와플스튜디오 Spring Boot

최한결 @Hank-Choi

과제0

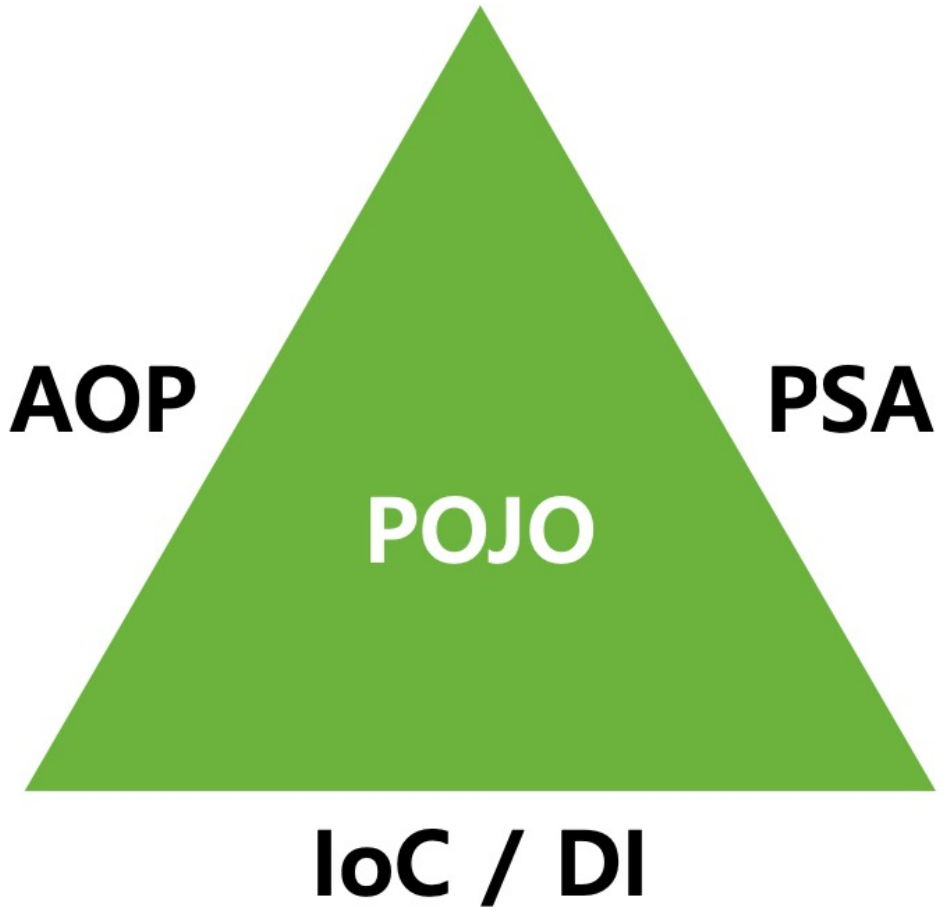
- ‘Issues 공유 매우 좋다’
- ‘어떻게 저렇게 빨리 찾지?’
- 생각보다 오래 걸린다.
- 안 알려주고 어떻게 하라는 거야?

좋은 코드란?

- 전달성
  - 읽기 쉬운 코드
  - 실제 코드 작성 시간은 총 작업시간 1/10

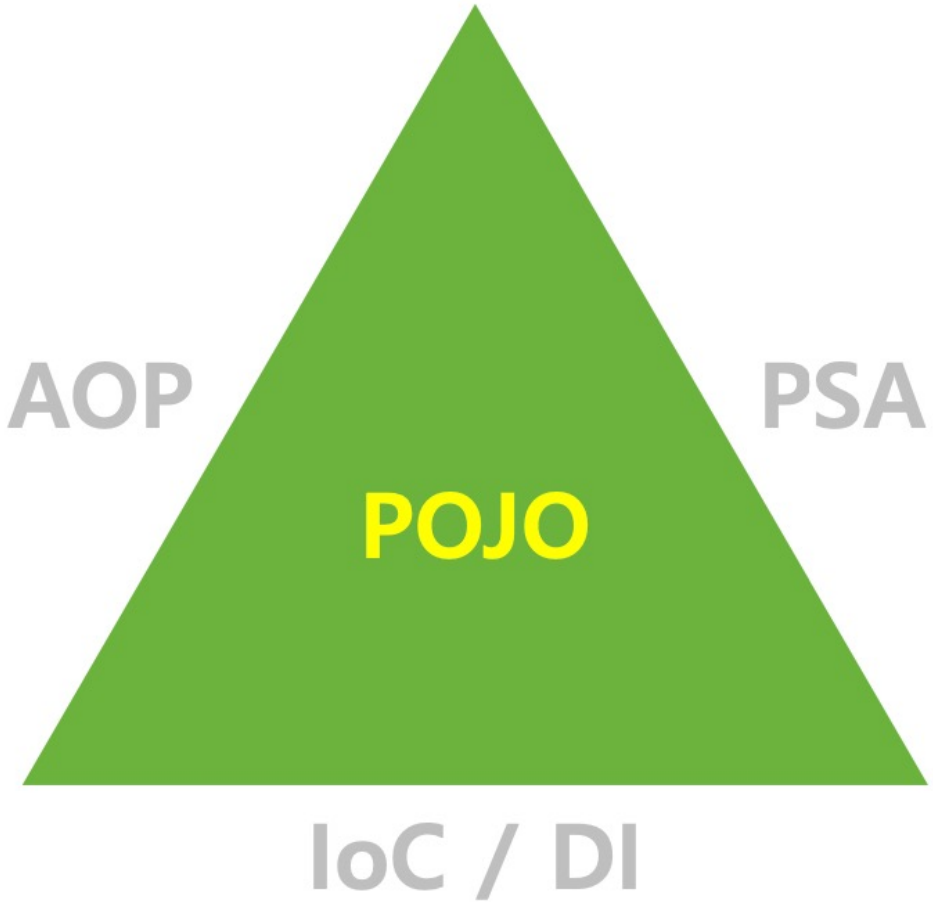
- 유연성
  - 인터페이스 , abstract class를 사용하는 이유?
  - 유통기한 3년

Spring



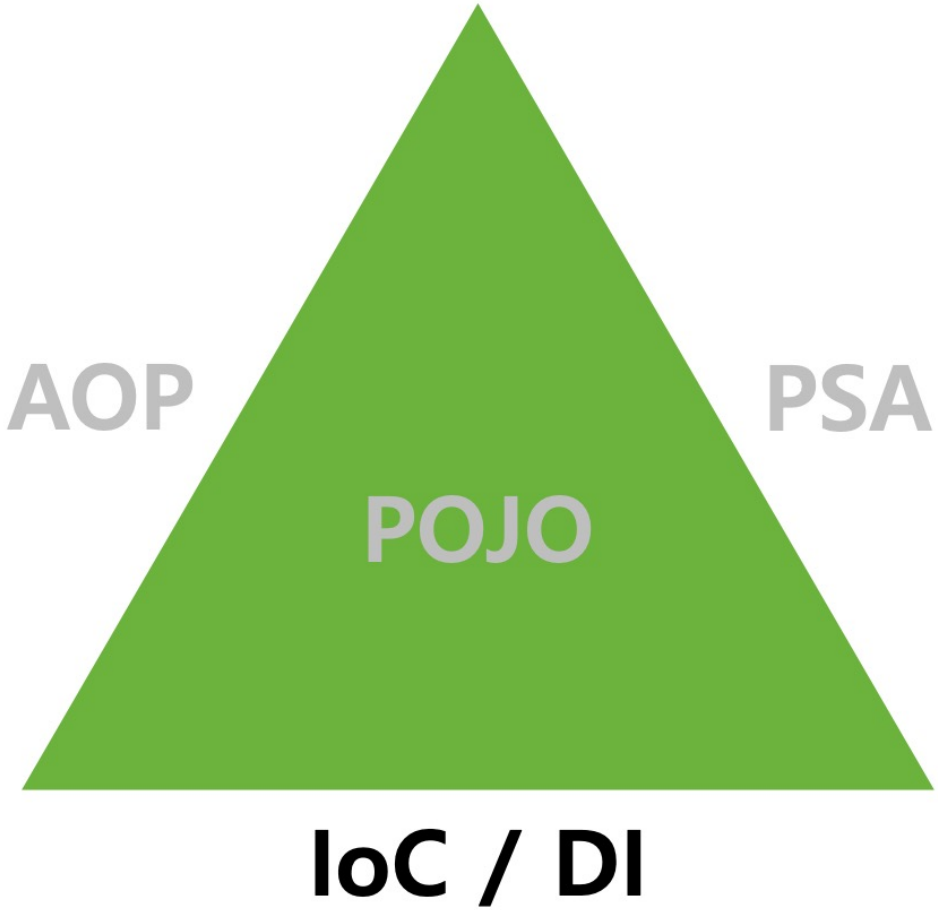
POJO(Plain Old Java Object)

- 객체지향 프로그래밍 기법과 언어가 주는 장점인 **유연한 설계**와 **재사용성** 활용하여 비즈니스의 복잡성과 변화를 상대함

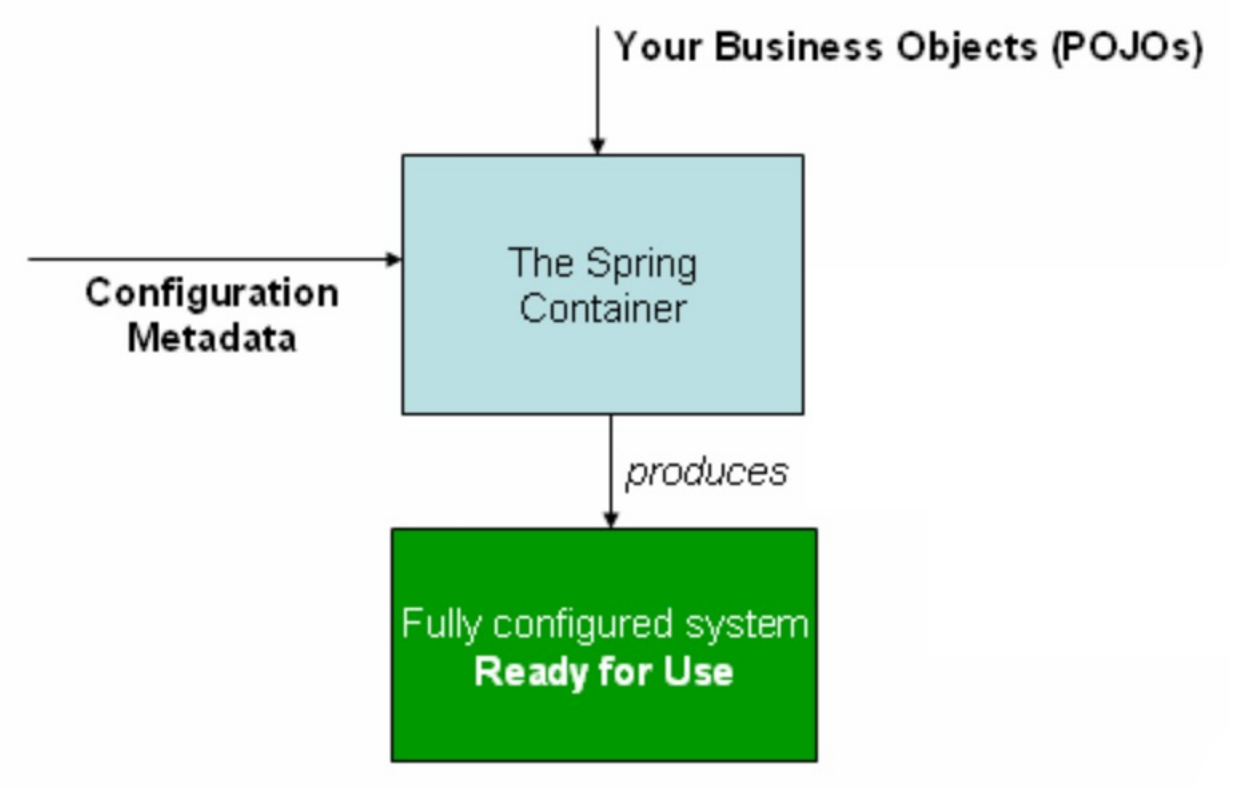


IoC / DI

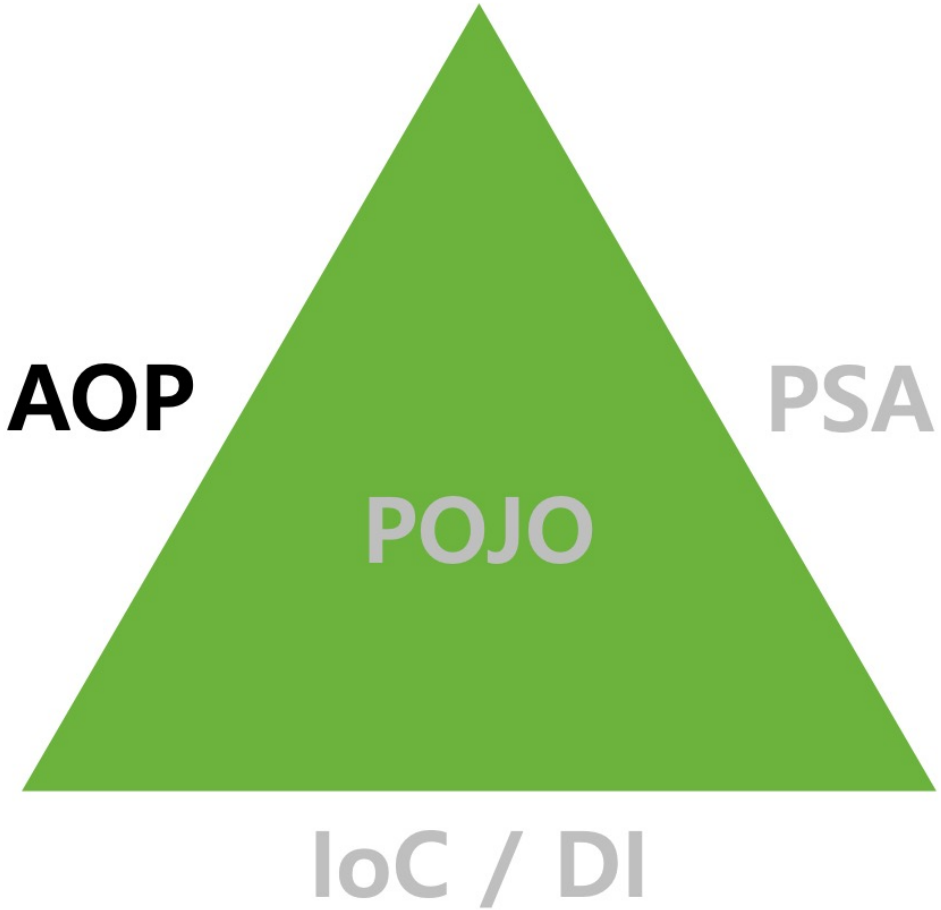
IoC (Inversion of Control)
DI (Dependency injection)



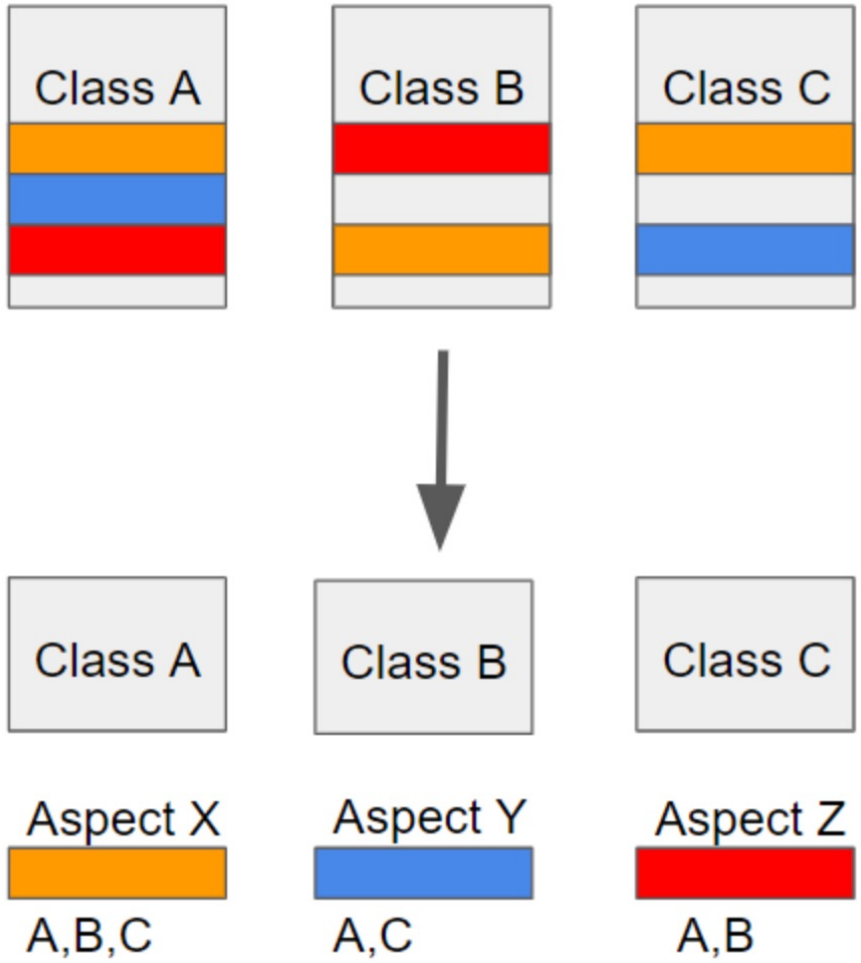
Bean


Your Business Objects (POJOs)
Configuration Metadata
The Spring Container
produces
Fully configured system
Ready for Use

AOP (Aspect Oriented Programming)

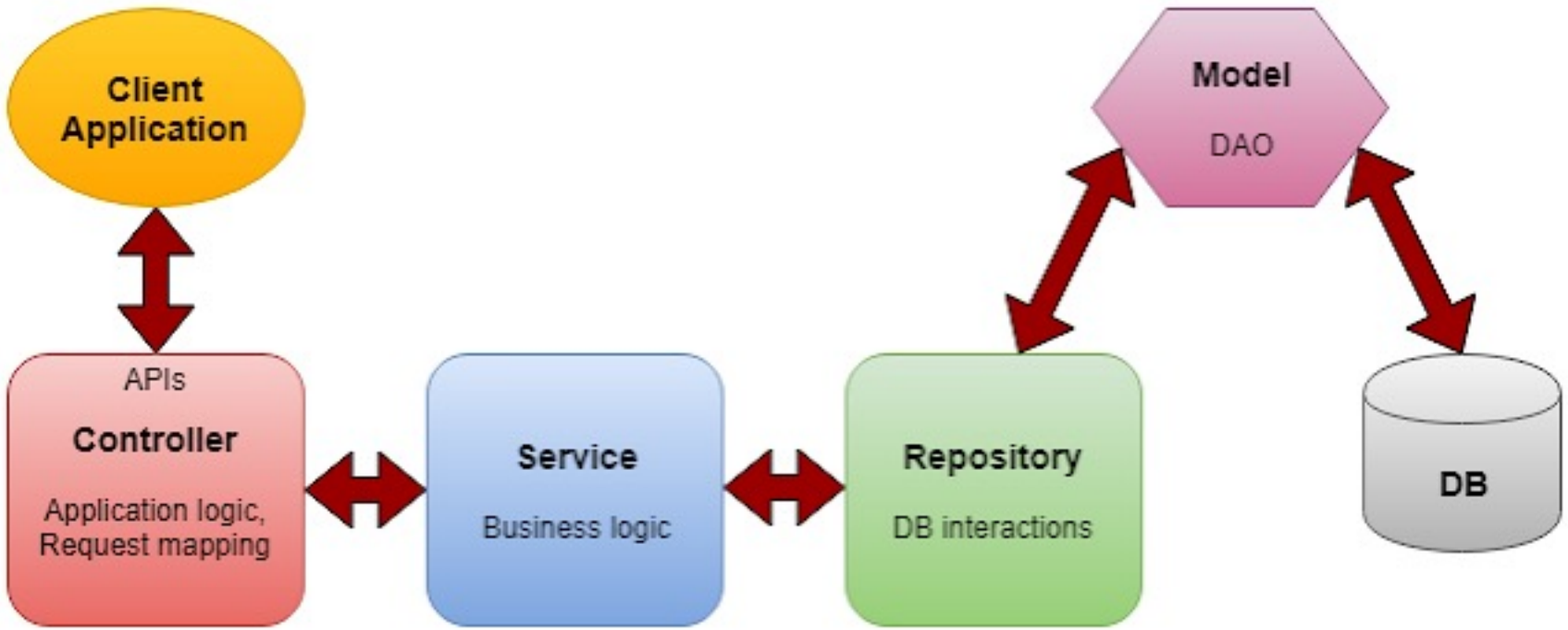


AOP (Aspect Oriented Programming)



레이어 아키텍쳐

- 프리젠테이션 레이어
- 서비스 레이어
- 데이터 엑세스 레이어



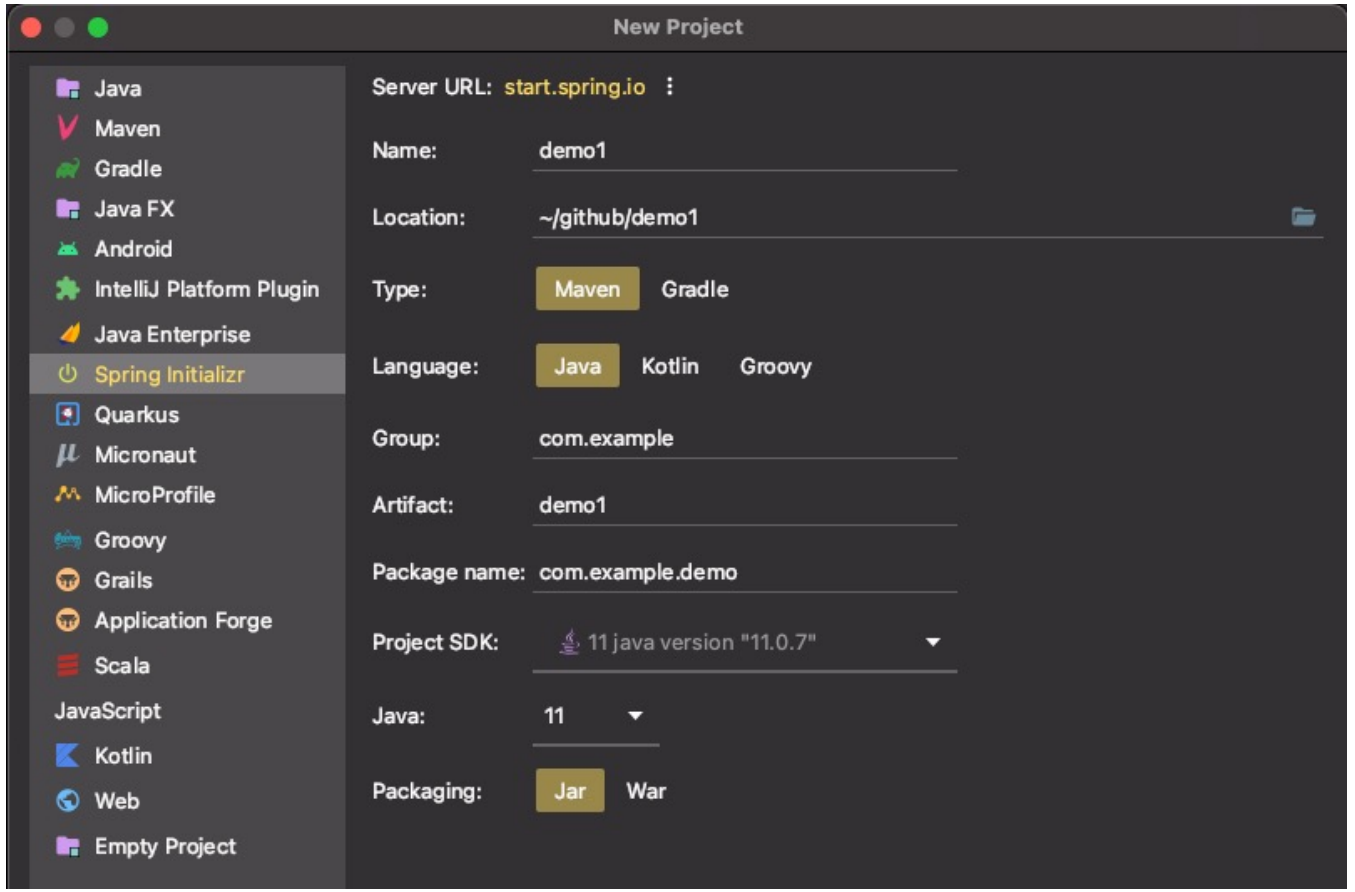
디자인

- 레이어 아키텍쳐
  - 관심사의 분리
  - 커플링이 느슨해짐
  - 애플리케이션 설계가 유연하고 견고해짐
  - 유지보수 및 테스트 용이

- 레이어간 데이터 전달에는 DTO(Data Transaction Object) 사용
  - 순수한 데이터 객체
  - Presentation layer 등에서 사용 (필요한 정보만 노출 가능)

Spring 시작하기

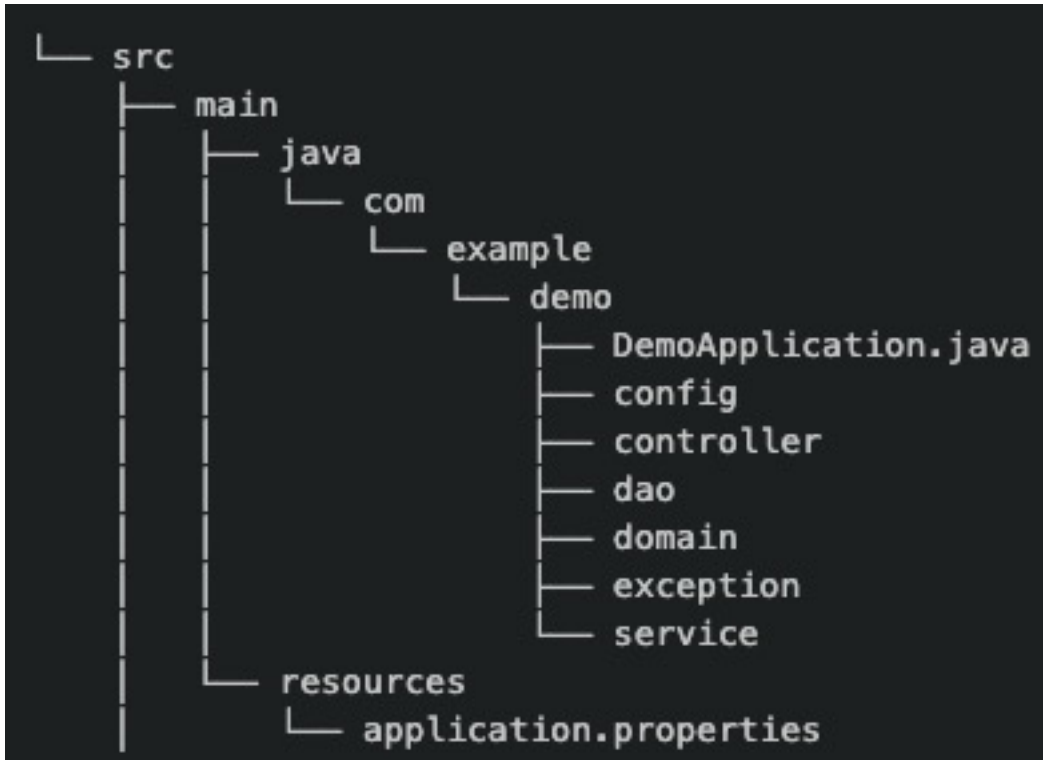
- https://start.spring.io/



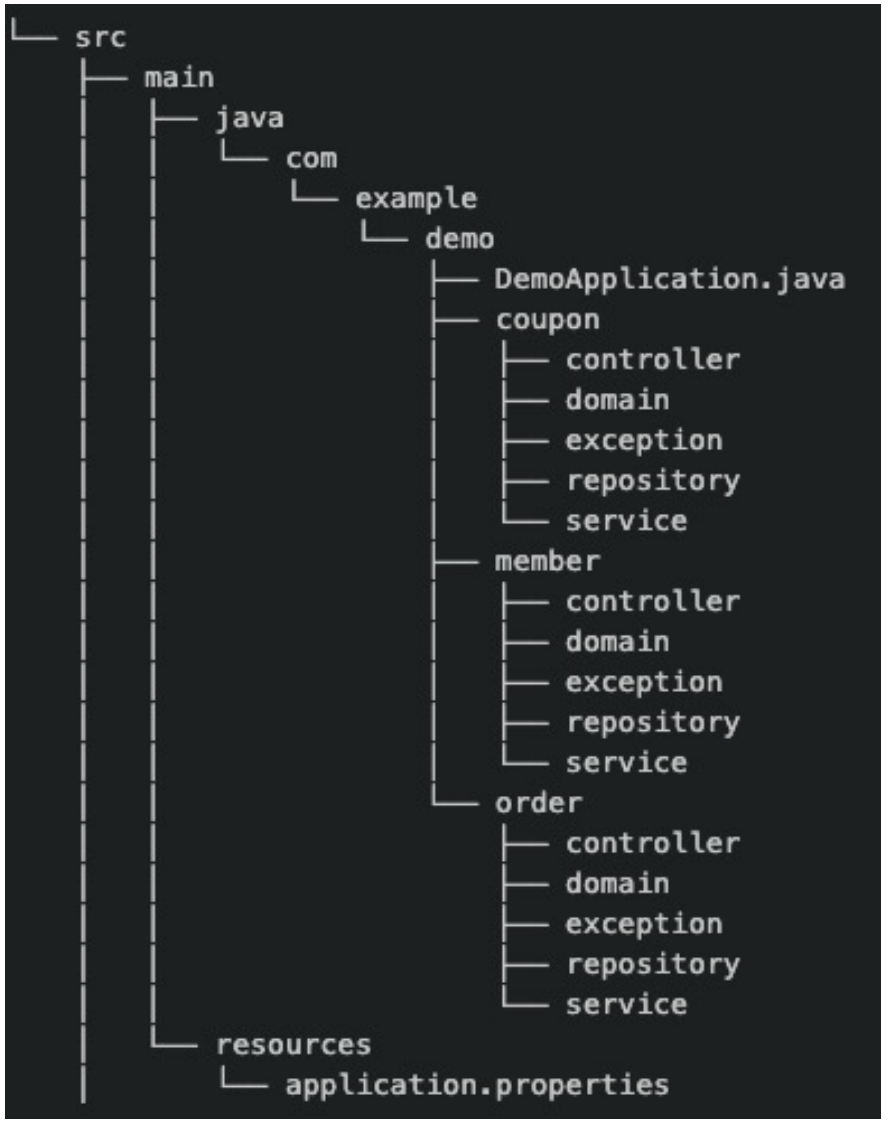
Gradle

- Java 진영의 빌드 툴
- 패키지 Dependency 관리
- 빌드 관리

Spring 시작하기



계층형



도메인형

Java vs Kotlin

- constructor
- 상속관계 표시
- 함수형 프로그래밍
- Nullable

함수형 프로그래밍

- Side effect를 최소화
- 간결한 표현

- https://github.com/wafflestudio/18.5-rookies/issues/159

스프링 세미나 과제 진행 방식

- 5인 1조
- 마감일까지 제출,
- 다음날 Peer review 작성
- 마감 이틀 뒤까지 리뷰 반영해서 제출