

只适合针对对象类型的ref属性。降低可读性。

模式	描述
no	默认，不自动装配。
<u>byName</u>	属性名自动装配。Spring 容器在 XML 配置文件中 bean 的自动装配的属性设置为 byName。然后尝试匹配，并且将它的属性与在配置文件中被定义为相同名称的 beans 的属性进行连接。
<u>byType</u>	属性数据类型自动装配。Spring 容器在 XML 配置文件中 bean 的自动装配的属性设置为 byType。如果 类型 匹配到配置文件中的一个 bean 名称，将尝试匹配和连接属性的类型。如果存在不止一个这样的 bean，则一个致命的异常将会被抛出。
<u>constructor</u>	类似于 byType，但该类型适用于构造函数参数类型。如果在容器中没有一个构造函数参数类型的 bean，则一个致命错误将会发生。
autodetect	Spring 首先尝试通过 constructor 使用自动装配来连接，如果它不执行，Spring 尝试通过 byType 来自动装配。

```
<bean id=" " class=" " autowire="byName">
```

ioc 容器中有一个 bean 的 id 为该 bean 的类的属性名，则自动将该 bean 装载进去。ps. byName 本质 byId。例子：

```
<bean id="textEditor" class="com.tutorialspoint.TextEditor">
    <property name="spellChecker" ref="spellChecker" />
    <property name="name" value="Generic Text Editor" />
</bean>
```

```
<bean id="spellChecker" class="com.tutorialspoint.SpellChecker">
</bean>
```

使用自动装配 “byName”：

```
<bean id="textEditor" class="com.tutorialspoint.TextEditor"
    autowire="byName">
    <property name="name" value="Generic Text Editor" />
</bean>
```

```
<bean id="spellChecker" class="com.tutorialspoint.SpellChecker">
</bean>
```

```
<bean id=" " class=" " autowire="byType">
```

看bean的class和该bean的ref属性类型是否一致。如果有两个id不同class类相同的bean，则报错。例子：

```
<bean id="textEditor" class="com.tutorialspoint.TextEditor">
    <property name="spellChecker" ref="spellChecker" /> //看
spellChecker的类型
    <property name="name" value="Generic Text Editor" />
</bean>
```

```
<bean id="spellChecker" class="com.tutorialspoint.SpellChecker">
</bean>
```

使用自动装配 “byName”：

```
<bean id="textEditor" class="com.tutorialspoint.TextEditor"
    autowire="byName">
    <property name="name" value="Generic Text Editor" />
</bean>
```

```
<bean id="spellChecker" class="com.tutorialspoint.SpellChecker">
</bean>
```

```
<bean id=" " class=" " autowire="constructor">
```

看bean的class和该bean的构造方法的参数类型是否一致。和byType类型类似。例子：

```
<!-- Definition for textEditor bean -->
<bean id="textEditor" class="com.tutorialspoint.TextEditor">
    <constructor-arg ref="spellChecker" /> //ref属性的
类型
```

```

        <constructor-arg value="Generic Text Editor"/>
    </bean>

    <!-- Definition for spellChecker bean -->
    <bean id="spellChecker" class="com.tutorialspoint.SpellChecker">
    </bean>

```

使用自动装配 “by constructor”：

```

    <bean id="textEditor" class="com.tutorialspoint.TextEditor"
        autowire="constructor">
        <constructor-arg value="Generic Text Editor"/>
    </bean>

    <!-- Definition for spellChecker bean -->
    <bean id="SpellChecker" class="com.tutorialspoint.SpellChecker">
    </bean>

```

全部统一设置自动装配：<beans
 defalut-autowire="..">

自动装配的局限性

限制	描述
重写的可能性	你可以使用总是重写自动装配的 <constructor-arg>和 <property> 设置来指定依赖关系。
原始数据类型	你不能自动装配所谓的简单类型包括基本类型，字符串和类。
混乱的本质	自动装配不如显式装配精确，所以如果可能的话尽可能使用显式装配。