

实验一、Spring 环境搭建

(一) 实验目的

1. 通过本实验学会搭建 Spring 项目
2. 通过本实验掌握 Spring 的 IoC 功能

(二) 实验内容

1. 新建 Web 工程
2. 将 Spring 的核心 jar 包文件导入项目
3. 创建 applicationContext.xml 文件，引入 Spring 的约束
4. 分层实现 service 层和 dao 层 bean
5. 在 applicationContext.xml 配置每一层 bean
6. 访问 service 层的 bean 对象

(三) 实验要求

1. 实验过程要求每个同学根据实验指导书认真完成
2. 每个同学认真书写实验总结，不能有雷同

(四) 实验步骤

1. 在 eclipse 中新建一个 java web 项目
2. 将 spring 的 4 个核心包和 common-logging 包导入到项目中
3. 在 src 下新建 xml 文件 applicationContext.xml
4. 从 spring 的帮助文档中的 Configuration Metadata 中将以下代码拷贝到 applicationContext.xml 中

```
<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>
```

5. 在 src 下建立 cn.edu.abtu.dao 包
6. 在 dao 包下新建接口类 IMyFirstSpringDao，在接口中新建 sayHello() 接口方法
7. 在 src 下新建 cn.edu.abtu.dao.impl 包
8. 在 impl 包下新建 IMyFirstSpringDao 接口的实现类 MyFirstSpringDaoImpl 类，实现方法 sayHello()，在 sayHello 方法中打印一句话“Hello, Spring”
9. 在 src 下建立 cn.edu.abtu.service 包
10. 在 service 包下新建接口类 IMyFirstSpringService，在该接口中添加接口方法 sayHello()

11. 在 src 下建立 cn.edu.abtu.service.impl 包,
12. 在 service.impl 包下新建 IMyFirstSpringService 接口的实现类 MyFirstSpringServiceImpl, 并实现接口中的 sayHello() 方法。方法中暂时不写任何实现代码。
13. 在 MyFirstSpringServiceImpl 类中声明 dao 层 IMyFirstSpringDao 接口的引用:
private IMyFirstSpringDao myFirstSpringDao
14. 在 MyFirstSpringServiceImpl 类中生成 myFirstSpringDao 的 setter 和 getter 方法
15. 在 MyFirstSpringServiceImpl 类的 sayHello 方法中添加调用 dao 层的代码
myFirstSpringDao.sayHello()
16. 在 applicationContext.xml 中配置 dao 层的 bean 组件, 在 <beans></beans> 标签中添加如下代码
<bean id="" class="" />
自己将双引号中的内容填进去
17. 在 applicationContext.xml 中添加 service 层 bean 组件, 并将 service 层多依赖的 dao 层组件通过属性注入到 service bean 组件中
<bean id="" class="">
 <property name="" ref="">
</bean>
自己将双引号中的内容填进去
18. 在 src 下建立 cn.edu.abtu.test 包
19. 在测试包下建立测试类, 在 main 方法中添加测试代码: 首先实例化 ApplicationContext 容器, 实例化 ApplicationContext 容器通过 ClassPathXmlApplicationContext 进行实例化, 传入的参数为 applicationContext.xml。再次通过 ApplicationContext 的 getBean 方法获取 service 层组件 bean, 参数为 applicationContext.xml 中 service 层 bean 的 id。最后调用 service 接口类的 sayHello() 方法。查看控制台的输出结果。

(五) 实验总结

通过观察输出结果和自己实验整个过程总结一下自己的学习所得