Spring bean scopes

- > Spring supports the following 5 types of bean scopes and the default scope is **singleton**
 - 1. singleton Return a single bean instance per Spring IoC container
 - 2. prototype Return a new bean instance each time when requested
 - **3.** request Return a single bean instance per HTTP request.
 - **4.** session Return a single bean instance per HTTP session.
 - **5.** *globalSession* Return a single bean instance per global HTTP session.

Singleton scope example:

26

27

28 29

30

31 32 33

}

1. Create the Engine class

@Override

public class Engine { 12 13 14 private String engineID; 15 private String validity; 16 17 public String getEngineID() { 18 return engineID; 19 20 public void setEngineID(String engineID) { 21 this.engineID = engineID; 22 23 public String getValidity() { 24 return validity; 25

2. Configure Engine class in Spring-Core.xml configuration file.

public void setValidity(String validity) {

this.validity = validity;

public String toString() {

```
10
     <beans xmlns="http://www.springframework.org/schema/beans"</pre>
11
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
12
         xsi:schemaLocation="http://www.springframework.org/schema/beans
13
         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
14
15
         <bean id="engine" class="com.java2learn.core.Engine" scope="singleton">
16
             cproperty name="engineID" value="123"></property>
17
             cproperty name="validity" value="3Yrs."></property>
18
         </bean>
19 </beans>
```

return "Engine [engineID=" + engineID + ", validity=" + validity + "]";

3. Create the client and run it.

```
10
       package com.java2learn.app;
 11
 12
       import org.springframework.context.ApplicationContext;
 13
       import org.springframework.context.support.ClassPathXmlApplicationContext;
 14
       import com.java2learn.core.Engine;
 15
       public class ApplicationContextDemo {
 16
 17
           public static void main(String[] args) {
 18
ApplicationContext applicationContext=new
ClassPathXmlApplicationContext("Spring-Core.xml");
  21
                Engine engine=(Engine)applicationContext.getBean("engine");
  22
                System.out.println("Validity::"+engine.getValidity());
  23
                //setting value to the validaity.
  24
                engine.setValidity("10 Yrs.");
                System.out.println("Validity::"+engine.getValidity());
System.out.println("Engine ID::"+engine.getEngineID());
  25
  26
  27
  28
                //retrieve bean again
  29
                Engine engine1=(Engine)applicationContext.getBean("engine");
  30
                System.out.println("Validity::"+engine1.getValidity());
  31
            }
  32
       }
```

> Output:

Validity::3Yrs. Validity::10 Yrs. Engine ID::123 Validity::10 Yrs.

Singleton scope example:

1. Create the Engine class

```
12
     public class Engine {
13
14
         private String engineID;
15
         private String validity;
16
17
         public String getEngineID() {
             return engineID;
18
19
20
         public void setEngineID(String engineID) {
21
             this.engineID = engineID;
22
23
         public String getValidity() {
24
             return validity;
25
26
         public void setValidity(String validity) {
27
             this.validity = validity;
28
29
         @Override
         public String toString() {
31
             return "Engine [engineID=" + engineID + ", validity=" + validity + "]";
32
         }
33
```

2. Configure Engine class in Spring-Core.xml configuration file.

```
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
11
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
12
           xsi:schemaLocation="http://www.springframework.org/schema/beans
13
           http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
 14
 15
           <bean id="engine" class="com.java2learn.core.Engine" scope="singleton">
16
               cproperty name="engineID" value="123"></property>
               property name="validity" value="3Yrs."></property>
17
 18
           </bean>
19
      </beans>
3. Create the client and run it.
       import org.springframework.context.ApplicationContext;
 13
       import org.springframework.context.support.ClassPathXmlApplicationContext;
 14
       import com.java2learn.core.Engine;
 15
       public class ApplicationContextDemo {
 16
 17
           public static void main(String[] args) {
ApplicationContext applicationContext=new
ClassPathXmlApplicationContext("Spring-Core.xml");
 21
               Engine engine=(Engine)applicationContext.getBean("engine");
 22
               System.out.println("Validity::"+engine.getValidity());
               //setting value to the validaity.
 23
               engine.setValidity("10 Yrs.");
System.out.println("Validity::"+engine.getValidity());
System.out.println("Engine ID::"+engine.getEngineID());
 24
 25
 26
 27
 28
               //retrieve bean again
               Engine engine1=(Engine)applicationContext.getBean("engine");
 29
```

System.out.println("Validity::"+engine1.getValidity());

Output:

30

31

32 }

Validity::3Yrs. Validity::10 Yrs. Engine ID::123 Validity::10 Yrs.

}

Prototype scope example:

1. Create the Engine class

```
public class Engine {
13
14
         private String engineID;
15
         private String validity;
16
17
         public String getEngineID() {
18
             return engineID;
19
20
         public void setEngineID(String engineID) {
21
             this.engineID = engineID;
22
23
         public String getValidity() {
24
             return validity;
25
         public void setValidity(String validity) {
27
             this.validity = validity;
29
         @Override
         public String toString() {
             return "Engine [engineID=" + engineID + ", validity=" + validity + "]";
31
32
33
    }
```

2. Configure Engine class in Spring-Core.xml configuration file.

```
cbeans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans-3.0.xsd">

cbean id="engine" class="com.java2learn.core.Engine" scope="prototype"/>
cbeans>
```

3. Create the client and run it.

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.java2learn.core.Engine;
public class ApplicationContextDemo {
    public static void main(String[] args) {
```

ApplicationContext applicationContext=new
ClassPathXmlApplicationContext("Spring-Core.xml");

```
21
             Engine engine=(Engine)applicationContext.getBean("engine");
22
             //setting value to the validaity.
23
             engine.setValidity("10 Yrs.");
             System.out.println("Validity::"+engine.getValidity());
24
25
26
             //retrieve bean again
27
             Engine engine1=(Engine)applicationContext.getBean("engine");
             System.out.println("Validity::"+engine1.getValidity());
28
29
         }
30 }
```

\triangleright

Output: Validity::10 Yrs. Validity::null