

## 1. 소개

가. 스프링 프레임워크에서 제공하는 스테레وتا입 애노테이션

나. 지금까지 오브젝트를 생성하고 의존성 주입을 하기 위해 XML 설정을 사용

다. 스테레وتا입 애노테이션

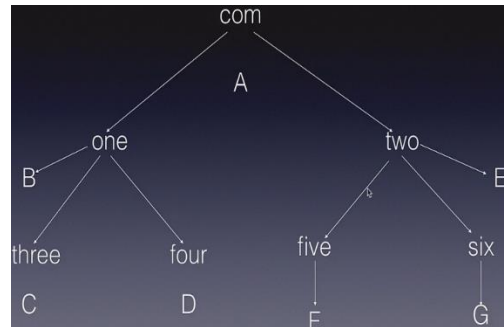
- @Component: XML 설정 bean tag와 동일

XML `<bean...>`

@Component

class Instructor

`<context:component-scan base-package="com.bharath"/>`



- 미리 컴파일된 클래스(서드파티)는 접근할 수 없어, 우리가 만든 클래스만 적용 가능

@Component

class Instructor

Instructor instructor = new Instructor();

User Defined Types

## 2. 애노테이션을 이용한 오브젝트 생성

```
1 package com.springcoreadvanced.stereotype.annotations;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class Instructor {
7
8     private int id;
9     private String name;
10
11     public int getId() {
12         return id;
13     }
14
15     public void setId(int id) {
16         this.id = id;
17     }
18
19     public String getName() {
20         return name;
21     }
22
23     public void setName(String name) {
24         this.name = name;
25     }
26
27     @Override
28     public String toString() {
29         return "Instructor [id=" + id + ", name=" + name + "];"
30     }
31 }
```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:p="http://www.springframework.org/schema/p"
6       xmlns:c="http://www.springframework.org/schema/c"
7       xsi:schemaLocation="http://www.springframework.org/schema/beans
8       http://www.springframework.org/schema/beans/spring-beans.xsd
9       http://www.springframework.org/schema/context
10      http://www.springframework.org/schema/context/spring-context.xsd">
11
12     <context:component-scan
13         base-package="com.springcoreadvanced.sterotype.annotations" />
14
15 </beans>

```

Context:annotation-config 활성화

더 이상 필요하지 않음

### 3. 다른 오브젝트를 이름을 이용하기

```

1 package com.springcoreadvanced.sterotype.annotations;
2
3 import org.springframework.stereotype.Component;
4
5 @Component("inst") 사용자 빈 이름 지정
6 public class Instructor {
7
8     private int id;
9     private String name;
10
11     public int getId() {
12         return id;
13     }
14
15     public void setId(int id) {
16         this.id = id;
17     }
18
19     public String getName() {
20         return name;
21     }
22
23     public void setName(String name) {
24         this.name = name;
25     }
26
27     @Override
28     public String toString() {
29         return "Instructor [id=" + id + ", name=" + name + "];"
30     }
31 }

```

```

1 package com.springcoreadvanced.sterotype.annotations;
2
3 import org.springframework.context.ApplicationContext;
4
5 public class Test {
6
7     private static ApplicationContext context;
8
9     public static void main(String[] args) {
10         context = new ClassPathXmlApplicationContext(
11             "com/springcoreadvanced/sterotype/annotations/config.xml");
12         Instructor instructor = (Instructor) context.getBean("inst");
13         System.out.println(instructor);
14     }
15 }

```

### 4. 스코프 애노테이션 사용

디폴트 **singleton**

prototype

@Scope request

session

globalsession

```

1 package com.springcoreadvanced.sterotype.annotations;
2
3 import org.springframework.context.ApplicationContext;
4
5 public class Test {
6
7     private static ApplicationContext context;
8
9     public static void main(String[] args) {
10         context = new ClassPathXmlApplicationContext(
11             "com/springcoreadvanced/sterotype/annotations/config.xml");
12         Instructor instructor = (Instructor) context.getBean("inst");
13         System.out.println(instructor.hashCode());
14         Instructor instructor2 = (Instructor) context.getBean("inst");
15         System.out.println(instructor2.hashCode());
16     }
17 }

```

두 개의 객체 생성

Markers Properties

<terminated> Test (16) [Java  
4월 08, 2023 7:06:19 S  
INFO: Refreshing org.s  
4월 08, 2023 7:06:19 S  
INFO: Loading XML bean

1571798597

1571798597

두 객체의 해시코드 동일 -> 스프링 컨테이너는 한 개의 객체만 생성

```
1 package com.springcoreadvanced.sterotype.annotations;  
2  
3 import org.springframework.context.annotation.Scope;  
4 import org.springframework.stereotype.Component;  
5  
6 @Component("inst")  
7 @Scope("prototype")  
8 public class Instructor {  
9  
10     private int id;  
11     private String name;
```

Markers Properties

<terminated> Test (16) [Java A  
4월 08, 2023 7:11:11 오  
INFO: Refreshing org.sp  
4월 08, 2023 7:11:11 오  
INFO: Loading XML bean

1351478315

508512860

두 객체의 해시코드 상이 -> 스프링 컨테이너는 두 개의 객체 생성

## 5. 원시타입을 애노테이션 값으로 사용

```
1 package com.springcoreadvanced.sterotype.annotations;  
2  
3 import org.springframework.context.ApplicationContext;  
4  
5  
6 public class Test {  
7  
8     private static ApplicationContext context;  
9  
10     public static void main(String[] args) {  
11         context = new ClassPathXmlApplicationContext(  
12             "com/springcoreadvanced/sterotype/annotations/config.xml");  
13         Instructor instructor = (Instructor) context.getBean("inst");  
14         System.out.println(instructor);  
15  
16         Instructor instructor2 = (Instructor) context.getBean("inst");  
17         System.out.println(instructor2);  
18     }  
19  
20 }
```

두 객체의 주소값이 아닌 기본값 출력

Markers Properties Serv

<terminated> Test (16) [Java Applicat  
4월 08, 2023 7:30:29 오후 org  
INFO: Refreshing org.springframework  
4월 08, 2023 7:30:30 오후 org  
INFO: Loading XML bean defini  
Instructor [id=0, name=null]  
Instructor [id=0, name=null]

## @Value

Primitive Types	@Value("20") @Value("Core Java")
Collection Types	util:Cn id="myList" @Value("#{myList}")
Object Types	@Autowired

The screenshot shows an IDE with three tabs: `Instructor.java`, `config.xml`, and `Test.java`. The `Instructor.java` file contains the following code:

```
1 package com.springcoreadvanced.stereotype.annotations;
2
3 import org.springframework.beans.factory.annotation.Value;
4 import org.springframework.context.annotation.Scope;
5 import org.springframework.stereotype.Component;
6
7 @Component("inst")
8 @Scope("prototype")
9 public class Instructor {
10
11     @Value("10")
12     private int id = 15;
13     @Value("woseven")
14     private String name = "kimht";
15
16     public int getId() {
17         return id;
18     }
19
20     public void setId(int id) {
21         this.id = id;
22     }
23
24     public String getName() {
25         return name;
26     }
27
28     public void setName(String name) {
29         this.name = name;
30     }
31
32     @Override
33     public String toString() {
34         return "Instructor [id=" + id + ", name=" + name + "]";
35     }
36 }
```

The console output on the right shows the following messages:

```
<terminated> Test (16) [Java Application] C
4월 08, 2023 11:21:07 오후 org.sp
INFO: Refreshing org.springframework
4월 08, 2023 11:21:07 오후 org.sp
INFO: Loading XML bean definition:
Instructor [id=10, name=woseven]
Instructor [id=10, name=woseven]
```

- 컬렉션타입을 애노테이션 값으로 사용
- 오브젝트(object) 오토 와이어링(auto-wiring)
- 요약

The slide is titled "Stereotype Annotations" and lists the following annotations:

- `@Component`
- `<context:component-scan base-package = "com.app.beans">`
- `@Scope`
- `@Value`
- `@Autowired`