**Spring Autowiring Type**

In the previous chapter we learn about Autoworing. In this chapter we will discuss the different ways we can do the Autowiring.

**There are 5 modes in Autowiring but 3 specifiers are available.**

**Pictorial view of different Autowiring mode**

Now it is time to makes our hands dirty.

We will use the same IPhone and Camera Example, we used earlier chapter

**Common Classes and Interfaces:**

**package** com.example.autowire;

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

String getDescription();

String getMegaPixel();

}

**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** **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.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);

}

}

Example, **byname**

Spring try to match java property name with Spring bean Id.

So, in this example IPhone has property name camera now Spring try to search Spring config file with bean id camera. It founds RearCamera and wire the same with IPhone.

**In Spring-Config file:**

Spring Config file (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=*"byName"*>

<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>

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

In iphone bean, we use declare autowire**=”byName”**

Example, **byType**

Spring try to match datatype of java property name with Spring bean type.

So, in this example IPhone has property name camera whose datatype is ICamera now Spring try to search Spring config file type ICamera. It founds RearCamera as it implements ICamera, and wire the same with IPhone.

**NB : If Multiple beans with same datatype present in Spring-config file fatal error occurred**

**In Spring-Config file:**

Spring Config file (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>

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

In iphone bean, we use declare autowire**=”byType”**

Example, **Constructor**

Spring try to match datatype of java property name with Spring bean type.

So, in this example IPhone has property name camera whose datatype is ICamera now Spring try to search Spring config file type ICamera. It founds RearCamera as it implements ICamera, and wire the same with IPhone.

**NB : If Multiple beans with same Datatype present in Spring-config file, fatal error occurred**

**In Spring-Config file:**

Spring Config file (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=*"constructor"*>

<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>

Java

**package** com.example.autowire;

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

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

String version;

**int** ram;

String iosVersion;

ICamera camera;

**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;

}

@Override

**public** String toString() {

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

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

}

}

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

In iphone bean, we use declare autowire**=”constructor”**

**Now, we will see if multiple beans are in Spring config and we Autowire by Type , then Spring throws any exception or not?**

To replicate that problem, we will add another bean into Spring-config file Let we call it, Front Camera.

**Modified 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>

<bean id=*"fcamera"* class=*"com.example.autowire.FrontCamera"*>

<property name=*"description"* value=*"VGA camera"*/>

<property name=*"megaPixel"* value=*"480px"*/>

</bean>

</beans>

**New FrontCamera java class**

**package** com.example.autowire;

**public** **class** FrontCamera **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 + "]";

}

}

**Output :**

org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'iphone' defined in class path resource [configFiles/IPhone.xml]: Unsatisfied dependency expressed through bean property 'camera': : No qualifying bean of type [com.example.autowire.ICamera] is defined: expected single matching bean but found 2: camera,fcamera; nested exception is org.springframework.beans.factory.NoUniqueBeanDefinitionException: No qualifying bean of type [com.example.autowire.ICamera] is defined: expected single matching bean but found 2: camera,fcamera

at

Spring Container says that it founds two bean of same datatype now container is confused about which one it takes. So very careful about this scenario when you set Autowire byType.