

# Advanced RxSwift – Day 2

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 – Adding a Reactive Extension to Custom UI Element,  
2 Way Binding, Advanced TableView – RxDataSources
- Day 5 – Schedulers (observeOn, subscribeOn),  
Unit Test (RxTest, RxBlocking)



## Integrate with API Call

- **API Call (create Observable)**
- **Flow Control in API calls ( filter, flatMap operation in Observables)**
- **Create View Model ( BehaviorSubject )**
- **Data Model ( demo 'Quick Type')**
- **UI Binding (TableView and ViewModel)**



# Integrate with API Call

## API Call (create Observable)

### Country Info

<https://restcountries.eu/rest/v2/alpha/DEU>

### Border Countries

<https://restcountries.eu/rest/v2/alpha?codes=DEU;FRA;ITA;>



## Integrate with API Call

```
func contentRequestCall(urlString: String) -> Observable<Result<Data, String>> {  
    guard let urlRequest = contentURLRequest(urlString: urlString) else {  
        return Observable<Result<Data, String>>.just(.failure(nil))  
    }  
    return Observable.create { observer in  
        let task = URLSession.shared.dataTask(with: urlRequest) { (data, response, error) in  
            guard error == nil, let data = data else {  
                observer.onNext(.failure(error?.localizedDescription))  
                observer.onCompleted()  
                return  
            }  
            observer.onNext(.success(data))  
            observer.onCompleted()  
        }  
        task.resume()  
        return Disposables.create{  
            task.cancel()  
        }  
    }  
}
```



## ▬ Generic Enum (Video)



## Error Handling with Result Type

- Error Handling with Result Type (Video)





## Integrate with API Call

**Generic Enum : Result in Alamofire**

<https://github.com/Alamofire/Alamofire/blob/master/Source/Result.swift>



10