GETTING UP TO SPEED WITH REACTIVECOCOA VERSION 4

Steffen D. Sommer (@steffendsommer)

iOS Developer at Unwire

Unwire

IS HIRING

YES THIS IS A

PINK

SLIDESHOW

YES RAC IS

HARD

- RxSwift
- ReactKit
- Bond
- Interstellar
- VinceRP

Functional reactive programming (FRP) is a programming paradigm for reactive programming (asynchronous dataflow programming) using the building blocks of functional programming (e.g. map, reduce, filter).

-- Wikipedia (April 23, 2016)

BUT WHAT DOES IT MEAN?

INPUTS

OUTPUTS

```swift
func isFormValid() -> Bool {
    return
        !(self.emailField.text?.isEmpty ?? true) &&
        !(self.usernameField.text?.isEmpty ?? true) &&
        !(self.passwordField.text?.isEmpty ?? true)

}

func textField(textField: UITextField,
shouldChangeCharactersInRange range: NSRange,
replacementString string: String) -> Bool {
    self.submitButton.enabled = isFormValid()

    return true
}
```

Example from NSHipster

```
self.submitButton.rex_enabled <~ combineLatest(
        self.emailField.rex_textSignal.producer,
        self.usernameField.rex_textSignal.producer,
        self.passwordField.rex_textSignal.producer)
    .reduce(true, { result, textfields in
        result && !textfields.0.isEmpty && !textfields.1.isEmpty && !textfields.2.isEmpty
    })
```

Example from NSHipster

THE SUPER EXCITING RELEASE STORY OF ReactiveCocoa

- Feb 14, 2013: ReactiveCocoa v.1.0.0
- Sep 15, 2013: ReactiveCocoa v.2.0.0
- Sep 12, 2015: ReactiveCocoa v.3.0.0 Swift 1.2
- Jan 28, 2016: ReactiveCocoa v.4.0.0 Swift 2.1.x

RAC2

↓

RAC4

RACSIGNAL

🔥

```
RACSignal *hot = RACObserve(self.emailTextfield, text);
```

⛄

```
RACSignal *cold = [RACSignal createSignal:
^RACDisposable *(id<RACSubscriber> subscriber) {
    NSLog(@"Doing some work..");
    [subscriber sendCompleted];
    return nil;
}];
```

SIGNAL 🔥

```swift
func createSignal() -> Signal<String, NoError> {
    var count = 0
    return Signal { observer in
        NSTimer.schedule(repeatInterval: 0.1) { timer in
            print("Emitting a next event")
            count += 1
            observer.sendNext("tick #\(count)")
        }
    }
}

let signal = createSignal()
```

Example from Colin Eberhardt's blog

SIGNALPRODUCER ☃️

```swift
func createSignalProducer() -> SignalProducer<String, NoError> {
    var count = 0
    return SignalProducer { observer, disposable in
        NSTimer.schedule(repeatInterval: 0.1) { timer in
            print("Emitting a next event")
            count += 1
            observer.sendNext("tick #\(count)")
        }
    }
}

let signalProducer = createSignalProducer()
signalProducer.start()
```

Example from Colin Eberhardt's blog

INTERRUPTED JOINS THE CLUB OF EVENTS

```swift
/// Represents a signal event.
///
/// Signals must conform to the grammar:
/// `Next* (Failed | Completed | Interrupted)?`
public enum Event<Value, Error: ErrorType> {
    /// A value provided by the signal.
    case Next(Value)

    /// The signal terminated because of an error. No further events will be
    /// received.
    case Failed(Error)

    /// The signal successfully terminated. No further events will be received.
    case Completed

    /// Event production on the signal has been interrupted. No further events
    /// will be received.
    case Interrupted

    ....
```

```swift
let signalProducer = createSignalProducer()
let disposable = signalProducer.startWithInterrupted {
    print("I just got interrupted 😑")
}
disposable.dispose()
```

```
func createSignalProducer() -> SignalProducer<String, NoError> {
    var count = 0
    return SignalProducer { observer, disposable in
        let timer = NSTimer.schedule(repeatInterval: 0.1) { timer in
            print("Emitting a next event")
            count += 1
            observer.sendNext("tick #\(count)")
        }

        disposable.addDisposable(timer.invalidate)
    }
}
```

PROPERTYTYPE OVER RAC AND RACOBSERVE

```
// SomeViewController.m

// Sync the title of the view.
RAC(self, title) = RACObserve(self.viewModel, title);


// SomeViewModel.m
@property (nonatomic, strong, readonly) NSString *title;
```

```swift
/// Represents a property that allows observation of its changes.
public protocol PropertyType {
    associatedtype Value

    /// The current value of the property.
    var value: Value { get }

    /// A producer for Signals that will send the property's current value,
    /// followed by all changes over time.
    var producer: SignalProducer<Value, NoError> { get }

    /// A signal that will send the property's changes over time.
    var signal: Signal<Value, NoError> { get }
}
```

- AnyProperty
- ConstantProperty
- MutableProperty
- DynamicProperty

```swift
// SomeViewModel.swift
let headline = ConstantProperty<String>("Today's Menu")
private let menu = MutableProperty<Menu?>(nil)
private let mutableMainCourse = MutableProperty("")
var mainCourse : AnyProperty<String>!

init() {
    self.mainCourse = AnyProperty(self.mutableMainCourse)
}


// SomeViewController.swift
private let headline = UILabel()
private let mainCourse = UILabel()
...
self.headline.rac_text <~ self.viewModel.headline.signal.observeOn(UIScheduler())
self.mainCourse.rac_text <~ self.viewModel.mainCourse.signal.observeOn(UIScheduler())
```

RACCOMMAND

= ACTION

.. MORE OR LESS

```
enum Vote {
    case Up
    case Down
}

let voteAction = Action<Vote, String, NoError> { test in
    // Call some web API ..
    return self.createSignalProducer()
}

self.upVoteButton?.rex_pressed.value = CocoaAction(voteAction, input: (.Up))
```

-flatten
-flattenMap:
+merge:
-concat
+concat:
-switchToLatest

- flatten
- flatMap

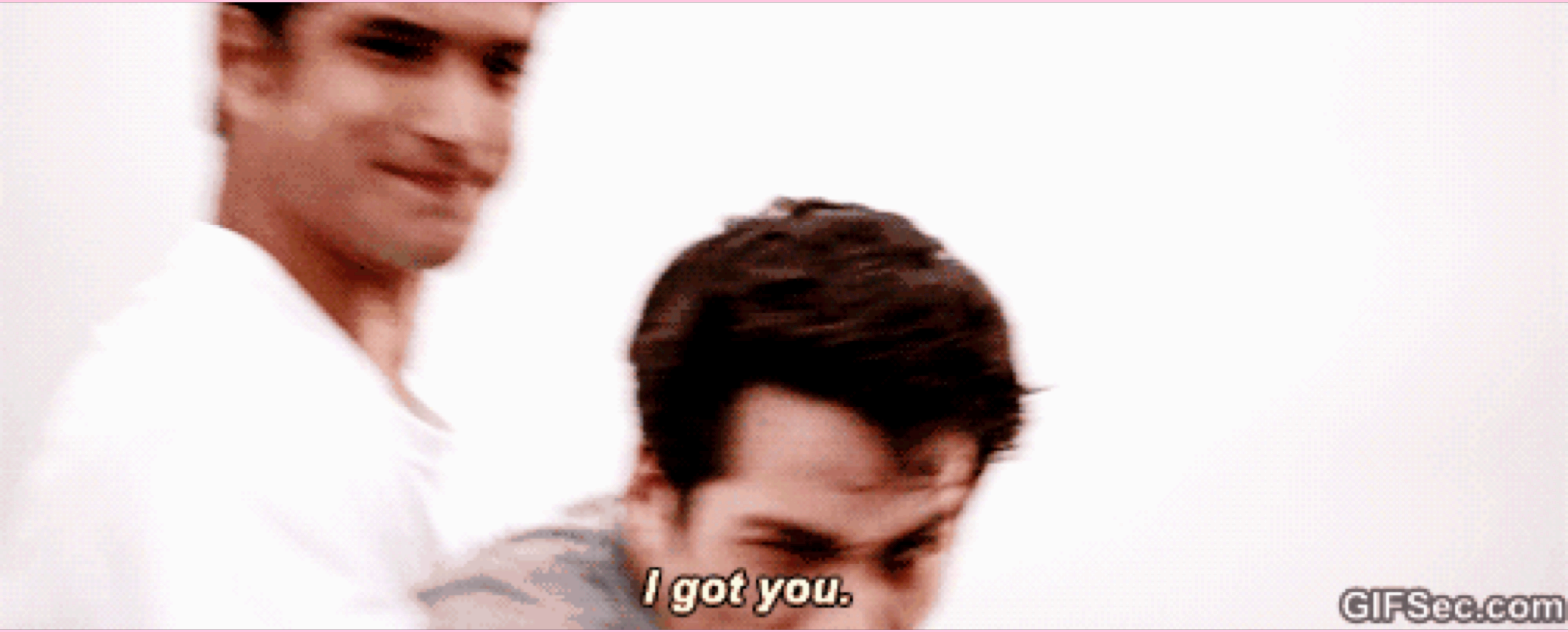
RAC4 ♥ SWIFT

```
/**
 *  A signal that will send next values (as strings) every time some text changes.
 *
 *  @return A signal that sends the current text as a NSString on next.
 */
- (RACSignal *)textChanged;
```

NO MORE ID MADNESS

PARAMETERIZATION FTW

```
Signal<Int, NSError>
Signal<String, NoError>
```


I got you.
GIFSec.com

RAC4 DOES NOT COME WITH UI BINDINGS 😢

REX → RAC.ORG[1]

[1] #2790

BUT WHAT ABOUT OBJECTIVE-C?

- RACSignal.toSignalProducer()
- toRACSignal()
- RACCommand.toAction()
- toRACCommand()

SHOW ME THE CODE

CAKE DAY

Today's Menu

at
Unwire

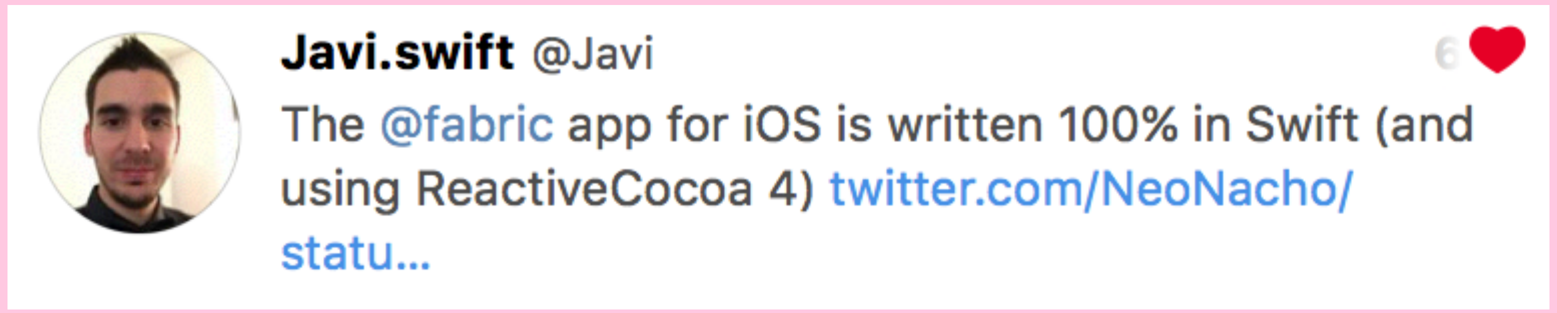
Æggekage med kartofler, løg, tomater og bacon.

Rødbeder med solsikkekerner, balsamico og persille. Broccoli og cherrytomater. Blandet salat med dressing. Dagens råkost med rosiner. Æg, rejer, mayonnaise og asparges. Leverpostej med ristet bacon, champignoner og surt. Pålægsfad med afskåret pålæg, hjemmelavede salater samt. 3 slags ost. Friskbagt solsikkerugbrød, maltbrød og græskarkernebrød. Frisk frugt.

```swift
// SomeViewModel.swift

self.headline <~ self.menu.producer
    .ignoreNil()
    .map { fetchedMenu in
        if (fetchedMenu.isTodaysMenu()) {
            return fetchedMenu.mainCourse!
        } else {
            return "The chef is working hard on getting Today's Menu ready. Please come back later."
        }
    }
    .flatMapError { _ in
        return SignalProducer<String, NoError>(value: "Something went wrong in the kitchen. Please come back later.")
    }
```

```swift
// SomeArchivableProtocol.swift

static func loadUsingIdentifier(identifier: String) -> SignalProducer<Self, ArchivableError> {

    return SignalProducer { observer, disposable in
        guard let data = NSUserDefaults.standardUserDefaults().objectForKey(identifier) as? [String: AnyObject] else {
            observer.sendFailed(ArchivableError.ValueNotFound)
            return
        }

        observer.sendNext(data)
        observer.sendCompleted()
    }
    .flatMap(.Latest) { value in
        return Self.deserializeFromJSON(value)
            .mapError { error in
                return ArchivableError.LoadFailed
            }
    }

}
```

CAN I USE IT?



Javi.swift @Javi

6

The @fabric app for iOS is written 100% in Swift (and using ReactiveCocoa 4) twitter.com/NeoNacho/statu…

#2697

- The current version (4.1) is stable
- Already used in production apps
- The team is working hard on making it more user-friendly
- The list of resources is growing
- I'm personally starting to feel more comfortable using it

- The best FRP iOS resources
- Rex
- TodaysReactiveMenu
- ReactiveCocoa's Changelog

Thanks