**Spring Autowiring:**

Till now we have seen that when we need to inject a bean into an another bean either we inject it through setter or by the constructor. But in Spring there is an another approach by which you can inject a bean into another bean automatically without declaring anything. we called it Autowiring.

Spring community’s motto is not forcing the developer to write extra code to be precise boiler plate code, rather concentrate on business logic. Autowire takes one step forward to that convention.

To understand the Autowiring functionality let takes a look what we have done so far to inject a bean.

Step 1. Create POJO classes.

Step 2. Describe the bean in Spring configuration file

Step 3. Initialize every property with an initialized value by Property tag.

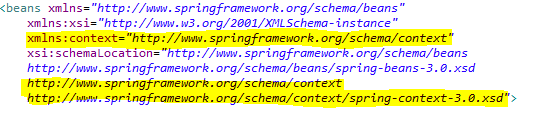
Step 4. Wiring each bean by ref attribute or by constructor-args.

So, if a class has many beans associated with it. We have to associate every bean by ref attribute. So for a long class, it is cumbersome not only that we can identify that as a boilerplate code.

Spring Autowire comes to rescue from that when we introduce Autowiring we don’t have to perform step 4. So we not need to wire beans by ref property, Spring container will take care for the same.

To introduce Autowiring we need to perform following changes in our Spring configuration XML , and Spring bean java file.

1.



In beans tag, we introduce a new schema context and define the path where it’s XSD(XML Schema Definition) is located

2. Please add this line in Spring definition file,

By this command we enable Spring container’s Search for annotation capability, if Spring found any annotation with name @Autowired it injects the bean definition into it.

3. Put @Autowired annotation on top of the properties, I n runtime Spring injected Actual Bean into that property.



**Problem Statement: We will create an IPhone and autowire the camera facilities into IPhone.**

So, We will create two beans, The IPhone and camera then inject camera by Autowiring.

**IPhone.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"*

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<context:annotation-config/>

<bean id=*"iphone"* class=*"com.example.autowire.IPhone"*>

<property name=*"version"* value=*"IPhone6"*/>

<property name=*"ram"* value=*"2"*/>

<property name=*"iosVersion"* value=*"IOS6"*/>

</bean>

<bean id=*"rearCamera"* class=*"com.example.autowire.RearCamera"*>

<property name=*"description"* value=*"18-210 lens with Apperature f1.2"*/>

<property name=*"megaPixel"* value=*"1024px"*/>

</bean>

</beans>

**Java Code :**

**package** com.example.autowire;

**public** **interface** ICamera {

String getDescription();

String getMegaPixel();

}

**package** com.example.autowire;

**public** **class** RearCamera **implements** ICamera{

**private** String description;

**private** String megaPixel;

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** **void** setMegaPixel(String megaPixel) {

**this**.megaPixel = megaPixel;

}

@Override

**public** String getDescription() {

// **TODO** Auto-generated method stub

**return** description;

}

@Override

**public** String getMegaPixel() {

// **TODO** Auto-generated method stub

**return** megaPixel;

}

@Override

**public** String toString() {

**return** "RearCamera [description=" + description + ", megaPixel="

+ megaPixel + "]";

}

}

**package** com.example.autowire;

**import** org.springframework.beans.factory.annotation.Autowired;

**public** **class** IPhone {

String version;

**int** ram;

String iosVersion;

@Autowired

ICamera camera;

**public** String getVersion() {

**return** version;

}

**public** **void** setVersion(String version) {

**this**.version = version;

}

**public** **int** getRam() {

**return** ram;

}

**public** **void** setRam(**int** ram) {

**this**.ram = ram;

}

**public** String getIosVersion() {

**return** iosVersion;

}

**public** **void** setIosVersion(String iosVersion) {

**this**.iosVersion = iosVersion;

}

**public** ICamera getCamera() {

**return** camera;

}

**public** **void** setCamera(ICamera camera) {

**this**.camera = camera;

}

@Override

**public** String toString() {

**return** "IPhone [version=" + version + ", ram=" + ram + "GB , iosVersion="

+ iosVersion + ", camera=" + camera + "]";

}

}

**package** com.example.autowire;

**import** org.springframework.context.ApplicationContext;

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

**public** **class** Manager {

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

ApplicationContext context = **new** ClassPathXmlApplicationContext("configFiles/IPhone.xml");

IPhone iphone = (IPhone) context.getBean("iphone");

System.***out***.println(iphone);

}

}

**Output:**

IPhone [version=IPhone6, ram=2GB , iosVersion=IOS6, camera=RearCamera [description=18-210 lens with Apperature f1.2, megaPixel=1024px]]

***In IPhone class we mention the @Autowired annotation above the camera field so in runtime Spring container search for the actual implementation of the camera and then inject Rear camera into IPhone.***

***NB please note that in IPhone.XML I did not mention the camera property as we injected it through @Autowiring annotation.***

**Autowiring ways:**

We can use Autowiring annotation three different ways.

1. **Field Injection**: Use ***@Autowired annotation*** above the java field/properties. If we use field injection there is no need to provide Setter for that field as Spring container inject the bean in field level.

**2. Setter Injection:** Use ***@Autowired annotation*** above the Setter of java field/property. Spring will inject the bean by Setter method.

**Code snippet:**

**package** com.example.autowire;

**import** org.springframework.beans.factory.annotation.Autowired;

**public** **class** IPhone {

String version;

**int** ram;

String iosVersion;

ICamera camera;

**public** String getVersion() {

**return** version;

}

**public** **void** setVersion(String version) {

**this**.version = version;

}

**public** **int** getRam() {

**return** ram;

}

**public** **void** setRam(**int** ram) {

**this**.ram = ram;

}

**public** String getIosVersion() {

**return** iosVersion;

}

**public** **void** setIosVersion(String iosVersion) {

**this**.iosVersion = iosVersion;

}

**public** ICamera getCamera() {

**return** camera;

}

@Autowired

**public** **void** setCamera(ICamera camera) {

**this**.camera = camera;

}

@Override

**public** String toString() {

**return** "IPhone [version=" + version + ", ram=" + ram + "GB , iosVersion="

+ iosVersion + ", camera=" + camera + "]";

}

}

Please look the Setter method of the camera property here I mention the @Autowired annotation.

**Constructor Injection:**

Use ***@Autowired annotation*** above the constructor. Spring will inject the bean when container calls the constructor.

**Code snippet:**

**package** com.example.autowire;

**import** org.springframework.beans.factory.annotation.Autowired;

**public** **class** IPhone {

String version;

**int** ram;

String iosVersion;

ICamera camera;

@Autowired

Public IPhone(Icamera camera)

{

this.camera=camera;

}

**public** String getVersion() {

**return** version;

}

**public** **void** setVersion(String version) {

**this**.version = version;

}

**public** **int** getRam() {

**return** ram;

}

**public** **void** setRam(**int** ram) {

**this**.ram = ram;

}

**public** String getIosVersion() {

**return** iosVersion;

}

**public** **void** setIosVersion(String iosVersion) {

**this**.iosVersion = iosVersion;

}

**public** ICamera getCamera() {

**return** camera;

}

**public** **void** setCamera(ICamera camera) {

**this**.camera = camera;

}

@Override

**public** String toString() {

**return** "IPhone [version=" + version + ", ram=" + ram + "GB , iosVersion="

+ iosVersion + ", camera=" + camera + "]";

}

}

**Modes of Autowiring:**

**There are two ways we can use Autowire feature.**

1. **Anotation Based** : When we use @Autowired annotation.
2. **By XML** : Also we can provide autowire property into bean tag. Bu doing so Spring container autwire the bean via **Setter method.**

**BY XML** Example :

**IPhone.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"*

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<context:annotation-config/>

<bean id=*"iphone"* class=*"com.example.autowire.IPhone"* autowire=*"byType"*>

<property name=*"version"* value=*"IPhone6"*/>

<property name=*"ram"* value=*"2"*/>

<property name=*"iosVersion"* value=*"IOS6"*/>

</bean>

<bean id=*"camera"* class=*"com.example.autowire.RearCamera"*>

<property name=*"description"* value=*"18-210 lens with Apperature f1.2"*/>

<property name=*"megaPixel"* value=*"1024px"*/>

</bean>

</beans>

**package** com.example.autowire;

**import** org.springframework.beans.factory.annotation.Autowired;

**public** **class** IPhone {

String version;

**int** ram;

String iosVersion;

ICamera camera;

**public** String getVersion() {

**return** version;

}

**public** **void** setVersion(String version) {

**this**.version = version;

}

**public** **int** getRam() {

**return** ram;

}

**public** **void** setRam(**int** ram) {

**this**.ram = ram;

}

**public** String getIosVersion() {

**return** iosVersion;

}

**public** **void** setIosVersion(String iosVersion) {

**this**.iosVersion = iosVersion;

}

**public** ICamera getCamera() {

**return** camera;

}

**public** **void** setCamera(ICamera camera) {

**this**.camera = camera;

}

@Override

**public** String toString() {

**return** "IPhone [version=" + version + ", ram=" + ram + "GB , iosVersion="

+ iosVersion + ", camera=" + camera + "]";

}

}

**package** com.example.autowire;

**public** **interface** ICamera {

String getDescription();

String getMegaPixel();

}

**package** com.example.autowire;

**public** **class** RearCamera **implements** ICamera{

**private** String description;

**private** String megaPixel;

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** **void** setMegaPixel(String megaPixel) {

**this**.megaPixel = megaPixel;

}

@Override

**public** String getDescription() {

// **TODO** Auto-generated method stub

**return** description;

}

@Override

**public** String getMegaPixel() {

// **TODO** Auto-generated method stub

**return** megaPixel;

}

@Override

**public** String toString() {

**return** "RearCamera [description=" + description + ", megaPixel="

+ megaPixel + "]";

}

}

Look at the XML, here we define an attribute called “autowire” and set the value to “byType” by doing so we tell container to inject the camera bean to a field whose type is matched with camera type. Spring container injects RearCamera into camera property.

Next chapter we will discuss Different Types of Autowiring.