



RxSwift (1/3)

2017.8.25 キムギボン

目次

1.RxSwiftとは？

- 始め
- 現況
- 概要
- なぜ導入する？

2.既存プログラミング概念を捨てる！

3.MVCとは？ MVVMとは？

4.実習





始め

実は最近出た新しい概念ではない

2009.11.17 MSの.NETのReactive Extensions for .NETがその始め

その後他のプラットフォームからも導入されました。

全てプロジェクトはオープンプロジェクトで進まれています。


GitHub: <https://github.com/ReactiveX>

公式サイト: <http://reactivex.io>



現況

RxSwift以外にも現在38個のRxプロジェクトが進んでいる🤖



ReactiveX

Reactive Extensions for Async Programming

<http://reactivex.io>

Repositories 38

People 17

Type: All ▼ Language: All ▼

RxKotlin

RxJava bindings for Kotlin


kotlin rxjava

Kotlin

★ 2,563

🔗 165

Updated an hour ago



RxSwift

Reactive Programming in Swift


swift ios reactivex reactive unidirectional observer

Swift

★ 10,498

🔗 1,605

Updated 3 hours ago



Top languages

Java

JavaScript

Ruby

Groovy


TypeScript

Most used topics

reactivex rxjava java reactive reactive-streams

People

17 >





概要

- Reactive Extensions for Async Programming 🤔 ?
- 🤔 既存の命令形プログラミングの反対概念
- Observable(Signal)概念で通信する 🤔 🤔 🤔
 - KVO、Closureと似た部分が多い 😓



なぜ導入する？

- 最近みんな使おうとしているから？ 😅
- 複雑はロジックや処理がシンプルになる
- MVVMアーキテクチャ導入でソースの構造も明確になり、把握しやすくなる



上記の全ての話はRxSwiftになれば～



既存プログラミング概念を捨てる！🧐

- 全てはObservableとその購読(subscribe)で行われる
- そのためにはどのようにObservable化するかが大事！

```
Observable.of("A", "B", "C")  
  .subscribe({ (event) in  
    print(event)  
  })  
  .dispose()
```

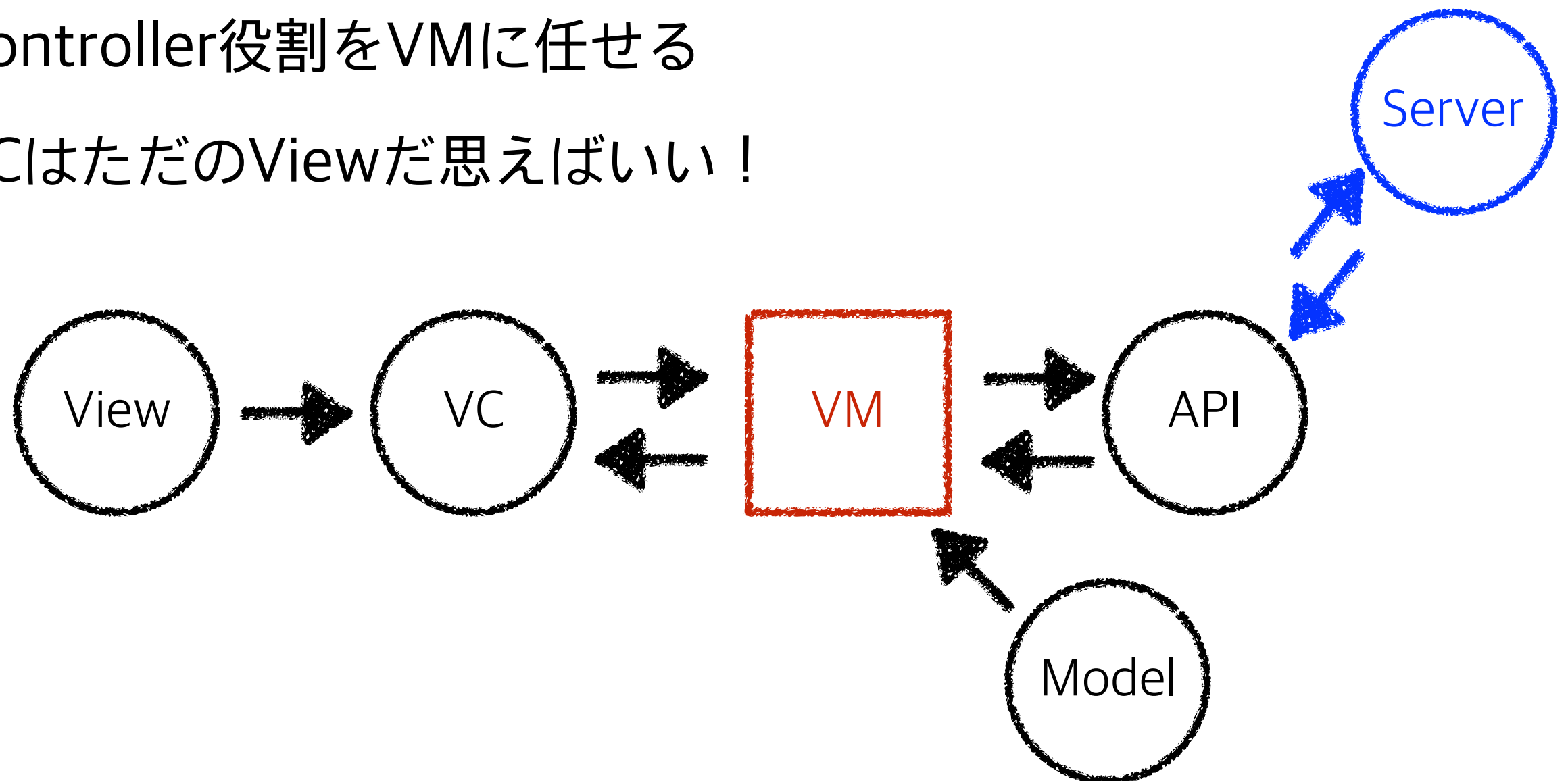


MVVMとは？

- MVC → MVVM
- 明確なReactiveプログラミングのため既存VCが持ってた

Controller役割をVMに任せる

- VCはただのViewだ思えばいい！





実習🤖

- RxSwiftインストール😐
- Observableを生成してみる😮
- 購読(Subscribe)を試してみる😮
- Disposeを試してみる😵
- 他の形のObservableを知っておく😬
- RxCocoaを使ってみよう！🧐
- MVVMで簡単に実装してみる👉