



iOS Seminar 5

한상현

오늘 배울 내용

- Pagination
- RxSwift
- MVVM : Model - View - ViewModel 로 구성된 Architecture
- Login (Authentication) + JWT (Access Token, Refresh Token)

Pagination

- 우리가 서버로부터 데이터를 많이 받아올 때 한 번에 받아온다면??
-> 로딩 작업하는 동안 UI는 멈춰있게 된다
- 모든 데이터가 다 필요할지라도 조금씩 쪼개서 받아와야할 필요성 존재
-> 서버에서도 페이지 단위로 데이터를 쪼개서 내려주기로 약속
: Pagination

Pagination의 방법론

1. 내가 현재 가진 데이터의 마지막이 보였는가?
2. 데이터가 올려진 ScrollView가 끝까지 스크롤되었는가?

```
extension MovieCollectionViewController: UICollectionViewDataSourcePrefetching {  
    func collectionView(_ collectionView: UICollectionView, prefetchItemsAt indexPaths: [IndexPath]) {  
    }  
}  
  
extension MovieCollectionViewController: UIScrollViewDelegate {  
    func scrollViewDidScroll(_ scrollView: UIScrollView) {  
    }  
}
```

Xcode로 고고!



예시

- <https://medium.com/@ferhanakkan/how-to-use-pagination-in-rxswift-5aaf0283792e>

RxSwift

- 이벤트의 송출과 구독
 - Publish / Subscribe
 - Observable -> Subscribe -> Dispose
-
- 항상 중요한건 데이터(이벤트)가
어떻게 흘러가고, 그걸 누가 수정하고, 누가 구독해서 사용하는가!

Observable / Single

- 데이터를 한 번만 받아서 publish하는가? 아니면 데이터가 여러 차례 내려오는가?
- 사실 observable로 다 써도 무방하지만 코드의 의도를 잘 드러내기 위해 여러 trait들이 존재하는 것

<https://sodp2.tistory.com/14>

Subject / Relay

- Subject : complete, error가 있음
 - Relay : complete, error가 없고 dispose 되기 전까지 작동
-
- 이 아이들은 pagination과 같이 값이 들어있으나 중간에 수정이 필요할 경우에 많이 사용 (유저의 액션에 따라 추가적인 데이터를 넣거나 빼거나 하는 경우 : pagination)
 - 변수처럼 현재 저장되어있는 값을 그냥 가져다 쓸 수도 있음 (value)

<https://joycestudios.tistory.com/88>

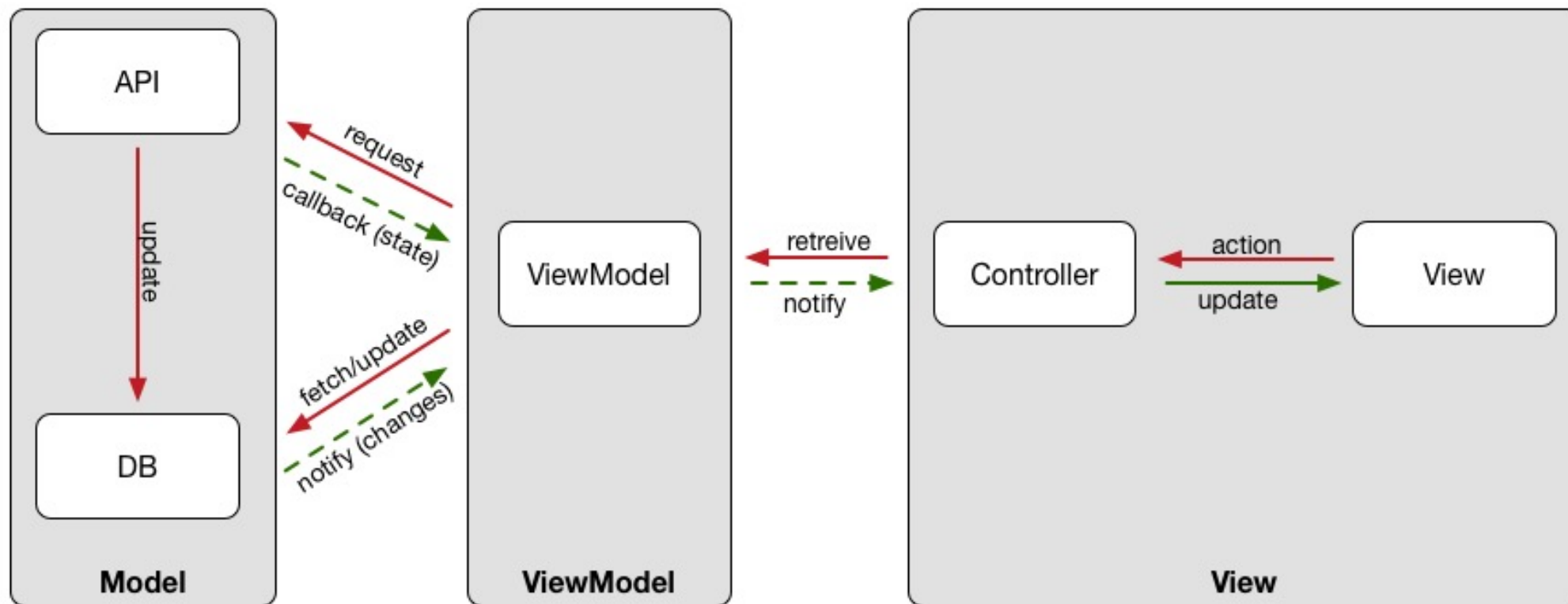
Rx의 다양한 활용

- Textfield 입력값의 변화 / 각종 Gesture 인식 등 UIKit과의 간편한 연계
- viewDidLoad, viewWillAppear와 같은 View Lifecycle을 Wrapping해 구독 가능
- 여러 이벤트를 병합하는 것도 가능 / 하나의 이벤트를 공유해서 여러 곳에서 구독하는 것도 가능
- 추가적으로 보면 좋은 내용
 - Hot Observable vs Cold Observable
 - Combine
 - 각종 mapping 메소드 (map, filter, flatmap 등)

Rx 관련 보면 좋은 자료 링크

- <https://devmjun.github.io/archive/Traits>
- <https://brunch.co.kr/magazine/reactivex>
- <https://rxmarbles.com/>
- **항상 공식문서를 유심히 뜯어보자!**

MVVM이란?



MVVM의 장점과 단점

- 장점

1. ViewController에 집중되어 있던 로직이 분산된다
: UI 요소는 ViewController에, 비즈니스 로직은 ViewModel에
2. 잘 binding 하면 데이터의 흐름이 눈에 잘 보인다 -> 디버깅, 리팩토링 용이

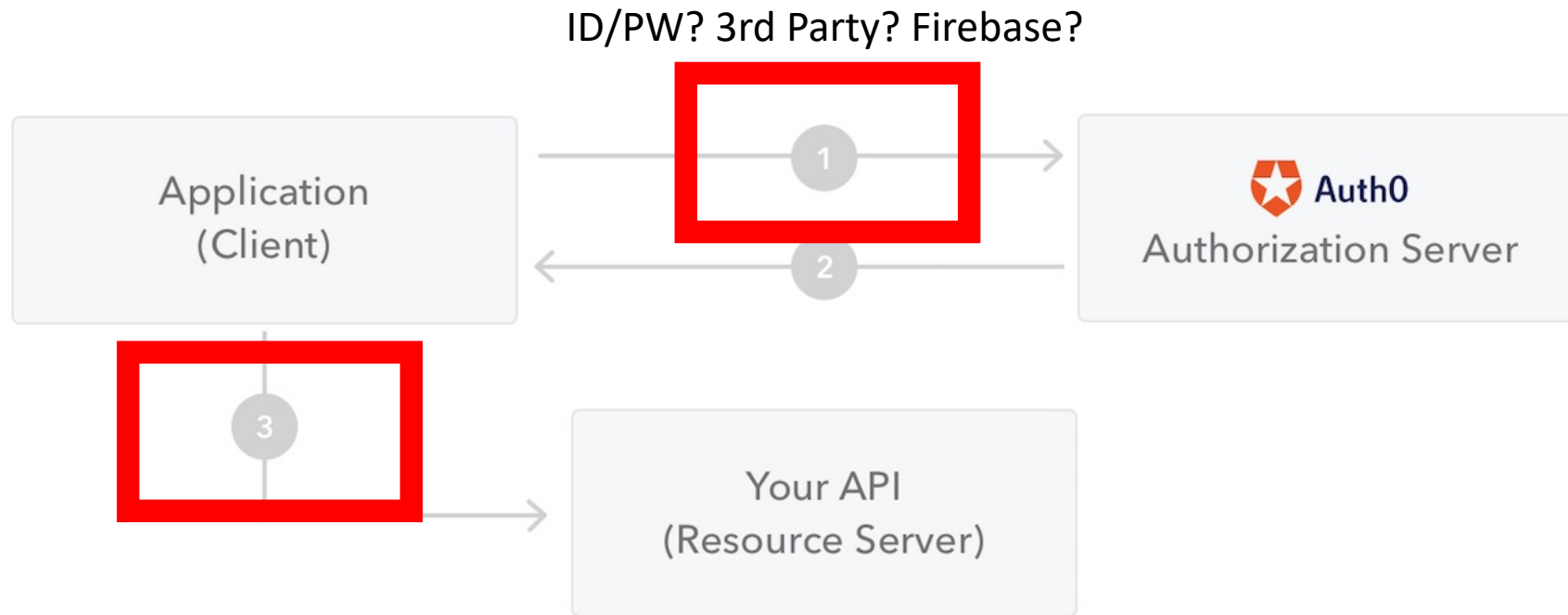
- 단점

1. 헷갈린다 (이거 VC에 들어가야하나 VM에 들어가야 하나?)
2. MVVM에 모든 걸 담을 수 없어 결국 다른 객체가 필요하고, 아키텍처는 어차피 복잡해진다
3. 짤 때 신경 쓸게 조금 더 많아진다?

Xcode로 고고!



로그인 (Authentication)



JWT란?

- <https://jwt.io/introduction>
- Auth 정보를 담은 토큰
 1. 로그인하면서 서버로부터 access token을 받는다
 2. 서버와의 규약에 맞춰 암호화된 jwt를 생성한다
 3. Auth 정보가 필요한 API를 요청할 때 header로 실어보낸다
(token이 없거나, 만료되면 reponse가 에러로 떨어질 것!)

Authentication Header

1. Request를 생성하면서 header를 만들어 넣어주는 방법
2. Adapt 메소드에서 request를 잠깐 가로채 header를 주입해주는 방법

<https://github.com/Alamofire/Alamofire/blob/master/Documentation/AdvancedUsage.md#request-adapter>

```
let accessToken: String

func adapt(_ urlRequest: URLRequest, for session: Session, completion: @escaping (Result<URLRequest, Error>) -> Void) {
    var urlRequest = urlRequest
    urlRequest.headers.add(.authorization(bearerToken: accessToken))

    completion(.success(urlRequest))
}
```

앞으로 여러분들이 공부해보셔야 할 내용

- ARC (Auto Retain Cycle) -> Memory Leak 방지
- View Lifecycle : View는 어떻게 생성되고 어떻게 유저에게 보여지는가
- App Lifecycle : App은 어떻게 관리되는가
- Window의 개념 : Window는 무엇이고, View는 Window에 어떤 식으로 올라가는 것일까?
- Architecture : MVP, MVC, MVVM, VIPER, RIPS 등의 App 아키텍처들은 어떤 원리로 동작하며 왜 고안되었는가?
- CI / CD : 배포, 테스트 자동화
- 테스트 작성, TDD : 유닛 테스트의 작성을 통한 정상 동작 보장 + 리팩토링의 수월한 진행, 테스트를 먼저 작성하면서 얻는 설계적 이득

Assignment 5 : 영화 소개 앱 만들기

0. 로그인 뷰 (옵션)

- 1) TMDB 로그인 창을 구현한다 (username, password 입력)
- 2 - 옵션 1) 데이터를 바탕으로 request token을 만들고, 이를 이용하여 Access Token을 발급받아 auth를 세팅한다 (adapt 이용)
- 2 - 옵션 2) 데이터를 바탕으로 request token을 만들고, 이를 이용하여 Session ID를 발급받아 API 호출에 활용한다

Assignment 5 : 영화 소개 앱 만들기

- 메인 화면에는 두 개의 탭이 존재

1. Movie 탭 (1 탭)

- 1) 지난 과제 프로젝트에 이어서 구현하시면 됩니다
 - 2) 동일한 스펙에서 영화를 선택했을 때 세부정보 뷰에서 추가 스펙 존재
- * 세부 정보 오른쪽 위에는 별 버튼이 있어 누르면 POST Mark as Favorite을 이용해 관심 영화 등록 / 해제를 한다.

Assignment 5 : 영화 소개 앱 완성

2. Favorite 탭

- 1) 타이틀은 “관심 영화” / Navigation Bar 넣어도 되고 View로 구현해도 됨 (자유)
- 2) 지난 과제4와 같이 collectionView는 2열로 구성되어 있으며, 관심 영화의 정보가 화면에 표시됨 (포스터 + 제목 + 평점)
- 3) 데이터는 반드시 Pagination을 통해 추가로 로드하고, collectionView를 갱신함
- 4) 영화를 선택하면 Custom Animation이 적용된 UINavigationController의 push로 세부 정보 뷰가 뜸 (포스터 + 제목 + 평점 + overview)
- 5) 세부 정보 오른쪽 위에는 별 버튼이 있어 favorite 여부 체크할 수 있음 (여기는 관심 영화들만 모아놓은 View니까 처음에 다 체크가 되어있겠죠?)
- 6) 만약 체크를 해제하고 뒤로 돌아오게 되면 그 영화가 관심 영화 목록에서 빠져야 있어야한다!

Assignment 5 : 영화 소개 앱 완성

3. 설정 탭

- 1) 셀은 하나
- 2) 그 셀을 탭하면 로그아웃 여부를 묻는 팝업이 뜨고, 확인을 누르면 로그인 화면으로 돌아간다

Assignment 5 : 영화 소개 앱 완성

- 이 과제의 중요 달성 조건

1. Favorite 탭에 정상적으로 관심 영화가 불러와지는가?
2. 영화 세부정보에서 관심 영화 여부를 변경했을 때 Favorite 탭에 잘 반영되는가?

- 이 과제의 세부 달성 조건 (안해도 된다는 뜻)

1. 로그인뷰, 3탭 만들기
2. Session ID가 아닌 Access Token으로 GET Favorite Movies 호출하기

-> 세부 조건을 수행하지 않고 중요 조건을 수행하려면 api key를 이용해 session id를 발급 받아서 API들을 호출하면 됩니다

(이왕이면 Access Token 써보는 연습을 해보는 걸 추천합니다 : 토이 플젝에서 무조건 쓰게 되어있음)