컴포넌트 스캔

스프링이 직접 클래스를 검색해서 빈으로 등록해주는 기능이다.

@Component 애노테이션으로 스캔 대상 지정

- @Component 애노테이션에 값을 주지 않으면 클래스 이름의 첫 글자를 소문자로

바꾼 이름을 빈이름으로 사용한다.

@ComponentScan 애노테이션으로 스캔 설정

- 빈 설정 클래스에서 @ComponentScan 애노테이션을 적용하면 @Component 애노테이션을

붙인 클래스를 검색해서 빈으로 등록해준다.

- @ComponentScan 애노테이션의 basePackages 속성값은 스캔 대상 패키지 목록을 지정한다.

ex) @ComponentScan(basePackages ={”패키지 명”})

스캔 대상에서 제외하거나 포함하기

- excludeFilters 속성을 사용하면 스캔할 때 특정 대상을 자동 등록 대상에서 제외할 수 있다.

ex) @ComponentScan(basePackages = {”패키지 명”},

excludeFilters = @Filter(type = **FilterType.REGEX**, pattern = “spring\\..*DAO”))

FilterType.REGEX 타입은 정규표현식을 사용해서 제외 대상을 지정한다는 의미이다. spring 패키지 아래 있으면서 DAO로 끝나는 대상을 스캔대상에서 제외

- FilterType.ASPECTJ 타입은 정규표현식 대신 AspectJ 패턴을 사용해서 대상을 지정한다.

ex) @ComponentScan(basePackage = {”패키지명”},

excludeFilters = @Filter(type = **FilterType.ASPECTJ**, pattern = “spring.*DAO”))

spring 패키지 하위에 DAO로 끝나는 대상 스캔대상에서 제외

(AspectJ 패턴이 동작하려면 의존대상에 aspectjweaver 모듈을 추가해야한다.)

- patterns 속성은 String[]타입이므로 배열을 이용해서 패턴을 한 개 이상 지정 할 수 있다.

- 특정 애노테이션을 붙인 타입을 제외할 수 있다.

ex) @ComponentScan(basePackages = {”패키지명”, “패키지명”},

excludeFilters = @Filter(type = **FilterType.ANNOTATION**,

classes = {NoProduct.class, ManualBean.class}))

Noproduct와 ManualBean을 어노태이션으로 가진 클래스는 제외된다.

- 특정 타입이나 그 하위 타입을 컴포넌트 스캔 대상에서 제외할 수 있다.

ex) @ComponentScan(basePackages = {”패키지명”},

excludeFilters = @Filter(type = **FilterType.ASSIGNABLE_TYPE**,

classes = MemberDao.class))

- 필터가 두개 이상이라면 @ComponentScan의 excludeFilters 속성에 배열을 사용해서,

@Filter 목록을 전달하면 된다.

기본 스캔 대상

- @Component
- @Controller
- @Service
- @Repository

- @Aspect
- @Configuration

컴포넌트 스캔에 따른 충돌 처리

- 서로 다른 타입인데 같은 빈 이름이 같다면 충돌 꼭 둘 중 하나에 명시적으로 빈이름 지정하기!

- 스캔할 때 사용하는 빈 이름과 수동 등록한 빈 이름이 같은경우 수동 등록한 빈이 우선 순위를 가진다.