Spring-Transaction

트랜잭션이란

DB의 상태를 변화시키는 논리적 단위

웹 애플리케이션에선 Service 클래스의 각 메서드가 애플리케이션의 단위 기능

트랜잭션의 속성

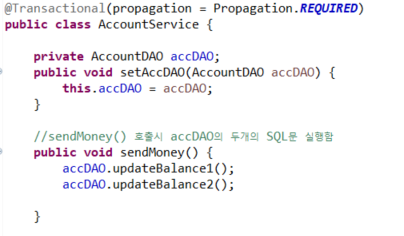
원자성(Atomicity) - 나눌수 없는 하나의 작업으로 다뤄져야 함

일관성(Consistency) - Tx 수행 전과 후가 일관된 상태를 유지해야 함

고립성(Isolation) - 각 Tx는 독립적으로 수행되어야 함

영속성(Durability) - 성공한 Tx의 결과는 유지되어야 함

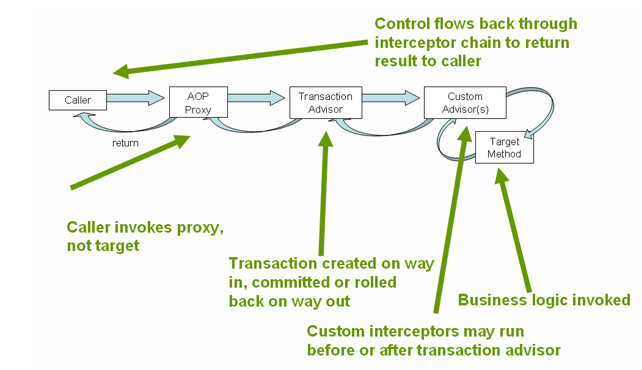
@Transactional



클래스나 메서드에 붙일 수 있음, 해당 범위 내 메서드는 트랜잭션이 보장됨

Spring에서 트랜잭션 동작 원리

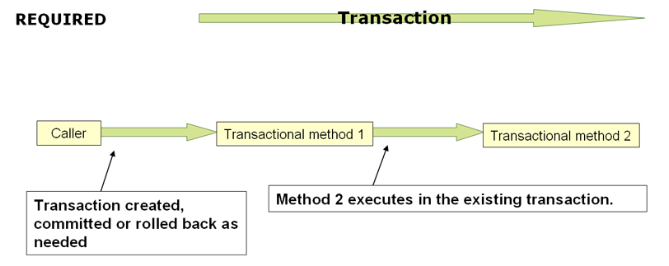
Proxy 방식(aspect,target)



bean 인 aspcct를 target과 연결해주는 proxy클래스

propagation(전파 방식)

REQUIRED



progatiaon의 디폴트값

매서드1에 대한 트랜잭션이 진행중이고 매서드2작업이 시작된다면

매서드2 => 매서드1 나머지 순서로 처리됨, 예외 발생시 둘 다 취소

REQUIRED\_NEW

각자의 트랜잭션이 진행, 매서드2 커밋or 롤백 후 매서드1 커밋or롤백

10. Spring Annotation

어노테이션이란

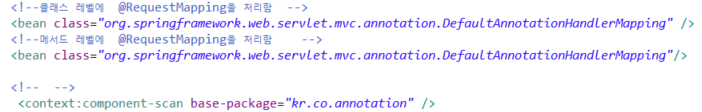
코드의 Metadata로 작성, 컴파일 런타임에 활용됨

어노테이션을 통해 연결, 데이터 가공, 예외 처리등을 구성함

JDK가 제공하는 빌트인과 직접 작성하는 커스텀으로 분류됨

@{AnnotationName}으로 표기

Spring Annotation 제공 클래스



DefaultAnnotationHandlerMapping

클래스 레벨에서 @ResqusetMapping을 처리함

AnnotationMethodHandlerAdapte

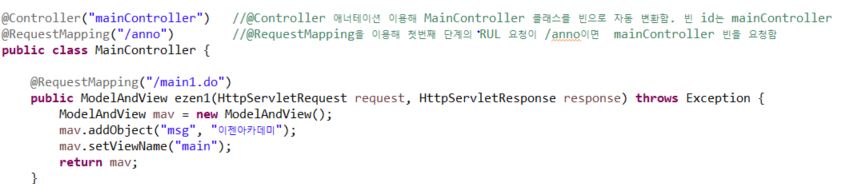
메소드 레벨에서 @ResqusetMapping을 처리함

<context:component-scan base-package="패키지주소" />

스프링 컨테이너가 해당 패키지를 스캔하여 어노테이션 지정된 클래스나

메소드를 빈으로 만들어줌

여러가지 스테레오 타입 어노테이션



@Controller, @Service : 컨트롤러와 서비스 빈

@Repository : DAO, DB에 접근하는 메소드를 가진 class에 쓰임

@component : 개발자가 직접 작성한 Class를 Bean으로 등록하기 위한 어노테이션

@Controller, @Service, @RequestMapping는 @component가

구체화된 유스케이스

@RequestMapping : 컨트롤러가 실행될 요청 URL 매핑