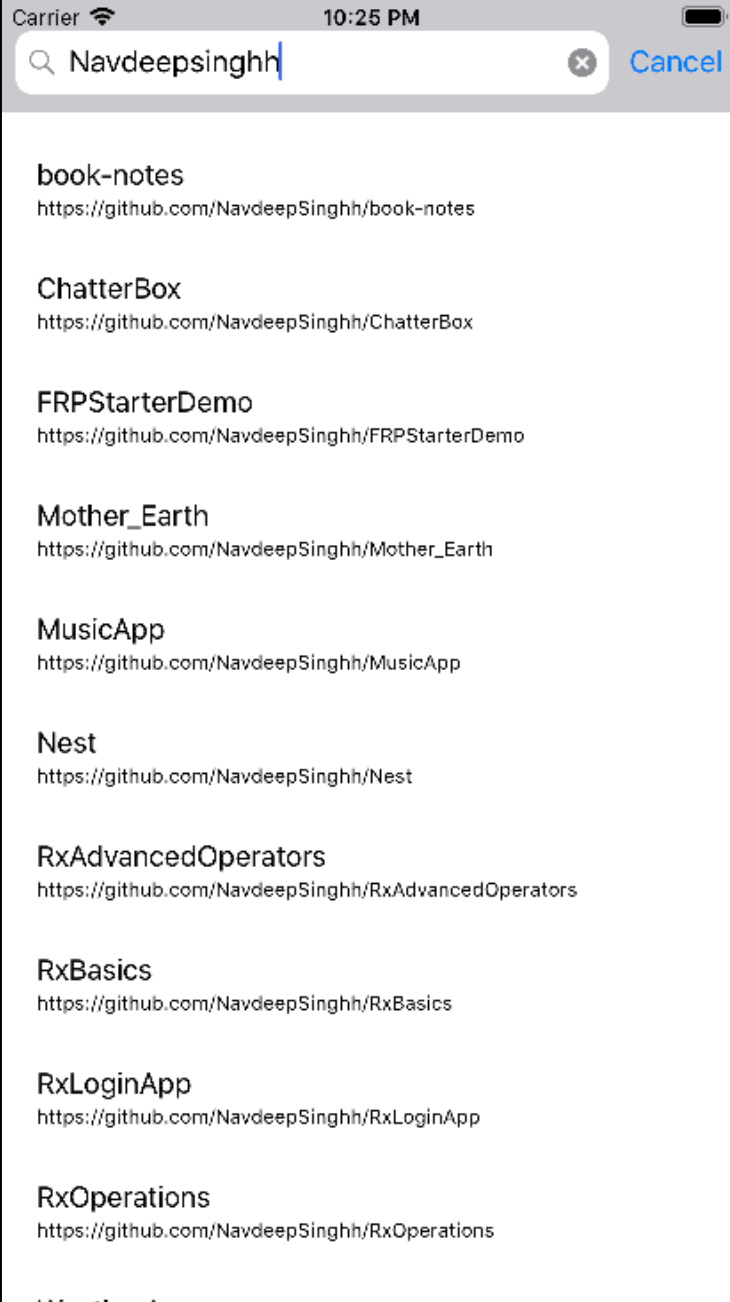


Table of Contents

Index

2

Index