

09 - Autowiring

Autowiring is a mechanism used by Spring to inject dependencies into beans automatically.

Wiring refers to connecting beans by specifying dependencies. It is done by manually specifying references using the **ref** attribute in XML.

Autowiring allows Spring to automatically inject dependencies by analyzing the bean's name or type, without requiring explicit **ref** properties in the XML configuration.

Spring provides multiple autowiring modes:

- **byName**: Autowires the bean by matching the property name with the bean id.
- **byType**: Autowires the bean by matching the property type with the bean type.
- **constructor**: Autowires the bean via constructor injection.
- **autodetect**: Tries constructor first, then falls back to **byType**.

👉 Autowiring by Name:

In **byName** autowiring, Spring looks for a bean in the container with the same name as the property. If such a bean is found, Spring injects it automatically.

Example:

```
<bean id="alien1" class="com.telusko.Alien" autowire="byName">
  <property name="age" value="21" />
  <!-- <property name="comp" ref="lap1" /> -->
</bean>

<bean id="comp" class="com.telusko.Laptop"></bean>
```

- Here, the Alien bean depends on the Computer interface. In the Alien class, there is a comp property. Spring will automatically wire the Laptop bean because it has an id of comp, which matches the property name.

Output:

```
21
Coding
Compiling using Laptop
```

★ If we explicitly specify a property in the bean definition, Spring will prioritize this over the autowiring mechanism.

```
<bean id="alien1" class="com.telusko.Alien" autowire="byName">
  <property name="age" value="21" />
  <property name="comp" ref="comp1" />
</bean>

<bean id="comp" class="com.telusko.Laptop"></bean>
<bean id="comp1" class="com.telusko.Desktop"></bean>
```

Output:

```
21
Coding
Compiling using Desktop
```

👉 **Autowiring by Type:**

In **byType** autowiring, Spring looks for a bean with a type that matches the property type in the dependent bean. If such a bean is found, it is injected automatically.

Example:

```
<bean id="alien1" class="com.telusko.Alien" autowire="byType">
```

```
<property name="age" value="21" />
</bean>
```

```
<bean id="comp" class="com.telusko.Laptop"></bean>
```

- In this, Spring injects the Laptop bean into the comp property because the Laptop class implements the Computer interface, and the comp property in the Alien class is of type Computer.

Output:

```
21
Coding
Compiling using Laptop
```

★ If there are multiple beans of the same type in the Spring container, autowiring by type will cause a conflict, and Spring will throw an error. To resolve this, we can switch to **byName** autowiring, or primary bean, or provide an explicit reference.

Code Link:

<https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/2%20Exploring%20Spring%20Framework/2.9%20Autowiring/Spring1>