

Dependency Injection by Constructor Example

We can inject the dependency by constructor. The **<constructor-arg>** subelement of **<bean>** is used for constructor injection. Here we are going to inject

1. primitive and String-based values
2. Dependent object (contained object)
3. Collection values etc.

Injecting primitive and string-based values

Let's see the simple example to inject primitive and string-based values. We have created three files here:

- Employee.java
- applicationContext.xml
- Test.java

Employee.java

It is a simple class containing two fields id and name. There are four constructors and one method in this class.

```
package com.javatpoint;

public class Employee {
    private int id;
    private String name;

    public Employee() {System.out.println("def cons");}

    public Employee(int id) {this.id = id;}

    public Employee(String name) { this.name = name;}

    public Employee(int id, String name) {
```

↑ SCROLL TO TOP

```
    this.id = id;
    this.name = name;
}

void show(){
    System.out.println(id+ " "+name);
}

}
```



applicationContext.xml

We are providing the information into the bean by this file. The constructor-arg element invokes the constructor. In such case, parameterized constructor of int type will be invoked. The value attribute of constructor-arg element will assign the specified value. The type attribute specifies that int parameter constructor will be invoked.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

    <bean id="e" class="com.javatpoint.Employee">
        <constructor-arg value="10" type="int"></constructor-arg>
    </bean>

</beans>
```

↑ SCROLL TO TOP

This class gets the bean from the applicationContext.xml file and calls the show method.

```
package com.javatpoint;

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.*;

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

        Resource r=new ClassPathResource("applicationContext.xml");
        BeanFactory factory=new XmlBeanFactory(r);

        Employee s=(Employee)factory.getBean("e");
        s.show();

    }
}
```

Output:10 null

[download this example](#)

Injecting string-based values

If you don't specify the type attribute in the constructor-arg element, by default string type constructor will be invoked.

```
....
<bean id="e" class="com.javatpoint.Employee">
<constructor-arg value="10"></constructor-arg>
</bean>
....
```

If you change the bean element as given above, string parameter constructor will be invoked and the output will be 0 10

Output:0 10

You may also pass the string literal as following:

```
....
<bean id="e" class="com.javatpoint.Employee">
```

↑ SCROLL TO TOP

```
<constructor-arg value="Sonoo"> </constructor-arg>  
</bean>  
....
```

Output:0 Sonoo

You may pass integer literal and string both as following

```
....  
<bean id="e" class="com.javatpoint.Employee">  
<constructor-arg value="10" type="int" > </constructor-arg>  
<constructor-arg value="Sonoo"> </constructor-arg>  
</bean>  
....
```

Output:10 Sonoo

[download this example \(developed using Myeclipse IDE\)](#)

[download this example \(developed using Eclipse IDE\)](#)

← Prev

Next →



For Videos Join Our Youtube Channel: [Join Now](#)

↑ SCROLL TO TOP

