

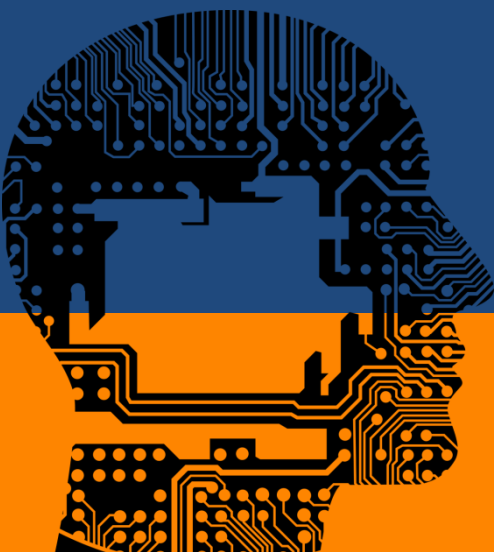


JAVA 웹 개발자 양성과정

Spring: Spring Core

5강 - 컴포넌트 스캔

By SoonGu Hong



JAVA 웹 개발자 양성과정

Spring: Spring Core

1. 컴포넌트 스캔

컴포넌트 스캔과 의존관계 자동 주입 시작하기

- 지금까지 스프링 빈을 등록할 때는 자바 코드의 @Bean이나 XML의 <bean> 등을 통해서 설정 정보에 직접 등록할 스프링 빈을 나열했다.
- 예제에서는 몇개가 안되었지만, 이렇게 등록해야 할 스프링 빈이 수십, 수백개가 되면 일일이 등록하기도 귀찮고, 설정 정보도 커지고, 누락하는 문제도 발생한다. 역시 개발자는 반복을 싫어한다.(무엇보다 귀찮다 π π)
- 그래서 스프링은 설정 정보가 없어도 자동으로 스프링 빈을 등록하는 컴포넌트 스캔이라는 기능을 제공한다.
- 또 의존관계도 자동으로 주입하는 @Autowired 라는 기능도 제공한다.

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

}
```

- 컴포넌트 스캔을 사용하려면 먼저 `@ComponentScan` 을 설정 정보에 붙여주면 된다.
- 기존의 AppConfig와는 다르게 `@Bean`으로 등록한 클래스가 하나도 없다!

컴포넌트 스캔은 이름 그대로 `@Component` 애노테이션이 붙은 클래스를 스캔해서 스프링 빈으로 등록한다. `@Component` 를 붙여주자.

```
@Component
public class MemberServiceImpl implements MemberService {

    private final MemberRepository memberRepository;

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

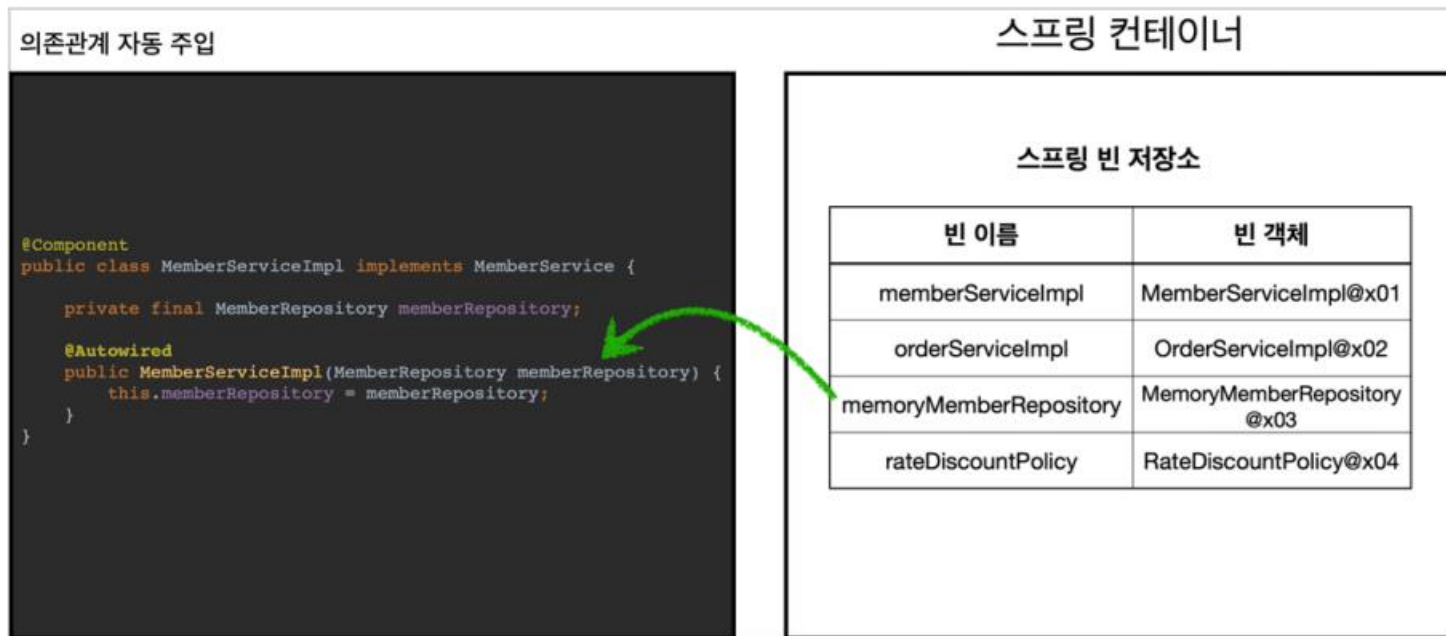
- 이전에 AppConfig에서는 `@Bean` 으로 직접 설정 정보를 작성했고, 의존관계도 직접 명시했다. 이제는 이런 설정 정보 자체가 없기 때문에, 의존관계 주입도 이 클래스 안에서 해결해야 한다.

1. @ComponentScan



- @ComponentScan 은 @Component 가 붙은 모든 클래스를 스프링 빈으로 등록한다.
- 이때 스프링 빈의 기본 이름은 클래스명을 사용하되 맨 앞글자만 소문자를 사용한다.
 - **빈 이름 기본 전략:** MemberServiceImpl 클래스 → memberServiceImpl
 - **빈 이름 직접 지정:** 만약 스프링 빈의 이름을 직접 지정하고 싶으면
@Component("memberService2") 이런식으로 이름을 부여하면 된다.

2. @Autowired 의존관계 자동 주입



- 생성자에 `@Autowired` 를 지정하면, 스프링 컨테이너가 자동으로 해당 스프링 빈을 찾아서 주입한다.
- 이때 기본 조회 전략은 타입이 같은 빈을 찾아서 주입한다.
 - `getBean(MemberRepository.class)` 와 동일하다고 이해하면 된다.
 - 더 자세한 내용은 뒤에서 설명한다.

```
@ComponentScan(  
    basePackages = "hello.core",  
)
```

- `basePackages` : 탐색할 패키지의 시작 위치를 지정한다. 이 패키지를 포함해서 하위 패키지를 모두 탐색한다.
 - `basePackages = {"hello.core", "hello.service"}` 이렇게 여러 시작 위치를 지정할 수도 있다.
- `basePackageClasses` : 지정한 클래스의 패키지를 탐색 시작 위치로 지정한다.
- 만약 지정하지 않으면 `@ComponentScan` 이 붙은 설정 정보 클래스의 패키지가 시작 위치가 된다.

권장하는 방법

개인적으로 즐겨 사용하는 방법은 패키지 위치를 지정하지 않고, 설정 정보 클래스의 위치를 프로젝트 최상단에 두는 것이다. 최근 스프링 부트도 이 방법을 기본으로 제공한다.

예를 들어서 프로젝트가 다음과 같이 구조가 되어 있으면

- `com.hello`
- `com.hello.serivce`
- `com.hello.repository`

`com.hello` → 프로젝트 시작 루트, 여기에 AppConfig 같은 메인 설정 정보를 두고, @ComponentScan 애노테이션을 붙이고, `basePackages` 지정은 생략한다.

컴포넌트 스캔 기본 대상

컴포넌트 스캔은 @Component 뿐만 아니라 다음과 내용도 추가로 대상에 포함한다.

- @Component : 컴포넌트 스캔에서 사용
- @Controlller : 스프링 MVC 컨트롤러에서 사용
- @Service : 스프링 비즈니스 로직에서 사용
- @Repository : 스프링 데이터 접근 계층에서 사용
- @Configuration : 스프링 설정 정보에서 사용

감사합니다
THANK YOU