



URLsession + Firebase

넷플릭스 스타일 영상 앱 제작

Presenter Ye Jin-hee

INDEX

URLsession +
Firebase 넷플릭스
스타일 영상 앱

제작개요 01

개발 및 구현 02

영상 및 관리 03

논의 및 고찰 04

01. 제작개요



URLsession을 이용한 데이터 검색,
컬렉션 뷰에 결과 업데이트



Firebase를 통해 검색
이력을 서버와 연동



플레이어뷰 UI구성하여 검색결과
클릭시 가로화면으로 영상 자동재생

02. 개발 및 구현

1 메인화면

1 재생버튼 클릭시 동작

1. 영화제목으로 검색 결과(movies)를 interstellar 상수에 할당
2. 뷰 컨트롤러의 preview url를 interstellar preview URL로 교체 및 재생

2 재생 시 모드가 가로, 우측만 되도록 설정

```
1 HomeViewController.swift -----
2 1 Action func playButtonTapped(_ sender: Any) {
3     // [x] interstellar에 대한 정보를 검색 API로 가져옴
4     // [x] 거기서 영화 객체 하나를 가져옴
5     // [x] 객체를 이용해서 playviewController를 띄운다.
6
7     SearchAPI.search("interstellar") { movies in
8         guard let interstellar = movies.first else { return }
9         DispatchQueue.main.async {
10             let url = URL(string: interstellar.previewURL!)
11             let item = AVPlayerItem(url: url)
12             let sb = UIStoryboard(name: "Player", bundle: nil)
13             let vc = sb.instantiateViewController(identifier: "PlayerViewController")
14             vc.player.replaceCurrentItem(with: item)
15
16             vc.modalPresentationStyle = .fullScreen
17             self.present(vc, animated: false, completion: nil)
18         }
19     }
20 }
21
22 // PlayerViewController.swift -----
23 class PlayerViewController: UIViewController {
24     // ... 생략 ...
25     let player = AVPlayer()
26 2 override var supportedInterfaceOrientations: UIInterfaceOrientationMask {
27         return .landscapeRight
28     }
29 }
```



02. 개발 및 구현

2 검색화면

1 컬렉션 뷰 셀 크기정의

1. N행 3열
2. 높이 비율 : 7:10

2 검색결과

1. 검색결과를 movies 오브젝트로 설정
2. Collectionview 데이터를 reload
3. Firebase 서버에 검색어, 검색시간을 기록



```
1 extension SearchViewController: UICollectionViewDelegateFlowLayout {
2     func collectionView(_ collectionView: UICollectionView,
3         layout collectionViewLayout: UICollectionViewLayout,
4         sizeForItemAt indexPath: IndexPath) -> CGSize {
5
6         let margin: CGFloat = 8
7         let itemSpacing: CGFloat = 10
8         let width = (collectionView.bounds.width - margin * 2 - itemSpacing * 2) / 3
9         let height = width * 10/7
10        return CGSize(width: width, height: height)
11    }
12 }
13
14 2 func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {
15     dismissKeyboard() // 키보드 올라와 있을 때 내려가게 처리
16     guard let searchTerm = searchBar.text, searchTerm.isEmpty == false else { return }
17
18     // 네트워크를 통한 검색
19     // - 목표: 서치함을 가지고 네트워크를 통해서 영화 검색
20     // - [x] 검색API가 필요
21     // - [x] 결과를 받아올 모델 Movie, Response
22     // - [x] 결과를 받아와서, collectionView로 표현
23
24     SearchAPI.search(searchTerm) { movies in
25         // [x] collectionView로 표현
26         print("----> 몇개? \(movies.count), 첫번째 제목: \(movies.first?.title)")
27         DispatchQueue.main.async {
28             self.movies = movies
29             self.resultCollectionView.reloadData()
30
31             let timestamp: Double = Date().timeIntervalSince1970.rounded()
32             self.db.childByAutoId().setValue(["term": searchTerm, "timestamp": timestamp])
33         }
34     }
35     print("----> 검색어: \(searchTerm)")
36 }
```



02. 개발 및 구현

3 Nested ScrollView 구현

각 뷰컨트롤러 연결

- 1 세그웨이 식별자 일치시 해당 추천 뷰 컨트롤러로 형 변환
- 2 식별자로 타입 업데이트 및 아이템 fetch

```
1 class UpComingViewController: UIViewController {
2     var awardRecommendListViewController: RecommendListViewController!
3     var hotRecommendListViewController: RecommendListViewController!
4
5     override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
6         if segue.identifier == "award" {
7             let destinationVC = segue.destination as? RecommendListViewController
8             awardRecommendListViewController = destinationVC
9             awardRecommendListViewController.viewModel.updateType(.award)
10            awardRecommendListViewController.viewModel.fetchItems()
11        } else if segue.identifier == "hot" {
12            let destinationVC = segue.destination as? RecommendListViewController
13            hotRecommendListViewController = destinationVC
14            hotRecommendListViewController.viewModel.updateType(.hot)
15            hotRecommendListViewController.viewModel.fetchItems()
16        }
17    }
18    // ... 생략 ...
19 }
```

1~2 설명과 동일한 동작



03. 영상 및 형상관리

슬라이드 쇼 중일 때 이미지가 재생됩니다.

GitHub QR-Code



논의 및 고찰

앱 기능 확장 및 구현

기존 스토어에 있는 앱과 사용자 가치를 고려한 개선된 기능을 구현한다면 사용성이 더욱 높아질 것이다.

(사용자 취향과 비슷한 영상을 추천하는
기능 외에 기피하는 장르 비 추천 등)