04 - Scopes

In Spring, beans can have different scopes that define how many instances of the bean are created and how they are shared across the application.

👉 **Types of Bean Scopes:**

1. **Singleton** (Default Scope):
   - In the Singleton scope, only one instance of the bean is created, even if there are multiple references to it.
   - Every time we retrieve the bean from the Spring container, we will get the same instance.
2. **Prototype**:
   - In the Prototype scope, a new object is created each time when we call the getBean() method.
   - Each reference will point to a distinct instance of the bean.
3. **Request** (for Spring Web):
   - In the Request scope, a new bean instance is created for every HTTP request. This scope is available in web applications.
4. **Session** (for Spring Web):
   - In the Session scope, a new bean instance is created for every HTTP session. This scope is specific to web applications.

★ In Spring, beans are **Singleton by default**. This means that only one instance of the bean is created and shared across the entire application context.

Example:
App.java

```
public class App
{
   public static void main( String[] args )
   {


```

```
ApplicationContext context = new
ClassPathXmlApplicationContext("spring.xml");  // create a container


        Alien obj1 = (Alien) context.getBean("alien1");
        obj1.age = 21;
        System.out.println(obj1.age);
        Alien obj2 = (Alien) context.getBean("alien1");
        System.out.println(obj2.age);
    }
}
```

Alien.java

```
int age;
public Alien() {
        System.out.println("Object Created");
}
public void code() {
        System.out.println("Coding");
}
```

Spring.xml

```
<bean id="alien1" class="com.telusko.Alien">
</bean>
<bean id="lap" class="com.telusko.Laptop">
</bean>
```

Output:

```
Object Created
Laptop Object Created
21
21
```

👉 Changing the Scope in XML:

We can explicitly define the scope of a bean using the scope attribute in the <bean> tag.

Singleton:

```
<bean id="alien1" class="com.telusko.Alien" />
                Or
<bean id="alien1" class="com.telusko.Alien" scope="singleton" />
```

- In this case, the alien1 bean will be Singleton by default.

Prototype:

```
<bean id="alien1" class="com.telusko.Alien" scope="prototype" />
```

- Here, alien1 bean is defined with a prototype scope.
- In this, a new object will be created for each call to getBean().

```
Laptop Object Created
Object Created
21
Object Created
0
```

➤ The **Prototype scope** is useful when we need a new instance of the bean for each operation, such as in multi-threaded environments, or when dealing with stateless objects that require frequent instantiation.

Code Link:

https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/2%20Exploring%20Spring%20Framework/2.4%20Scopes/Spring1