

Scope and Value Annotations

1. Scope Annotation (@Scope)

- **Purpose:**

Specifies the lifecycle and scope of a bean in a Spring application.

The default scope in Spring is **singleton**, which means only one instance of the bean is created for the application context.

- **Usage:**

- `@Scope("prototype")`: Creates a new instance of the bean every time it is requested.
- Other scopes: `singleton` (default), `request`, `session`, `application`, etc.

Example:

```
@Component
@Primary
@Scope("prototype")
public class Desktop implements Computer {
    public Desktop() {
        System.out.println("Desktop Object Created..");
    }

    @Override
    public void compile() {
        System.out.println("Compiling using Desktop");
    }
}
```

- **Explanation:**

- The `@Scope("prototype")` ensures that a **new instance of Desktop** is created every time it is injected or retrieved from the Spring context.

- The `@Primary` annotation marks this as the default bean for injection when there are multiple implementations of the `Computer` interface.

2. Value Annotation (`@Value`)

- **Purpose:**
Used to inject a specific value into a field, method parameter, or constructor. It can be used to set default values, read values from property files, or inject constant values.
- **Usage:**
 - Can inject literals, expressions, or property values.
 - Format: `@Value("value")`.

Example:

```
@Component
public class Alien {
    @Value("21")
    private int age;

    private Computer com;

    public Alien() {
        System.out.println("Alien Object Created");
    }
}
```

- **Explanation:**
 - The `@Value("21")` injects the value `21` into the `age` field of the `Alien` class.
 - The `Alien` bean will always have `age` set to `21` unless overridden by external configuration.

Key Points to Remember

1. **@Scope("prototype"):**
 - Creates a new instance of the bean for every request.
 - Useful for beans with independent state for each use.
2. **@Value:**
 - Injects values directly into fields.
 - Can read values from property files or inject hardcoded constants.
3. **Practical Usage:**
 - Use **@Scope("prototype")** for beans where state isolation is required.
 - Use **@Value** for initializing fields with constant or configurable values.