# BEAN ALIASING

SpringTutors January 10, 2016 Features 125 Views

Bean Aliasing

In spring when we configure a class as bean we declare an id with which we want to retrieve it back from the IOC container. Along with id we can attach multiple names to the beans and these names act as alias names with which can look up the bean from the container. Giving multiple name to a bean is called bean aliasing.

## How does it work

There are two ways to specify the multiple name to bean-

1.Using “name” attribute at <bean> tag level.

2.Using <alias-name> tag.

Let’s see an example-

```
package com.ba;
class Student
{
private int sid;
private String sname;
public void setSid(int sid)
{
this.sid=sid;
}
public void setSname(String sname)
{
this.sname=sname;
}
public String toString() {
 return "Student [sid=" + sid + ",sname=" + sname+"]";
}
}
```

## Spring bean configuration file application-context.xml

```
<bean id=”student” class=”com.ba.Student”>
<property name="sid" value="10"/>
<property name="sname" value="jhon"/>
</bean>
```

## Using name attribute

```
<bean id=”student” name=”student1, student2” class=”com.ba.Student”/>
<property name="sid" value="10"/>
<property name="sname" value="jhon"/>
</bean>
```

## Using <alias-name> tag

```
<beans>
```

```
<bean id="student" class="com.ba.Student">
<property name="sid" value="10"/>
<property name="sname" value="jhon"/>
</bean>
<alias-name="student" alias="student3"/>
<alias-name="student3" alias="student4"/>
<alias-name="student2" alias="student5"/>
</beans>
```

In the above configuration we can see that <alias-name> is used as a tag outside of <bean> tag.

We can use both name attribute and <alias-name> in spring bean configuration file. Now IOC container can create the object of Student class by seeing any of the bean name student or student1 or student2 or student4 or student5.

```
package com.ba.Student;
import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;
```

```
public class BATest {
public static void main(String[] args)
{
 BeanFactory factory=new XmlBeanFactory(new ClassPathResource("application-context.xml"));
 Student s=factory.getBean("student",Student.class);
Student s1=factory.getBean("student1",Student.class);
Student s2=factory.getBean("student4",Student.class);
 System.out.println(s);
System.out.println(s1);
 System.out.println(s2);
}
}
```

any of the above configured name of bean can be used it will give same result.

## Difference between name and <alias-tag>

When we give comma(,) in case of name attribute between two name it acts as a separator.

While in case of alias if we give comma(,) it will be considered as the part of name.

Example:-

<bean id="student" name="student1, student2" class="com.ba.Student"/>

In this case there are three name of the bean student, student1, and student2. Here comma(,) separate the name of bean.

<bean id="student" class="com.ba.Student"/>

<alias-name="student" alias="student3,student4"/>

In this case there are only two names of the bean student and student3,student4. Here commoa (,) becomes the part of bean name.

## When Bean Aliasing is used

Let's see a real time use case as example:-

We know that Amazon is an e-commerce company. Now amazon is not directly providing services to

customer it has hired courier supplier vendors like BlueDart and Dtdc. Let's suppose the request for delivering the item in metro city is taken care by bluedart and for small city or town delivery of items is taken care by Dtdc. Now suppose for any reason bluedart came to amazon and tells that he will not be able to deliver the products to customer for one month. In this case amazon will have to face huge loss. To avoid the loss first he will talk to Dtdc if he can do the work of bluedart also. Dtdc agrees to do that work. Now all the request for delivering items comes from metro city for Bludart should go to Dtdc, for this application developer has to change the code and we know that changing the code is very costlier. To avoid the change in code spring has given the facility called bean aliasing. Now for one bean we can give any number of names-

```
<beans>
<bean id="amazon" class="Amazon">
<property name="address" ref="bluedart">
<bean id="bluedart" class="BlueDart"/>
<bean id="dtdc"class="Dtdc"/>
<alias name="dtdc" alias="bluedart">
</beans>
```

By configuring alias name as above whatever be the request comes from amazon to deliver the items it will goes to Dtdc. In this way we can see without touching the code through alias name we can solve the problem.