第二章习题答案

1.填空题

（1）<property>子元素、properties文件

（2）别名

（3）事务管理器的类型

（4）<select>

（5）<sql>

2.选择题

（1）B

（2）B

（3）B

（4）D

（5）D

3.思考题

（1）

配置文件对MyBatis的整个运行体系产生影响，它包含了很多控制MyBatis功能的重要信息，是MyBatis实现功能的重要保证。

（2）

映射文件是MyBatis中的又一重要组件，它包含了各类SQL语句、参数、结果集、映射规则等信息。

4.编程题

Ⅰ.Student.java实体类

1. package com.qfedu.pojo;
2. public class Student {
3. private Integer sid;
4. private String sname;
5. private String age;
6. private String course;
7. public Student() {
8. super();
9. // TODO Auto-generated constructor stub
10. }
11. public Student(Integer sid, String sname, String age, String course) {
12. super();
13. this.sid = sid;
14. this.sname = sname;
15. this.age = age;
16. this.course = course;
17. }
18. public Integer getSid() {
19. return sid;
20. }
21. public void setSid(Integer sid) {
22. this.sid = sid;
23. }
24. public String getSname() {
25. return sname;
26. }
27. public void setSname(String sname) {
28. this.sname = sname;
29. }
30. public String getAge() {
31. return age;
32. }
33. public void setAge(String age) {
34. this.age = age;
35. }
36. public String getCourse() {
37. return course;
38. }
39. public void setCourse(String course) {
40. this.course = course;
41. }
42. @Override
43. public String toString() {
44. return "Student [sid=" + sid + ", sname=" + sname + ", age=" + age + ", course=" + course + "]";
45. }
46. }

Ⅱ.StudentDao.java

1. package com.qfedu.dao;
2. import java.util.List;
3. import com.qfedu.pojo.Student;
4. public interface StudentDao {
5. public List<Student> findAllStudent();
6. }

Ⅲ.StudentMapper.xml

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "../mybatis-3-mapper.dtd" >
3. <mapper namespace="com.qfedu.dao.StudentDao">
4. <resultMap id="studentMap" type="com.qfedu.pojo.Student">
5. <id column="ID" jdbcType="INTEGER" property="id" />
6. <result column="sname" jdbcType="VARCHAR" property="sname" />
7. <result column="age" jdbcType="VARCHAR" property="age" />
8. <result column="course" jdbcType="VARCHAR" property="course" />
9. </resultMap>
10. <select id="findAllStudent" resultMap="studentMap">
11. select \* from student
12. </select>
13. </mapper>

Ⅳ.StudentService.java

1. package com.qfedu.service;
2. import java.util.List;
3. import com.qfedu.pojo.Student;
4. public interface StudentService {
5. public List<Student> findAllStudent();
6. }

Ⅴ.StudentServiceImp.java

1. package com.qfedu.service.impl;
2. import java.util.List;
3. import javax.annotation.Resource;
4. import org.springframework.beans.factory.annotation.Autowired;
5. import org.springframework.stereotype.Service;
6. import com.qfedu.dao.StudentDao;
7. import com.qfedu.pojo.Student;
8. import com.qfedu.service.StudentService;
9. @Service
10. public class StudentServiceImpl implements StudentService {
11. @Autowired(required=true)
12. private StudentDao studentDao;
13. @Override
14. public List<Student> findAllStudent() {
15. return studentDao.findAllStudent();
16. }
17. public StudentDao getStudentDao() {
18. return studentDao;
19. }
20. public void setStudentDao(StudentDao studentDao) {
21. this.studentDao = studentDao;
22. }
23. }

Ⅵ.application.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. xmlns:context="http://www.springframework.org/schema/context"
5. xmlns:aop="http://www.springframework.org/schema/aop"
6. xmlns:tx="http://www.springframework.org/schema/tx"
7. xsi:schemaLocation="http://www.springframework.org/schema/beans
8. http://www.springframework.org/schema/beans/spring-beans.xsd
9. http://www.springframework.org/schema/context
10. http://www.springframework.org/schema/context/spring-context.xsd
11. http://www.springframework.org/schema/aop
12. http://www.springframework.org/schema/aop/spring-aop.xsd
13. http://www.springframework.org/schema/tx
14. http://www.springframework.org/schema/tx/spring-tx.xsd">
15. <context:component-scan base-package="com.qfedu.\*" />
16. <!-- 引入外部 properties 文件 -->
17. <context:property-placeholder location="classpath:jdbc.properties"/>
18. <!-- 注册数据源 -->
19. <bean name="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
20. <property name="driverClass" value="${jdbc.driverClass}"/>
21. <property name="jdbcUrl" value="${jdbc.jdbcUrl}"/>
22. <property name="user" value="${jdbc.user}"/>
23. <property name="password" value="${jdbc.password}"/>
24. </bean>
25. <!--注册JdbcTemplate类 -->
26. <bean name="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
27. <property name="dataSource" ref="dataSource"/>
28. </bean>
29. </beans>

Ⅶ.test测试类

1. package com.qfedu.test;
2. import java.util.List;
3. import org.junit.Test;
4. import org.junit.runner.RunWith;
5. import org.springframework.beans.factory.annotation.Autowired;
6. import org.springframework.context.ApplicationContext;
7. import org.springframework.context.support.ClassPathXmlApplicationContext;
8. import org.springframework.test.context.ContextConfiguration;
9. import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
10. import com.qfedu.pojo.Student;
11. import com.qfedu.service.StudentService;
12. @RunWith(SpringJUnit4ClassRunner.class)
13. @ContextConfiguration(locations = { "classpath:application.xml" })
14. public class TestUtil {
15. @Autowired(required=true)
16. private StudentService studentService;
17. @Test
18. public void test1() {
19. List<Student> list = studentService.findAllStudent();
20. for(Student stu:list) {
21. System.out.println(stu.toString());
22. System.out.println("================");
23. }
24. }
25. public static void main(String[] args) {
26. ApplicationContext ctx=new ClassPathXmlApplicationContext("application.xml");
27. StudentService ss = (StudentService) ctx.getBean("studentService");
28. List<Student> list = ss.findAllStudent();
29. System.out.println("================");
30. }
31. }