

Toby's Spring 3.1 읽기 스터디 5mins for what I learned

15주차 2021.12.09

AOP

발표자: 김정수

목차

1. 6장 간단 리뷰
2. 다이내믹 프록시와 팩토리 빈
3. 어드바이스와 포인트컷
4. AOP: 애스펙트 지향 프로그래밍

1. 6장 간단 리뷰

테스트

항상 단위 테스트를 먼저 고려하자.
여러 개의 단위가 의존관계를 가지고 동작할 경우
-> 통합 테스트 필요. but 가능한 많은 단위 테스트로 검증할 것!

다이내믹 프록시, 팩토리 빈

어드바이스, 포인트컷

-> Spring에서 AOP가 어떻게 사용되는지 배울 수 있었던 챕터!

2. 다이나믹 프록시와 팩토리 빈

프록시

대신 클라이언트의 요청을 받아주는 **대리자**.

- 프록시 패턴 : 접근하는 방법 제어
- 데코레이터 패턴: 추가적인 기능 부여

다이나믹 프록시는 프록시 팩토리에 의해 런타임 시 만들어지며, 타깃 클래스와 같은 구성의 **인터페이스**를 만들고, 프록시로서 필요한 **부가기능**을 구현해야 한다.

팩토리 빈

스프링을 대신해서 **객체의 생성로직을 담당하는 Bean**이다. (공장 Bean)

FactoryBean이라는 인터페이스를 구현하며, getObject()에 다이나믹 프록시 오브젝트를 만들어주는 코드를 넣으면 다이나믹 프록시를 생성할 수 있다.
(재사용 가능 but 부가기능 추가할 때마다 메서드가 아닌 객체 추가 필요)

3. 어드바이스와 포인트컷

어드바이스

:프록시가 호출하는 부가 기능.

+

포인트컷

: 부가기능을 적용할지에 대한 필터링 기준

= 어드바이저

자동 프록시 만드는 법

1. MethodInterceptor를 구현한 어드바이스를 생성한다.
2. 스프링이 제공하는 pointcut Class를 하나 골라 객체를 생성한다.
3. 만든 pointcut 객체에 필터링 규칙을 지정한다.
4. 어드바이저에 포인트컷과 어드바이스를 넣어준다.

-> Bean 객체를 만들 때마다 스프링은 빈 후처리기에 빈을 보낸다.

-> 후처리기는 빈으로 등록된 모든 어드바이저 내의 포인트컷을 이용해 자동으로 대상이 되는 빈에 프록시를 적용한다.

-> 적용할 때, 빈을 프록시로 감싸는 방식을 사용한다.

4. AOP: 애스펙트 지향 프로그래밍

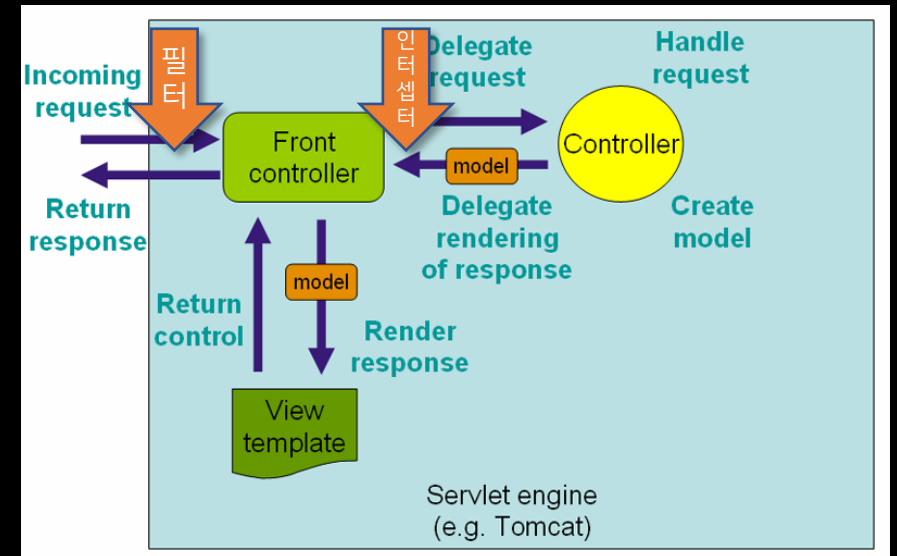
AOP: 애스펙트 지향 프로그래밍

: 애플리케이션의 핵심적인 기능에서 부가적인 기능을 분리해 모듈로 만들어서 설계하고, 개발하는 방법

- 부가기능 모듈을 애스펙트(Aspect)라고 한다.
- 애스펙트를 분리함으로써 핵심기능을 설계할 때 객체지향을 따를 수 있게 한다.
- 애플리케이션을 특정한 관점을 기준으로 바라볼 수 있게 해준다는 의미로 관점 지향 프로그래밍이라고 한다.

Use Case

- 트랜잭션 경계설정
- Servlet filter: Http 요청이나 응답을 원하는 객체로 감쌀 수 있다.
- Spring Interceptor: 요청의 전/후를 처리할 수 있으며, 매핑된 handle나 ModelAndView, 예외를 참조할 수 있다.



완독을 축하합니다😊