

# ReactiveCocoa Developers Conference (RACDC)

@jsphrsummers

ReactiveCocoa  
“Statelessness”  
of the Union

What is  
ReactiveCocoa?

Not just KVO

Not just bindings

Not just futures

# signals

unify all of these patterns

(and more)

# Problems with ReactiveCocoa 2



**WHAT'S IN THE BOX?!?!**

# Katy Perry

## HOT N COLD



Subjects  
Multicasting  
Replaying

# RAC Command

# ReactiveCocoa has too much magic!

– *the internet*



**SWAY**

Wild FEEL

appeared!

# Parameterized types (generics)

# No macros

# Less dynamic programming



# ReactiveCocoa 3



**Signal<T, E>**  
(parameterized values *and* errors)

Signal<String, CarthageError>

Signal<Int, NoError>

signals

and

Signal Producers

# Signals and Signal Producers

```
let producer = timer(1, onScheduler: QueueScheduler())  
  
let disposable = producer.start(next: { date in  
    println("First timer fired at \(date)")  
})  
  
producer.startWithSignal { signal, signalDisposable in  
    signal.observe(next: { date in  
        println("Second timer fired at \(date)")  
    })  
}  
}
```

# Signals and Signal Producers

```
let producer = timer(1, onScheduler: QueueScheduler())  
  
producer.startWithSignal { signal, signalDisposable in  
    signal.observe(next: { date in  
        println("Timer fired at \(date)")  
    })  
  
    signal.observe(anotherObserver)  
}
```

# Action

(instead of RACCommand)

# Actions

```
let searchAction = Action(enabledIf: hasText) { text in
    let resultsProducer = APIClient.searchText(text)
    return resultsProducer
}

searchAction.values.observe(next: { result in
    println("Search result: \(result)")
})

let searchCatsProducer = searchAction.apply("cats")
searchCatsProducer.start()
```

# PropertyType

# PropertyType

```
let text = MutableProperty("0")

text.producer.start(next: { string in
    println("Property value is now: \(string)")
})

text <~ timer(1, onScheduler: QueueScheduler())
    |> scan(0) { counter, _ in counter + 1 }
    |> map { counter in toString(counter) }
```

The theme is

simplicity

**EASY<sup>1</sup>**  
familiar  
approachable

---

<sup>1</sup> See Rich Hickey's talk, "Simple Made Easy"



# SIMPLE<sup>1</sup>

separate concerns  
less complex

---

<sup>1</sup> See Rich Hickey's talk, "Simple Made Easy"



ReactiveCocoa 2 is neither

easy

nor

simple

ReactiveCocoa 3 is

simple

(hopefully)

# Dr. Changelog, MD

– *@joshaber*

# Remaining work for 3.0

- **Long-form documentation**

# Contributors

- **Dave Lee** (@kastiglione)
- **Andy Matuschak** (@andymatuschak)
- **Nacho Soto** (@NachoSoto)
- **Javi Soto** (@Javi)
- **Syo Ikeda** (@ikesyo)
- **Neil Pankey** (@neilpa)
- ... many more!

# Interested in contributing?

Join Slack by emailing [justin@jspahrsummers.com](mailto:justin@jspahrsummers.com)