

Dependency Injection by setter method

We can inject the dependency by setter method also. The **<property>** subelement of **<bean>** is used for setter 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 by setter method

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

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

Employee.java

It is a simple class containing three fields id, name and city with its setters and getters and a method to display these informations.

```
package com.javatpoint;
```

```
public class Employee {
```

```
private int id;
private String name;
private String city;

public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}

public String getCity() {
    return city;
}
public void setCity(String city) {
    this.city = city;
}
void display(){
    System.out.println(id+" "+name+" "+city);
}

}
```

applicationContext.xml

We are providing the information into the bean by this file. The property element invokes the setter method. The value subelement of property will assign the specified value.

```
<?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="obj" class="com.javatpoint.Employee">
<property name="id">
<value>20</value>
</property>
<property name="name">
<value>Arun</value>
</property>
<property name="city">
<value>ghaziabad</value>
</property>

</bean>

</beans>
```

Test.java

This class gets the bean from the applicationContext.xml file and calls the display 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 e=(Employee)factory.getBean("obj");
        s.display();

    }
```

```
}
```

Output:20 Arun ghaziabad

download this example (developed using MyEclipse IDE)

download this example (developed using Eclipse IDE)

← prev

next →

Please Share



Learn Latest Tutorials



DS



PDFBox



Ant



JMeter



jBPM