

## Setter Injection

- The setter-based injection uses the setter property in beans for injecting values. Primitive values and reference values can be injected through the setter property.

- **Example:**

1. Create the Company POJO.

```
12 public class Company {
13
14     private String name;
15     private String compnayType;
16     private Employee employee;
17
18     public Employee getEmployee() {
19         return employee;
20     }
21     //setting employee as a setter method
22     public void setEmployee(Employee employee) {
23         this.employee = employee;
24     }
25
26     public String getName() {
27         return name;
28     }
29     public void setName(String name) {
30         this.name = name;
31     }
32     public String getCompnayType() {
33         return compnayType;
34     }
35     public void setCompnayType(String compnayType) {
36         this.compnayType = compnayType;
37     }
38 }
```

2. Create the Employee POJO.

```
12 public class Employee {
13
14     private String name;
15     private int age;
16     private double salary;
17
18     public String getName() {
19         return name;
20     }
21     public void setName(String name) {
22         this.name = name;
23     }
24     public int getAge() {
25         return age;
26     }
27     public void setAge(int age) {
28         this.age = age;
29     }
30 }
```

```

29     }
30     public double getSalary() {
31         return salary;
32     }
33     public void setSalary(double salary) {
34         this.salary = salary;
35     }
36 }
37 }

```

### 3. Add Company and Employee POJO's in the Spring configuration file.

```

10 <beans xmlns="http://www.springframework.org/schema/beans"
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="employee" class="com.java2learn.core.Employee">
16         <property name="name" value="Moon"></property>
17         <property name="age">
18             <value>10</value>
19         </property>
20         <property name="salary" value="350000"></property>
21     </bean>
22
23     <bean id="company" class="com.java2learn.core.Company">
24         <property name="name" value="Oracle"></property>
25         <property name="compnayType" value="serviceBased"></property>
26         <property name="employee" ref="employee"></property>
27     </bean>
28
29 </beans>

```

### 4. Create the client and run it.

```

12 import org.springframework.context.ApplicationContext;
13 import org.springframework.context.support.FileSystemXmlApplicationContext;
14
15 import com.java2learn.core.Company;
16
17
18 public class ApplicationContextDemo {
19
20     public static void main(String[] args) {
21         //ApplicationContext applicationContext=new
22         ClassPathXmlApplicationContext("Spring-Core.xml");
23         //(or)
24         ApplicationContext applicationContext=new
25         FileSystemXmlApplicationContext("classpath:Spring-Core.xml");
26         Company company=(Company) applicationContext.getBean("company");
27         System.out.println("Company name::"+company.getName());
28         System.out.println("Employee
29         name::"+company.getEmployee().getName());
30         System.out.println("Employee
31         Salary::"+company.getEmployee().getSalary());
32     }
33 }

```

➤ **Output:**

```
Company name::Oracle  
Employee name::Moon  
Employee Salary::350000.0
```