

스프링 2주차 공부

모각소 3차 발표(02.10)

박수인

1. 컴포넌트 스캔

설정 정보 없이 **자동으로 스프링 빈 등록**하는 스프링의 기능

- `@Component`가 붙은 클래스를 스캔 ➡ 스프링 빈으로 등록

```
@Configuration
@ComponentScan(
    excludeFilters = @Filter(type = FilterType.ANNOTATION, classes = Configuration.class))
public class AutoAppConfig {

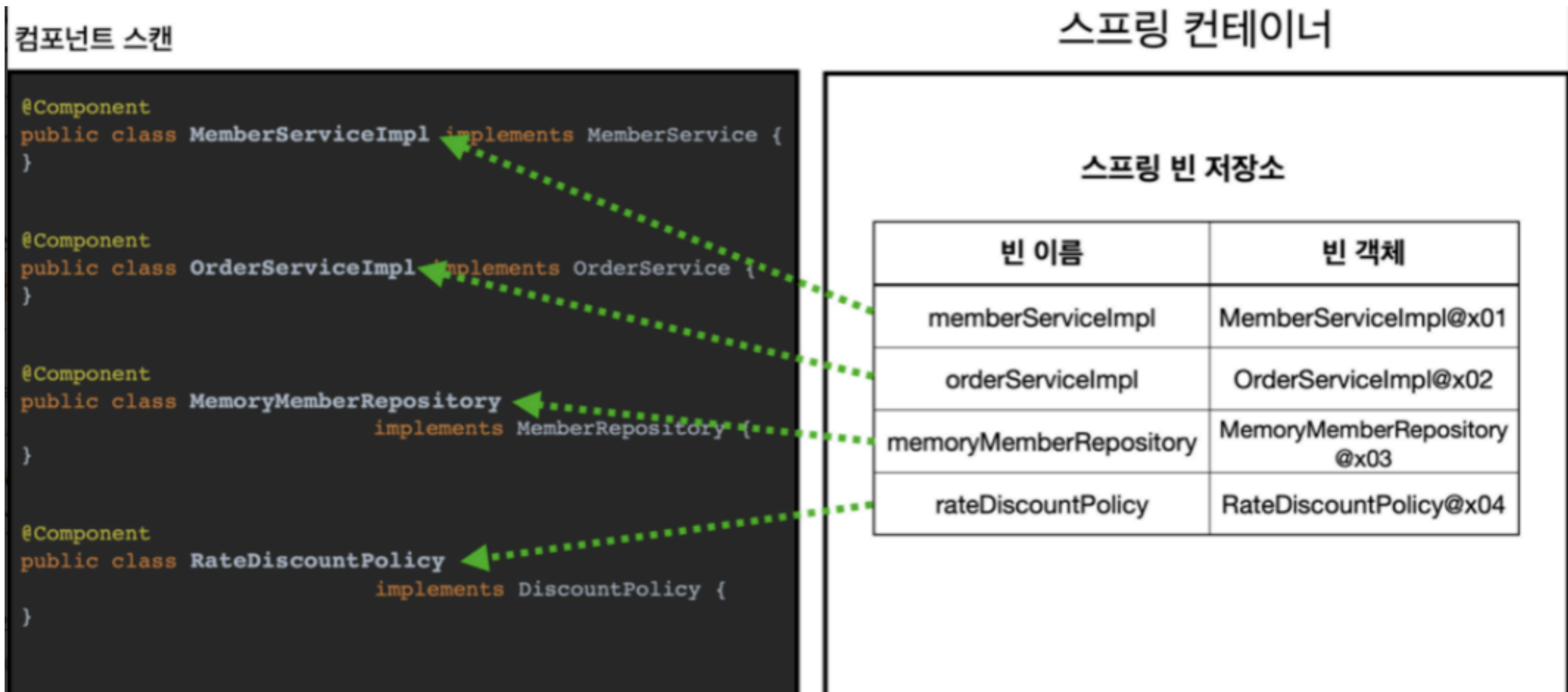
}
```

- `@ComponentScan`을 설정 정보에 붙여주면 됨
- 기존의 AppConfig와 달리, @Bean으로 등록한 클래스가 하나도 없음

1. 컴포넌트 스캔

설정 정보 없이 **자동으로 스프링 빈 등록**하는 스프링의 기능

- @Componentscan은 @Component가 붙은 클래스를 스프링 빈으로 등록



1. 컴포넌트 스캔 - 스캔 기본 대상

- `@Component` 뿐만 아니라 다음 내용도 추가로 대상에 포함
 - `@Component` : 컴포넌트 스캔에서 사용
 - `@Controller` : 스프링 MVC 컨트롤러에서 사용
 - `@Service` : 스프링 비즈니스 로직에서 사용
 - `@Repository` : 스프링 데이터 접근 계층에서 사용
 - `@Configuration` : 스프링 설정 정보에서 사용
- 다음 애노테이션은 컴포넌트 스캔 용도 + 부가 기능까지 수행함
 - `@Controller` : 스프링 MVC 컨트롤러로 인식
 - `@Repository` : 스프링 데이터 접근 계층으로 인식하고, 데이터 계층의 예외를 스프링 예외로 변환
 - `@Configuration` : 앞서 보았듯이 스프링 설정 정보로 인식하고, 스프링 빈이 싱글톤을 유지하도록 추가 처리
 - `@Service` : `@Service` 는 특별한 처리를 하지 않음 **But** 개발자들이 비즈니스 계층을 인식하는데 도움

2. 의존관계 주입

- `@Autowired` 사용 ➡ 생성자에서 여러 의존관계도 한 번에 주입받음

의존관계 자동 주입

```
@Component
public class MemberServiceImpl implements MemberService {

    private final MemberRepository memberRepository;

    @Autowired
    public MemberServiceImpl(MemberRepository memberRepository) {
        this.memberRepository = memberRepository;
    }
}
```

스프링 컨테이너

스프링 빈 저장소

빈 이름	빈 객체
memberServiceImpl	MemberServiceImpl@x01
orderServiceImpl	OrderServiceImpl@x02
memoryMemberRepository	MemoryMemberRepository @x03
rateDiscountPolicy	RateDiscountPolicy@x04

- 생성자에 `@Autowired`를 지정 ➡ 스프링 컨테이너가 자동으로 해당 스프링 빈 찾아서 주입

2. 의존관계 주입 - 주입 방법

- `@Autowired` 사용 ➡ 생성자에서 여러 의존관계도 한 번에 주입받음

의존관계 자동 주입

```
@Component
public class MemberServiceImpl implements MemberService {

    private final MemberRepository memberRepository;

    @Autowired
    public MemberServiceImpl(MemberRepository memberRepository) {
        this.memberRepository = memberRepository;
    }
}
```

스프링 컨테이너

스프링 빈 저장소

빈 이름	빈 객체
memberServiceImpl	MemberServiceImpl@x01
orderServiceImpl	OrderServiceImpl@x02
memoryMemberRepository	MemoryMemberRepository @x03
rateDiscountPolicy	RateDiscountPolicy@x04

- 생성자에 `@Autowired`를 지정 ➡ 스프링 컨테이너가 자동으로 해당 스프링 빈 찾아서 주입

2. 의존관계 주입 - 주입 방법

1. 생성자 주입 <- 권장 ★

- * 불변, 필수 의존관계에 사용
- * 생성자 호출 시점에 딱 1번만 호출

```
@Component
public class OrderServiceImpl implements OrderService {
    private final MemberRepository memberRepository;
    private final DiscountPolicy discountPolicy;

    @Autowired
    public OrderServiceImpl(MemberRepository memberRepository, DiscountPolicy discountPolicy) {
        this.memberRepository = memberRepository; this.discountPolicy = discountPolicy;
    }
}
```

2. 의존관계 주입 - 주입 방법

2. 수정자 주입(setter 주입)

* 선택, 변경 가능성 있는 의존관계에 사용

```
@Component
public class OrderServiceImpl implements OrderService {
    private final MemberRepository memberRepository;
    private final DiscountPolicy discountPolicy;
    @Autowired
    public void setMemberRepository(MemberRepository memberRepository) {
        this.memberRepository = memberRepository;
    }
    @Autowired
    public void setDiscountPolicy(DiscountPolicy discountPolicy) {
        this.discountPolicy = discountPolicy;
    }
}
```


3. 중복 등록과 충돌

1. 자동 빈 등록 vs 자동 빈 등록

오류 발생 - `ConflictingBeanDefinitionException` 예외 발생

2. 수동 빈 등록 vs 자동 빈 등록

수동 빈 등록이 우선권을 가짐 (수동 빈이 자동 빈을 오버라이딩)

최근 스프링 부트에서는 오류 발생하도록 기본 값을 바꿈