



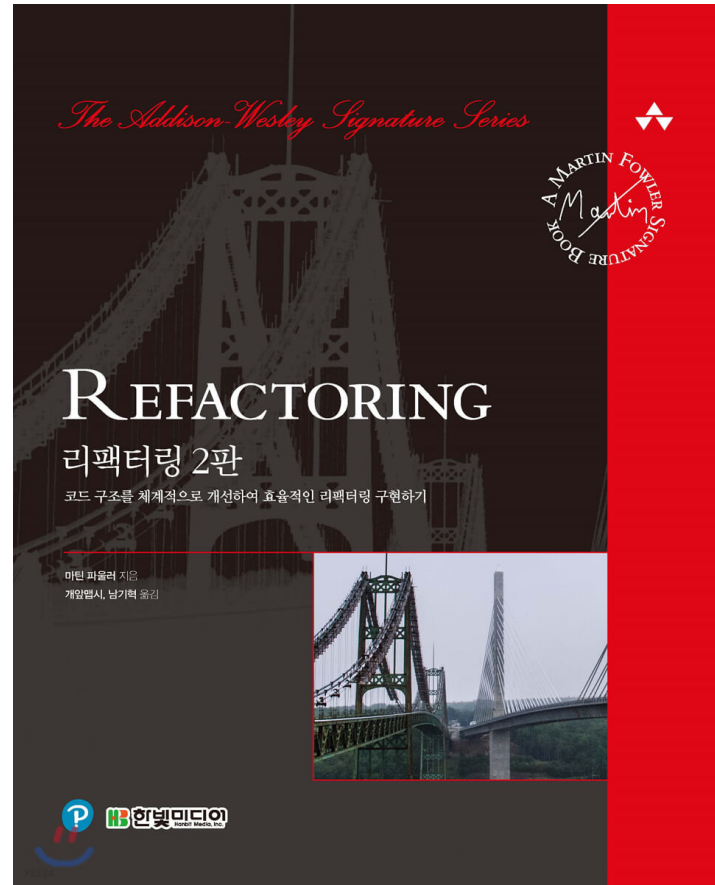
iOS Seminar 5

한상현

오늘 배울 내용

- 리팩토링 이어서 (의존성 역전 원칙 실습)
- 로그인의 원리 (Bearer Token, API key, OAuth Login 등)
- 앞으로 여러분이 해야 할 것들

리팩토링 (Refactoring)



의존성이란?

- “A가 B를 의존한다.” -> 의존대상 B가 변하면, 그것이 A에 영향을 미친다.

Ex)

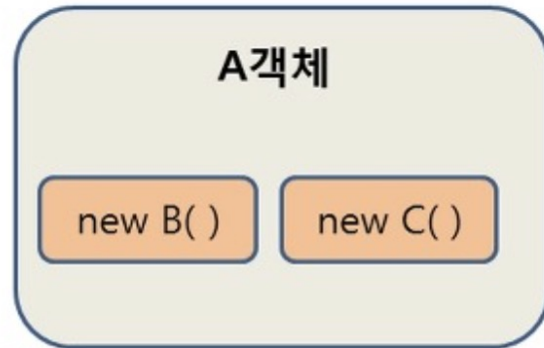
햄버거 가게 요리사는 햄버거 레시피에 의존한다.

햄버거 레시피가 변화하게 되었을 때, 변화된 레시피에 따라서
요리사는 햄버거 만드는 방법을 수정해야 한다.

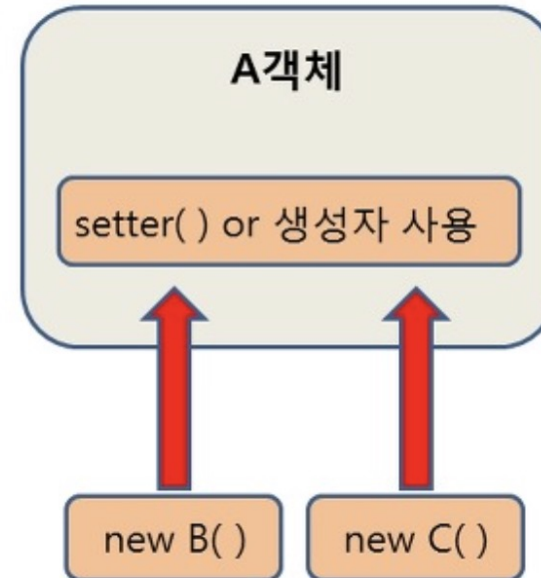
레시피의 변화가 요리사의 행위에 영향을 미쳤기 때문에,
“요리사는 레시피에 의존한다”고 말할 수 있다.

의존성 주입 (Dependency Injection : DI)

방법 1



방법 2



의존관계 역전 원칙 (Dependency Inversion Principle)

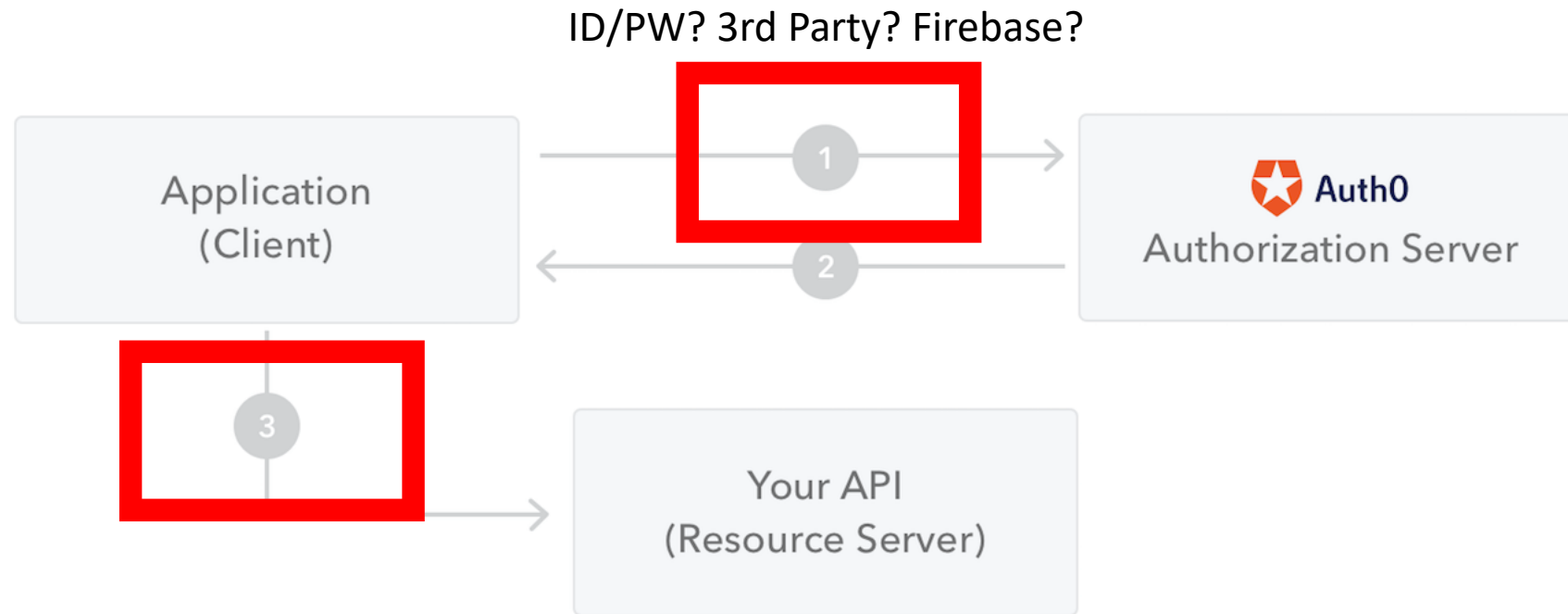
- SOLID (객체지향 설계)의 5가지 원칙 중 하나
- A가 B를 의존할 때, B는 실제 객체가 아닌 이를 추상화한 인터페이스 (protocol) 이어야 한다.
- B protocol을 실제로 구현하는 저수준 객체 (더 세부적이고 사소한(?) 기능을 하는 객체는 자주 바뀌고, 이 변화에 맞춰 A를 바꾸려면 힘이 들기 때문에 A를 바꾸지 않고 B를 적당히 변화시켜 추상화만 만족하는 방식으로 구조 유지가 가능하다.

함께 리팩토링해보기

- 저번에 리팩토링 하던 코드에서 이어가보겠습니다.



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

```
http
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 : 없음

- 고생하셨습니다
- 사실 과제를 내려면 낼 수는 있는데 이제 더 난이도를 올리기도 뭐하고 이정도면 토이프로젝트하는 데는 문제 없을 것 같네요.. 과제 4를 잘 하셨길 바랍니다