

# Advanced RxSwift – Day 3

Younghwan Kim



## RxSwift Basics

- Day 1 – Observable, Operator (Filter, Transform, Combine)
- Day 2 – Subject (flatMap, flatMapFirst, flatMapLatest)
- Day 3 – Two VCs communications with Subject, RxCocoa (Button)
- Day 4 – Sequential, Merged Observable Calls
- Day 5 – RxCocoa, UI Binding (Button, TextField, Label, TableView)



## Advanced RxSwift

- Day 1 – Protocol-Oriented Programming, Protocol Extension, AssociateType
- Day 2 – Network Call, Generic Enum
- **Day 3 – Binding Track Activity (show / hide ‘Loading’ )**
- Day 4 – Advanced TableView - RxDataSources
- Day 5 – Schedulers (observeOn, subscribeOn),  
Unit Test (RxTest, RxBlocking)



# ActivityIndicator

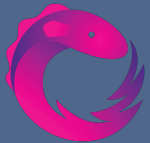
<https://github.com/ReactiveX/RxSwift/blob/master/RxExample/RxExample/Services/ActivityIndicator.swift>

```
public class ActivityIndicator : SharedSequenceConvertibleType {  
  
...  
  
}
```



# ActivityIndicator in ViewModel

```
class SimpleViewModel {  
    // Is signing process in progress  
    let signingIn: Observable<Bool>  
    let signingInIndicator = ActivityIndicator()  
  
    init() {  
        self.signingIn = signingInIndicator.asObservable()  
    }  
  
    func simpleObservable() -> Observable<String> {  
        return Observable<String>.create { observer in  
            DispatchQueue.main.asyncAfter(deadline: .now() + 5) {  
                observer.onNext("strawberry")  
                observer.onCompleted()  
            }  
            return Disposables.create()  
        }  
    }  
}
```



# ActivityIndicator : UI Binding 1

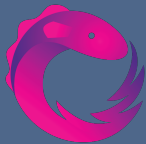
```
class ViewController: UIViewController {
    @IBOutlet weak var TrackActivityOutlet: UIActivityIndicatorView!
    @IBOutlet weak var backgroundView: UIView!
    @IBOutlet weak var trackActivityButton: UIButton!
    var disposeBag = DisposeBag()

    override func viewDidLoad() {
        super.viewDidLoad()

        let viewModel = SimpleViewModel()

        viewModel.signingIn
            .bind(to: TrackActivityOutlet.rx.isAnimating)
            .disposed(by: disposeBag)

        viewModel.signingIn
            .map { !$0 }
            .bind(to: backgroundView.rx.isHidden)
            .disposed(by: disposeBag)
    }
}
```



## ActivityIndicator : UI Binding 2

```
viewModel.signingIn
    .bind(to: trackActivityButton.rx.isHidden)
    .disposed(by: disposeBag)

trackActivityButton.rx.tap.asDriver()
    .drive(onNext: { _ in
        viewModel.simpleObservable()
            .observeOn(MainScheduler.instance)
            .trackActivity(viewModel.signingInIndicator)
            .subscribe(onNext: { _ in
                })
            .disposed(by: self.disposeBag)
    }).disposed(by: disposeBag)
}
```



Add ActivityIndicator in Borders App in day 2