**每日作业卷答案**

**Spring第2天**

极客营

# 关卡1

## 训练案例

### 训练描述

现在提供了UserServiceImpl和UserDaoImpl实现类，要求使用Spring框架IOC（注解方式）的依赖注入（注解方式）的方式来管理UserServiceImpl和UserDaoImpl，并且把UserDaoImpl对象注入到UserServiceImpl类中。在demo1测试类中编写测试方法。获取到UserServiceImpl对象，完成方法调用。

### 操作步骤答案

1. 请在applicationContext.xml配置文件中开启注解扫描

<context:component-scan base-package="com.igeek"/>

1. 请在UserDaoImpl类添加注解  
   @Repository(value="userDao")
2. 把UserServiceImpl注添加注解  
   @Service(value="userService")
3. 在UserServiceImpl类中的userDao属性上添加注解，完成依赖注入  
   @Resource(name="userDao")

private UserDao userDao;

1. 在demo1类中完成测试方法，获取到UserServiceImpl对象，调用save方法，在控制台输出结果  
    @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

UserService us = (UserService) ac.getBean("userService");

us.save();

}

# 关卡2

## 训练案例

### 训练描述

现在提供了UserServiceImpl和UserDaoImpl实现类，要求使用Spring框架IOC（注解方式）的依赖注入（注解方式）的方式来管理UserServiceImpl和UserDaoImpl，注意：要求使用纯注解的方式来完成。不能xml的配置文件。并且把UserDaoImpl对象注入到UserServiceImpl类中。在demo1测试类中编写测试方法。获取到UserServiceImpl对象，完成方法调用。

### 操作步骤答案

1. 创建SpringConfiguration类

public class SpringConfiguration {

}

1. 在SpringConfiguration类添加注解，声明是一个配置类  
   @Configuration
2. 在SpringConfiguration类添加注解，配置要扫描的包结构

@ComponentScan(basePackages = "com.igeek")

1. 请在UserDaoImpl类添加注解  
   @Repository(value="userDao")
2. 把UserServiceImpl注添加注解  
   @Service(value="userService")
3. 在UserServiceImpl类中的userDao属性上添加注解，完成依赖注入  
   @Resource(name="userDao")

private UserDao userDao;

1. 在demo1类中完成测试方法，获取到UserServiceImpl对象，调用save方法，在控制台输出结果

@Test

public void run1(){

ApplicationContext ac = new AnnotationConfigApplicationContext(SpringConfiguration.class);

UserService us = (UserService) ac.getBean("userService");

us.save();

}

# 关卡3

## 训练案例1

### 训练描述

使用Spring框架IOC注解方式和依赖注入注解方式，给User类的2个属性赋值，使用ApplicationContext工厂获取到User类的对象，重写User类的toString方法，在控制台上打印User对象。

### 操作步骤描述

1. 请在applicationContext.xml配置文件开启注解扫描。  
   <context:component-scan base-package="com.igeek"/>
2. 在User类上添加注解，给name和age属性注入值  
   @Component(value="user")

public class User {

@Value(value="赵四")

private String name;

@Value(value="20")

private Integer age;

1. 请在Demo1类中的run1()测试方法中编写从IOC容器中获取User对象的方法，在控制台打印结果。  
   @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

User u = (User) ac.getBean("user");

System.out.println(u);

}

# 关卡4

## 训练案例1

### 训练描述

现在有UserServiceImpl实现类和UserDaoImpl实现类，要求把这个类都使用Spring的IOC进行管理（要求使用注解的方式），并且需要把UserDaoImpl注入到UserServiceImpl属性中（使用注解的方式），在测试方法中获取到UserServiceImpl实现类对象，完成方法的调用（要求使用Spring整合Junit单元测试的方式完成）。

### 操作步骤描述

1. 请在applicationContext.xml配置文件中开启注解扫描。

<!-- 开启注解扫描 -->

<context:component-scan base-package="com.igeek"/>

1. 使用IOC注解的方式管理UserServiceImpl  
     
   @Service(value="userService")

public class UserServiceImpl implements UserService

1. 使用IOC注解的方式管理UserDaoImpl

@Repository(value="userDao")

public class UserDaoImpl implements UserDao

1. 使用IOC注解的方式在UserServiceImpl中注入UserDaoImpl的对象  
     
   @Resource(name="userDao")

private UserDao userDao;

1. 使用Spring整合Junit单元测试的方式，获取到UserServiceImpl对象，调用该对象的save方法，结果打印在控制台上  
   @RunWith(value=SpringJUnit4ClassRunner.class)

@ContextConfiguration(value="classpath:applicationContext.xml")

public class Demo1 {

@Resource(name="userService")

private UserService userService;

@Test

public void run1(){

userService.save();

}

}

# 关卡5

## 训练案例

### 训练描述

使用Spring和Hibernate框架，把User对象的数据存入到数据库中。提供了hibernate.cfg.xml配置文件，提供了HibernateUtils工具类，提供applicationContext.xml配置文件，提供了User类，提供了UserServiceImpl和UserDaoImpl两个实现类（要求使用IOC注解的方式）

### 操作步骤描述

1. 先在MySQL数据库中创建名称为spring\_day01的数据库  
   create database spring\_day01;
2. 根据User类编写User.hbm.xml的映射配置文件  
   <?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name="com.igeek.domain.User" table="t\_user">

<id name="id" column="id">

<generator class="native"/>

</id>

<property name="name" column="name"/>

<property name="age" column="age"/>

</class>

</hibernate-mapping>

1. 在hibernate.cfg.xml配置文件引入User.hbm.xml的映射配置文件  
   <mapping resource="com/igeek/domain/User.hbm.xml"/>
2. 在UserDaoImpl类的save方法编写代码，把User对象存入到数据库中  
   Session session = HibernateUtils.getCurrentSession();

session.save(user);

1. 在UserServiceImpl类的save方法编写代码，开启事务，调用dao保存User对象  
   public void save(User user) {

Session session = HibernateUtils.getCurrentSession();

Transaction tx = session.beginTransaction();

try {

userDao.save(user);

tx.commit();

} catch (Exception e) {

e.printStackTrace();

tx.rollback();

}

}

1. 在applicationContext.xml配置文件中，开启注解扫描  
   <context:component-scan base-package="com.igeek"/>
2. 使用注解方式管理UserServiceImpl类  
   @Service(value="userService")

public class UserServiceImpl implements UserService

1. 使用注解方式管理UserDaoImpl类  
   @Repository(value="userDao")

public class UserDaoImpl implements UserDao

1. 把dao对象通过依赖注入注解方式注入到service的属性中  
   @Resource(name="userDao")

private UserDao userDao;

1. 使用注解方式管理User类，并且给User类的name和age属性注入值（值任意）  
   @Component(value="user")

public class User {

private Integer id;

@Value(value="赵四")

private String name;

@Value(value="20")

private Integer age;

1. 在Demo1类的测试方法run1方法中，从IOC容器中获取到User对象和Service的对象，调用保存的方法，把User对象数据保存到数据库中（要求使用Spring整合Junit单元测试）  
   @RunWith(value=SpringJUnit4ClassRunner.class)

@ContextConfiguration(value="classpath:applicationContext.xml")

public class Demo1 {

@Resource(name="userService")

private UserService userService;

@Resource(name="user")

private User user;

/\*\*

\* 测试保存的方法

\*/

@Test

public void run1(){

userService.save(user);

}

}