

Spring03、HelloSpring

 狂神说 - SUIP 分类: 学习笔记 创建时间: 2021/04/13 10:48 ☒ 字体 ☐ 皮肤

最后修改于: 2021/04/13 21:56

3、HelloSpring

3.1、导入Jar包

注: spring 需要导入commons-logging进行日志记录. 我们利用maven, 他会自动下载对应的依赖项.

```
1. <dependency>
2.     <groupId>org.springframework</groupId>
3.     <artifactId>spring-webmvc</artifactId>
4.     <version>5.1.10.RELEASE</version>
5. </dependency>
```



3.2、编写代码

编写一个Hello实体类

```
1. public class Hello {
2.     private String name;
3.
4.     public String getName() {
5.         return name;
6.     }
7.     public void setName(String name) {
8.         this.name = name;
9.     }
10.
11.     public void show(){
12.         System.out.println("Hello,"+ name );
13.     }
14. }
```



编写我们的spring文件, 这里我们命名为beans.xml

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.     xsi:schemaLocation="http://www.springframework.org/schema/beans
5.         http://www.springframework.org/schema/beans/spring-beans.xsd">
6.
7.     <!--bean就是java对象, 由Spring创建和管理-->
8.     <bean id="hello" class="com.kuang.pojo.Hello">
9.         <property name="name" value="Spring"/>
10.    </bean>
11.
12. </beans>
```



我们可以去进行测试了.

```

1. @Test
2. public void test(){
3.     //解析beans.xml文件，生成管理相应的Bean对象
4.     ApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");
5.     //getBean：参数即为spring配置文件中bean的id。
6.     Hello hello = (Hello) context.getBean("hello");
7.     hello.show();
8. }

```

3.3、思考

- Hello 对象是谁创建的？【hello 对象是由Spring创建的】
- Hello 对象的属性是怎么设置的？【hello 对象的属性是由Spring容器设置的】

这个过程就叫控制反转：

- 控制：谁来控制对象的创建，传统应用程序的对象是由程序本身控制创建的，使用Spring后，对象是由Spring来创建的
- 反转：程序本身不创建对象，而变成被动的接收对象。

依赖注入：就是利用set方法来进行注入的。

==IOC是一种编程思想，由主动的编程变成被动的接收==

可以通过newClassPathXmlApplicationContext去浏览一下底层源码。

3.4、修改案例一

我们在案例一中，新增一个Spring配置文件beans.xml

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.     xsi:schemaLocation="http://www.springframework.org/schema/beans
5.         http://www.springframework.org/schema/beans/spring-beans.xsd">
6.
7.     <bean id="MysqlImpl" class="com.kuang.dao.impl.UserDaoMysqlImpl"/>
8.     <bean id="OracleImpl" class="com.kuang.dao.impl.UserDaoOracleImpl"/>
9.
10.    <bean id="ServiceImpl" class="com.kuang.service.impl.UserServiceImpl">
11.        <!--注意：这里的name并不是属性，而是set方法后面的那部分，首字母小写-->
12.        <!--引用另外一个bean，不是用value 而是用 ref-->
13.        <property name="userDao" ref="OracleImpl"/>
14.    </bean>
15.
16. </beans>

```

测试！

```

1. @Test
2. public void test2(){
3.     ApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");
4.     UserServiceImpl serviceImpl = (ServiceImpl) context.getBean("ServiceImpl");
5.     serviceImpl.getUser();
6. }

```

OK，到了现在，我们彻底不用再程序中去改动了，要实现不同的操作，只需要在xml配置文件中进行修改，所谓的IoC,一句话搞定：对象由Spring 来创建，管理，装配！

[关于我们](#) | [加入我们](#) | [联系我们](#)

Copyright © 广东学相伴网络科技有限公司 [粤ICP备 - 2020109190号](#)