**常用注解：**

@AutoWired 自动装配（DI），byName和byType最常见，需要在配置文件中注册AutoWiredAnnotationBeanPostProcessor。xml配置中有个对应的autowire属性，常见的有byType和byName两种，但是在找不到匹配Bean时不予理会，因此@Required应运而生，它需要注册RequiredAnnotationBeanPostProcessor。

@Inject，JSR330定义，可以作用在构造器、属性、方法上，可以设置required=false。

@Qualifier，当有多个符合条件的装配者时，使用该注解指定。相当于配置中的ref。

要使这些注解生效，得在配置文件中注册相应的BeanPostProcessor，也可以简化配置为：<context:annotation-config/>，注意引入相应的命名空间。

**Bean配置：**

静态工厂：

public class HelloApiStaticFactory {

//工厂方法

public static HelloApi newInstance(String message) {

//返回需要的Bean实例

return new HelloImpl2(message);

}

}

<!-- 使用静态工厂方法 -->

<bean id="bean3" class="cn.javass.spring.chapter2.HelloApiStaticFactory" factory-method="newInstance">

<constructor-arg index="0" value="Hello Spring!"/>

</bean>

通过factory-method方法创建实例，可以注入参数。问题：不知道创建出来的实例类型。

实例工厂：

package cn.javass.spring.chapter2;

public class HelloApiInstanceFactory {

public HelloApi newInstance(String message) {

return new HelloImpl2(message);

}

}

<!—1、定义实例工厂Bean -->

<bean id="beanInstanceFactory"

class="cn.javass.spring.chapter2.HelloApiInstanceFactory"/>

<!—2、使用实例工厂Bean创建Bean -->

<bean id="bean4"

factory-bean="beanInstanceFactory"

factory-method="newInstance">

<constructor-arg index="0" value="Hello Spring!"></constructor-arg>

</bean>

因为是实例方法，必须使用factory-bean单独指定工厂类。

**循环依赖：**

循环调用无解，死循环。

singleton的Bean可以循环依赖，因为Spring会将之缓存；prototype的Bean无法循环依赖，因为不缓存。对于singleton的循环依赖，Spring先将Bean实例化出来，不注入，缓存之，然后挨个注入。