# 第8章 Spring的数据库编程

## 实验目的

通过基于annotation的装配方式装配bean实例，来达到熟练掌握Spring中bean的装配方式的目的。

## 实验内容

本实验要求学生在控制台输入用户名密码进行注册和登录，先注册，后登录。登录时如果用户账号密码正确，则显示用户所属班级，如果登录失败则显示登录失败。

实现用户登录项目运行成功后控制台效果如下所示。

## 实验代码：

### jdbc.properties

jdbc.driver=com.mysql.jdbc.Driver  
jdbc.url=jdbc:mysql://192.168.120.20:3306/spring?useSSL=false  
jdbc.username=root  
jdbc.password=z!cGnFq9YSvS$yopjf&Vr

### JdbcConfig.java

package com.shf.sy8.config;  
  
import com.alibaba.druid.pool.DruidDataSource;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.context.annotation.Bean;  
  
import javax.sql.DataSource;  
  
public class JdbcConfig {  
 @Value("${jdbc.driver}")  
 private String driver;  
 @Value("${jdbc.url}")  
 private String url;  
 @Value("${jdbc.username}")  
 private String userName;  
 @Value("${jdbc.password}")  
 private String password;  
  
 @Bean  
 public DataSource dataSource(){  
 DruidDataSource ds = new DruidDataSource();  
 ds.setDriverClassName(driver);  
 ds.setUrl(url);  
 ds.setUsername(userName);  
 ds.setPassword(password);  
 return ds;  
 }  
}

### MybatisConfig.java

package com.shf.sy8.config;  
  
import org.mybatis.spring.SqlSessionFactoryBean;  
import org.mybatis.spring.mapper.MapperScannerConfigurer;  
import org.springframework.context.annotation.Bean;  
  
import javax.sql.DataSource;  
  
public class MybatisConfig {  
 @Bean  
 public SqlSessionFactoryBean sqlSessionFactory(DataSource dataSource){  
 SqlSessionFactoryBean ssfb = new SqlSessionFactoryBean();  
 ssfb.setTypeAliasesPackage("com.shf.sy8.pojo");  
 ssfb.setDataSource(dataSource);  
 return ssfb;  
 }  
  
 @Bean  
 public MapperScannerConfigurer mapperScannerConfigurer(){  
 MapperScannerConfigurer msc = new MapperScannerConfigurer();  
 msc.setBasePackage("com.shf.sy8.dao");  
 return msc;  
 }  
}

### SpringConfig.java

package com.shf.sy8.config;  
  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.Import;  
import org.springframework.context.annotation.PropertySource;  
  
@Configuration  
@ComponentScan("com.shf.sy8")  
@PropertySource({"classpath:jdbc.properties"})  
@Import({JdbcConfig.class,MybatisConfig.class})  
public class SpringConfig {  
}

### Student.java

package com.shf.sy8.pojo;  
  
import lombok.Data;  
  
import java.io.Serializable;  
  
@Data  
public class Student implements Serializable {  
 private Integer id;  
 private String username;  
 private String password;  
 private String course;  
  
 public Student(String username, String password, String course) {  
 this.username = username;  
 this.password = password;  
 this.course = course;  
 }  
}

### StudentDao.java

package com.shf.sy8.dao;  
  
import com.shf.sy8.pojo.Student;  
import org.apache.ibatis.annotations.Insert;  
import org.apache.ibatis.annotations.Param;  
import org.apache.ibatis.annotations.Select;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public interface StudentDao {  
  
 @Select("select *\** from student")  
 public List<Student> getAllStudent();  
  
 @Insert("insert into student (username,password,course) values (#{username},#{password},#{course})")  
 public int addStudent(Student student);  
}

### StudentService.java

package com.shf.sy8.service;  
  
import com.shf.sy8.pojo.Student;  
import org.apache.ibatis.annotations.Param;  
  
import java.util.List;  
  
public interface StudentService {  
 public List<Student> getAllStudent();  
  
 public int addStudent(Student student);  
}

### StudentServiceImpl.java

package com.shf.sy8.service.impl;  
  
import com.shf.sy8.dao.StudentDao;  
import com.shf.sy8.pojo.Student;  
import com.shf.sy8.service.StudentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class StudentServiceImpl implements StudentService {  
 @Autowired  
 private StudentDao studentDao;  
  
 public List<Student> getAllStudent(){  
 return studentDao.getAllStudent();  
 }  
  
 @Override  
 public int addStudent(Student student) {  
 return studentDao.addStudent(student);  
 }  
}

### LoginTest.java

package com.shf.sy8.test;  
  
import com.shf.sy8.config.SpringConfig;  
import com.shf.sy8.pojo.Student;  
import com.shf.sy8.service.StudentService;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.test.context.ContextConfiguration;  
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;  
  
import java.util.List;  
import java.util.Scanner;  
  
@RunWith(SpringJUnit4ClassRunner.class)  
@ContextConfiguration(classes = SpringConfig.class)  
public class LoginTest {  
 @Autowired  
 private StudentService studentService;  
  
 List<Student> allStudent = null;  
  
 private boolean flag = true;  
  
 @Test  
 public void test1() {  
 Scanner scanner = new Scanner(System.*in*);  
 while (flag){  
 System.*out*.println("=============欢迎来到学生管理系统===============");  
 System.*out*.println("请选择登录或注册：1-登录 2-注册 3-退出");  
 String type = scanner.nextLine();  
 if (type.equals("1")){  
 boolean isFInd = false;  
 allStudent = studentService.getAllStudent();  
 System.*out*.println("=============登录系统=============");  
 System.*out*.println("请输入用户名：");  
 String username = scanner.nextLine();  
 System.*out*.println("请输入 "+username+" 的密码：");  
 String password = scanner.nextLine();  
  
 for (Student student : allStudent) {  
 if (student.getUsername().equalsIgnoreCase(username)){  
 if (student.getPassword().equalsIgnoreCase(password)){  
 isFInd=true;  
 System.*out*.println("用户登录成功");  
 System.*out*.println(student.getUsername()+"是"+student.getCourse());  
 }  
 }  
 }  
  
 if (isFInd==false){  
 System.*out*.println("用户不存在或者用户名密码错误");  
 }  
 } else if (type.equals("2")){  
 System.*out*.println("===================注册用户===================");  
 System.*out*.println("请设置用户名：");  
 String username = scanner.nextLine();  
 System.*out*.println("请设置 "+username+" 的密码：");  
 String password = scanner.nextLine();  
 System.*out*.println("请设置 "+username+" 的班级：");  
 String course = scanner.nextLine();  
 if (username!=null && password!=null && course!=null){  
 Student student = new Student(username, password, course);  
 int res = studentService.addStudent(student);  
 if (res>=1){  
 System.*out*.println("学生添加成功，请登录");  
 } else {  
 System.*out*.println("学生添加失败，请重新注册");  
 }  
 }  
 } else if (type.equals("3")){  
 flag=false;  
 }  
 }  
 }  
}

## 实验截图：





## 实验小结

通过本次实验我学会通过注解的方式将spring和mybatis进行整合，通过springJDBC可以管理数据库连接资源，简化传统的JDBC的操作，进而提升数据库操作的效率，以及spring中对于事务的管理，要么同时成功要么同时失败。