**Bean Auto wiring**

In spring framework, you can wire beans automatically with auto-wiring feature. To enable it, just define the “**autowire**” attribute in <bean>.

<bean id="customer" class="com.mkyong.common.Customer" autowire="byName" />

In Spring, 5 Auto-wiring modes are supported.

* no – Default, no auto wiring, set it manually via “ref” attribute
* byName – Auto wiring by property name. If the name of a bean is same as the name of other bean property, auto wire it.
* byType – Auto wiring by property data type. If data type of a bean is compatible with the data type of other bean property, auto wire it.
* constructor – byType mode in constructor argument.
* autodetect – If a default constructor is found, use “autowired by constructor”; Otherwise, use “autowire by type”.

## Examples

A Customer and Person object for auto wiring demonstration.

package com.mkyong.common;

public class Customer

{

private Person person;

public Customer(Person person) {

this.person = person;

}

public void setPerson(Person person) {

this.person = person;

}

//...

}

package com.mkyong.common;

public class Person

{

//...

}

**1. Auto-Wiring ‘no’**

This is the default mode, you need to wire your bean via ‘ref’ attribute.

<bean id="customer" class="com.mkyong.common.Customer">

<property name="person" ref="person" />

</bean>

<bean id="person" class="com.mkyong.common.Person" />

**2. Auto-Wiring ‘byName’**

Auto-wire a bean by property name. In this case, since the name of “person” bean is same with the name of the “customer” bean’s property (“person”), so, Spring will auto wired it via setter method – “setPerson(Person person)“.

<bean id="customer" class="com.mkyong.common.Customer" autowire="byName" />

<bean id="person" class="com.mkyong.common.Person" />

**3. Auto-Wiring ‘byType’**

Auto-wire a bean by property data type. In this case, since the data type of “person” bean is same as the data type of the “customer” bean’s property (Person object), so, Spring will auto wired it via setter method – “setPerson(Person person)“.

<bean id="customer" class="com.mkyong.common.Customer" autowire="byType" />

<bean id="person" class="com.mkyong.common.Person" />

**4. Auto-Wiring ‘constructor’**

Auto-wire a bean by property data type in constructor argument. In this case, since the data type of “person” bean is same as the constructor argument data type in “customer” bean’s property (Person object), so, Spring auto wired it via constructor method – “public Customer(Person person)“.

<bean id="customer" class="com.mkyong.common.Customer" autowire="constructor" />

<bean id="person" class="com.mkyong.common.Person" />

**5. Auto-Wiring ‘autodetect’**

If a default constructor is found, uses “constructor”; Otherwise, uses “byType”. In this case, since there is a default constructor in “Customer” class, so, Spring auto wired it via constructor method – “public Customer(Person person)“.

<bean id="customer" class="com.mkyong.common.Customer" autowire="autodetect" />

<bean id="person" class="com.mkyong.common.Person" />

**Another Example:**

**Body.java**

**package** com.doj.spring;

**public** **class** Body {

String bodyType;

**public** String getBodyType() {

**return** bodyType;

}

**public** **void** setBodyType(String bodyType) {

**this**.bodyType = bodyType;

}

}

**Engine.java**

**package** com.doj.spring;

**public** **class** Engine {

String engineName;

**public** String getEngineName() {

**return** engineName;

}

**public** **void** setEngineName(String engineName) {

**this**.engineName = engineName;

}

}

**Tyre.java**

**package** com.doj.spring;

**public** **class** Tyre {

String tyreName;

**public** String getTyreName() {

**return** tyreName;

}

**public** **void** setTyreName(String tyreName) {

**this**.tyreName = tyreName;

}

}

**Car.java**

**package** com.doj.spring;

**public** **class** Car {

Engine engine;

Tyre tyre;

Body body;

**public** **void** setEngine(Engine engine) {

**this**.engine = engine;

}

**public** **void** setTyre(Tyre tyre) {

**this**.tyre = tyre;

}

**public** **void** setBody(Body body) {

**this**.body = body;

}

**public** Car(Engine engine, Tyre tyre, Body body) {

**super**();

**this**.engine = engine;

**this**.tyre = tyre;

**this**.body = body;

}

**public** Car() {

**super**();

}

**public** **void** driveCar(){

System.*out*.println("You are driving a car which has a engine "+**this**.engine.engineName

+" has a tyre "+**this**.tyre.getTyreName()+" has a body "+**this**.body.getBodyType());

}

}

**CarDemo.java**

**package** com.doj.spring;

**import** org.springframework.context.support.AbstractApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** CarDemo {

**public** **static** **void** main(String[] args) {

AbstractApplicationContext context = **new** ClassPathXmlApplicationContext("spring.xml");

Car car = (Car) context.getBean("car");

car.driveCar();

context.close();

}

}

**Spring.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<!-- Implicit or Autowiring of Bean Wiring in XML Configuration -->

<!-- bean definitions here -->

<!-- Bean Wiring Types

1. No-Autowiring Mode (explicit) wiring with setter injection- (Previuos example based on it)

2. By Name (Implicit)-(Current Example)

3. By Type

4. By Constructor

5. default

-->

<!-- 2. By Name (Implicit)-(Current Example) -->

<!-- <bean class="com.doj.spring.Car" id="car" autowire="byName"/> -->

<!-- 3. By Type (Implicit)-(Current Example) -->

<!-- <bean class="com.doj.spring.Car" id="car" autowire="byType"/> -->

<!-- 4. Constructor (Implicit)-(Current Example) -->

<!-- <bean class="com.doj.spring.Car" id="car" autowire="constructor"/> -->

<!-- 5. Autodetect or default (Implicit)-(Current Example) -->

<bean class=*"com.doj.spring.Car"* id=*"car"* autowire=*"default"*/>

<!-- STEEL -->

<bean class=*"com.doj.spring.Body"* id=*"body"*>

<property name=*"bodyType"* value=*"STEEL"*></property>

</bean>

<!-- FORD -->

<bean class=*"com.doj.spring.Engine"* id=*"engine"*>

<property name=*"engineName"* value=*"FORD"*></property>

</bean>

<!-- MRF -->

<bean class=*"com.doj.spring.Tyre"* id=*"tyre"*>

<property name=*"tyreName"* value=*"MRF"*></property>

</bean>

</beans>