D:\SpringJar(Part)CommonLogging\Spring430\spring-framework-reference\html\xsd-configuration.html

前置文件：建立spring工程，选自己导包idea

beans.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">

</beans>

一、简易入门：

0. 创建Bean类,提供私有变量的get/setter方法

1. 配置<bean id class><property>

<bean id="UserService" class="cn.cxd.beans.UserServiceImpl">

<property name="username" value="李白"></property>

<property name="password" value="123546"></property>

</bean>

2. 创建ApplicationContext的实现类ClassPathXmlApplicationContext(configLocation)

3. 使用ApplicationContext.getBean(id),强转类型为我们自己的类型

4. 完成对象的创建并将property标签的内容执行setter进入对象中

二、bean的作用

<bean> 的作用: create or find object in project

（scope 分为单例singleton，多例prototype）

三、对于bean标签的注入实现原理

1.解析XML文档，获取到id,class,property

2.实例化对象

3.为对象赋值value

四、使用上配置的xml信息

为工程配置xml:

// 1. ClassPathXmlApplicationContext 完成配置xml

ApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");

IUserService userService1 = (IUserService) context.getBean("UserService");

userService1.add();

// 2. FileSystemXmlApplicationContext 完成配置xml

ApplicationContext context1 = new FileSystemXmlApplicationContext("D:\\IDEAProjects\\day01-spring\\src\\beans.xml");

IUserService userService1 = (IUserService) context1.getBean("UserService");

userService1.add();

五、bean的三种装配和scope单多例的选择：

<?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">

<!-- bean的三种装配 -->

<!-- 第一种bean -->

<bean id="UserService" class="cn.cxd.beans.UserServiceImpl">

<property name="username" value="李白"></property>

<property name="password" value="123546"></property>

</bean>

<!-- 第二种bean(静态工厂create依靠factory-method) -->

<bean id="UserService2" class="cn.cxd.factorys.UserServiceStaticFactory" factory-method="createUserService">

<property name="username" value="杜甫"></property>

<property name="password" value="123546"></property>

</bean>

<!-- 第三种bean(工厂实例化create依靠factoryBean,factoryMethod)-->

<bean id="UserServiceInstanceFactory" class="cn.cxd.factorys.UserServiceInstanceFactory"></bean>

<bean id="UserService3" factory-bean="UserServiceInstanceFactory" factory-method="createUserService">

<property name="username" value="杜甫"></property>

<property name="password" value="123546"></property>

</bean>

<!-- bean的常见scope 单例singleton 多例prototype-->

<bean id="UserService4" class="cn.cxd.beans.UserServiceImpl" scope="singleton">

<property name="username" value="李清照"></property>

<property name="password" value="123546"></property>

</bean>

<bean id="UserService5" class="cn.cxd.beans.UserServiceImpl" scope="prototype">

<property name="username" value="辛弃疾"></property>

<property name="password" value="123546"></property>

</bean>

</beans>

六、bean的生命周期：

"D:\intellijIDEA1930\IntelliJ IDEA 2019.3.1\jbr\bin\java.exe" "-javaagent:D:\intellijIDEA1930\IntelliJ IDEA 2019.3.1\lib\idea\_rt.jar=50119:D:\intellijIDEA1930\IntelliJ IDEA 2019.3.1\bin" -Dfile.encoding=UTF-8 -classpath D:\IDEAProjects\testJava\out\production\day01-spring;D:\SpringJar(Part)CommonLogging\Spring430\Core-jars\commons-logging-1.2.jar;D:\SpringJar(Part)CommonLogging\Spring430\Core-jars\spring-core-4.3.0.RELEASE.jar;D:\SpringJar(Part)CommonLogging\Spring430\Core-jars\spring-beans-4.3.0.RELEASE.jar;D:\SpringJar(Part)CommonLogging\Spring430\Core-jars\spring-context-4.3.0.RELEASE.jar;D:\SpringJar(Part)CommonLogging\Spring430\Core-jars\spring-expression-4.3.0.RELEASE.jar test.UserServiceTest

2月 01, 2020 10:11:07 上午 org.springframework.context.support.ClassPathXmlApplicationContext prepareRefresh

信息: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@3d012ddd: startup date [Sat Feb 01 10:11:07 CST 2020]; root of context hierarchy

2月 01, 2020 10:11:07 上午 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions

信息: Loading XML bean definitions from class path resource [beans2.xml]

1:BeanFactoryPostProcessor(postProcessBeanFactory) configurableListableBeanFactory : org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0]; root of factory hierarchy

2.InstantiationAwareBeanPostProcessor(postProcessBeforeInstantiation):aClass:class cn.cxd.beans.UserServiceImpl,s:UserService8

3.Bean类的实例化：constructor here,username:null,password:null

4.InstantiationAwareBeanPostProcessor(postProcessAfterInstantiation):object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

5.InstantiationAwareBeanPostProcessor(postProcessPropertyValues):propertyValues:PropertyValues: length=2; bean property 'username'; bean property 'password',object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

propertyDescriptors:org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=class]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=password]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=username]

6.BeanNameAware(setBeanName):s:UserService8

7.BeanFactoryAware(setBeanFactory):beanFactory:org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0]; root of factory hierarchy

7-1.BeanPostProcessor(postProcessBeforeInitialization)object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

8.InstantiationAwareBeanPostProcessor(postProcessBeforeInitialization)object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

9.InitializingBean(afterPropertiesSet)空参void

10.beanInit(beans.xml.init\_method)空参void

10-1.BeanPostProcessor(postProcessAfterInitialization):object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

11.InstantiationAwareBeanPostProcessor(postProcessAfterInitialization):object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

12.bean的使用添加用户: 苏轼 123546

13.DisposableBean(destroy)空参void

14.beanDestroy(beans.xml.destroy\_method)void

end

2月 01, 2020 10:11:08 上午 org.springframework.context.support.ClassPathXmlApplicationContext doClose

信息: Closing org.springframework.context.support.ClassPathXmlApplicationContext@3d012ddd: startup date [Sat Feb 01 10:11:07 CST 2020]; root of context hierarchy

Process finished with exit code 0

bean 生命周期详解：

---1.如果有类实现了BeanFactoryPostProcessor接口，

需要重写postProcessBeanFactory(ConfigurableListableBeanFactory configurableListableBeanFactory)

其中：configurableListableBeanFactory :

org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy

1:BeanFactoryPostProcessor(postProcessBeanFactory) configurableListableBeanFactory : org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy

实现了BeanFactoryPostProcessor接口的类：root of factory hierarchy工程层次结构的根

---2.如果有类实现了InstantiationAwareBeanPostProcessor接口，

重写 postProcessBeforeInstantiation(Class<?> aClass, String s)

其中：aClass:class cn.cxd.beans.UserServiceImpl,s:UserService8

2.InstantiationAwareBeanPostProcessor(postProcessBeforeInstantiation):aClass:class cn.cxd.beans.UserServiceImpl,s:UserService8

---3.Bean类的实例化，调用Bean类的构造函数

3.Bean类的实例化：constructor here,username:null,password:null

---4.如果有类实现了InstantiationAwareBeanPostProcessor接口，

重写 postProcessAfterInstantiation(Object o, String s)

其中：object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

完成实例化，获取到bean对象

4.InstantiationAwareBeanPostProcessor(postProcessAfterInstantiation):object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

---5.如果有类实现了InstantiationAwareBeanPostProcessor接口，

重写postProcessPropertyValues(PropertyValues propertyValues, PropertyDescriptor[] propertyDescriptors, Object o, String s)

其中：

propertyValues:PropertyValues: length=2; bean property 'username'; bean property 'password'

object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

propertyDescriptors:org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=class]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=password]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=username]

id和属性property注入，初始化前

5.InstantiationAwareBeanPostProcessor(postProcessPropertyValues):propertyValues:PropertyValues: length=2; bean property 'username'; bean property 'password',object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

propertyDescriptors:org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=class]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=password]org.springframework.beans.GenericTypeAwarePropertyDescriptor[name=username]

获取到<bean>中的<property>中的id

---6.如果Bean类实现了BeanNameAware接口，

重写setBeanName(String s)

其中：s:UserService8

设置bean对象为<bean class id>中的id

6.BeanNameAware(setBeanName):s:UserService8

---7.如果Bean类实现了BeanFactoryAware接口，

重写setBeanFactory(BeanFactory beanFactory)

其中：beanFactory:

org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy

设置beanFactory 为实现了BeanFactoryPostProcessor接口的类

7.BeanFactoryAware(setBeanFactory):beanFactory:org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy

---7-1.如果有类实现了BeanPostProcessor接口，

重写postProcessBeforeInitialization(Object o, String s)

其中：object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

7-1.BeanPostProcessor(postProcessBeforeInitialization)object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

---8.如果有类实现了InstantiationAwareBeanPostProcessor接口，

重写postProcessBeforeInitialization(Object o, String s)

其中：object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

8.InstantiationAwareBeanPostProcessor(postProcessBeforeInitialization)object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

---9.如果Bean类实现了InitializingBean接口，

重写afterPropertiesSet()空参void的设定

在属性property注入后

9.InitializingBean(afterPropertiesSet)空参void

---10.如果在<beans.xml的bean中>配置了init-method,就会调用

这次的命名为beanInit()，也是空参void的设定

10.beanInit(beans.xml.init\_method)空参void

---10-1.如果有类实现了BeanPostProcessor接口，

重写postProcessAfterInitialization(Object o, String s)

其中：object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

10-1.BeanPostProcessor(postProcessAfterInitialization):object:cn.cxd.beans.UserServiceImpl@7f552bd3,s:UserService8

---11.如果有类实现了InstantiationAwareBeanPostProcessor接口，

重写postProcessAfterInitialization(Object o, String s),

其中object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

初始化后,property设置完后

11.InstantiationAwareBeanPostProcessor(postProcessAfterInitialization):object:cn.cxd.beans.UserServiceImpl@3eb7fc54,s:UserService8

---12.在getBean()后的使用,在main中获取到对象并使用方法

12.bean的使用添加用户: 苏轼 123546

---13.在main中将context强制关闭后context.getClass().getMethod("close").invoke(context);

如果Bean类实现了DisposableBean接口，需要重写destroy空参void方法

13.DisposableBean(destroy)空参void

---14.如果在<beans.xml的bean中>配置了destroy-method,就会调用

这次的命名为beanDestroy()，也是空参void的设定

14.beanDestroy(beans.xml.destroy\_method)void

end

beanFactory：org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy

configurableListableBeanFactory : org.springframework.beans.factory.support.DefaultListableBeanFactory@48524010: defining beans [UserService8,cn.cxd.beanLives.MyBeanFactoryPostProcessor#0,cn.cxd.beanLives.MyInstantiationAwareBeanPostProcessor#0]; root of factory hierarchy