Spring + Ibatis + MySql实例详解

1. 环境：  
将以下jar包加入到工程，commons-logging-1.0.4.jar、ibatis-2.3.0.677.jar、mysql-connector-java-5.0.3-bin.jar、spring.jar。

2. 在MySql中创建数据库和相应的表：

**[sql]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. #############################################################################################
2. **CREATE** **DATABASE** MYDB;
3. use MYDB;
5. **Drop** **TABLE** IF EXISTS `MYDB`.`student`;
6. **Create** **TABLE** `MYDB`.`student` (
7. `**name**` **varchar**(40) NOT NULL,
8. `psw` **varchar**(10) NOT NULL,
9. `enabled` boolean
10. );
11. **insert** **into** student **values**("lanp","lanpiao",**true**);
12. **insert** **into** student **values**("ph","ph",**true**);
13. **insert** **into** student **values**("wxh","wxh",**true**);

3. 创建实体Bean，Student.java

**[java]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **package** com.lanp.beans;
3. **import** java.io.Serializable;
5. /\*\*
6. \* Student Bean
7. \* @author LanP
8. \* @since 2011-11-27 15:36
9. \* @version V1.0
10. \*/
11. **public** **class** Student **implements** Serializable {
13. **private** **static** **final** **long** serialVersionUID = -7163004163334815825L;
15. **private** String name;
16. **private** String psw;
17. **private** Boolean enabled;
19. **public** String getName() {
20. **return** name;
21. }
22. **public** **void** setName(String name) {
23. **this**.name = name;
24. }
25. **public** String getPsw() {
26. **return** psw;
27. }
28. **public** **void** setPsw(String psw) {
29. **this**.psw = psw;
30. }
31. **public** Boolean getEnabled() {
32. **return** enabled;
33. }
34. **public** **void** setEnabled(Boolean enabled) {
35. **this**.enabled = enabled;
36. }
37. }

4. 创建Student实体Bean与数据库映射的SQLMap文件，student.xml：

**[html]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. <!DOCTYPE sqlMap
3. PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"
4. "http://ibatis.apache.org/dtd/sql-map-2.dtd"**>**
5. **<sqlMap>**
6. <!-- 为Person类设置一个别名 -->
7. **<typeAlias** alias="student" type="com.lanp.beans.Student"**/>**
9. <!-- 配置表和实体Bean之间的映射关系 -->
10. **<resultMap** id="studentMap" class="com.lanp.beans.Student"**>**
11. **<result** property="name" column="name"**/>**
12. **<result** property="psw" column="psw"**/>**
13. **<result** property="enabled" column="enabled"**/>**
14. **</resultMap>**
16. **<insert** id="insertStudent" parameterClass="student"**>**
17. <![CDATA[
18. insert into student values(#name#,#psw#,#enabled#);
19. ]]>
20. **</insert>**
22. <!-- 查看特定用户 -->
23. **<select** id="queryStudentById" parameterClass="string" resultMap="studentMap"**>**
24. <![CDATA[
25. SELECT \* FROM STUDENT WHERE NAME=#name#
26. ]]>
27. **</select>**
29. <!-- 查看所有的用户 -->
30. **<select** id="queryAllStudents" resultMap="studentMap"**>**
31. <![CDATA[
32. SELECT \* FROM STUDENT
33. ]]>
34. **</select>**
35. **</sqlMap>**

5. 创建访问数据库的DAO接口，StudentDao.java：

**[java]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **package** com.lanp.dao;
3. **import** com.lanp.beans.Student;
5. **public** **interface** StudentDao {
6. Student getStudent(String name);
7. }

6. 创建访问数据库的DAO接口实现类，StudentDaoImpl.java：

**[java]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **package** com.lanp.dao;
3. **import** org.springframework.orm.ibatis.support.SqlMapClientDaoSupport;
5. **import** com.lanp.beans.Student;
7. **public** **class** StudentDaoImpl **extends** SqlMapClientDaoSupport **implements** StudentDao {
9. @Override
10. **public** Student getStudent(String name) {
11. **try**{
12. **return** (Student)getSqlMapClientTemplate().queryForObject("queryStudentById", name);
13. } **catch**(Exception e) {
14. e.printStackTrace();
15. }
16. **return** **null**;
17. }
19. }

7. Ibatis总配置文件，sqlMapConfig.xml：

**[html]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. <!DOCTYPE sqlMapConfig
3. PUBLIC "-//ibatis.apache.org//DTD SQL Map Config 2.0//EN"
4. "http://ibatis.apache.org/dtd/sql-map-config-2.dtd"**>**
5. **<sqlMapConfig>**
6. <!-- 配置Ibatis要使用的SqlMap文件信息 -->
7. **<sqlMap** resource="com/lanp/beans/student.xml"**/>**
8. **</sqlMapConfig>**

8. spring配置文件，SQLMapClient.xml：

**[html]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd"**>**
3. **<beans>**
5. <!-- 相关数据源和事务管理的定义 -->
6. **<bean** id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource"**>**
7. **<property** name="driverClassName" value="com.mysql.jdbc.Driver"**/>**
8. **<property** name="url" value="jdbc:mysql://127.0.0.1:3306/MYDB"**/>**
9. **<property** name="username" value="root"**/>**
10. **<property** name="password" value="157891"**/>**
11. **</bean>**
13. **<bean** id="sqlMapClient" class="org.springframework.orm.ibatis.SqlMapClientFactoryBean"**>**
14. **<property** name="configLocation"**>**
15. **<value>**sqlMapConfig.xml**</value>**
16. **</property>**
17. **<property** name="dataSource"**>**
18. **<ref** bean="dataSource"**/>**
19. **</property>**
20. **</bean>**
22. **<bean** id="studentDao" class="com.lanp.dao.StudentDaoImpl"**>**
23. **<property** name="sqlMapClient"**>**
24. **<ref** bean="sqlMapClient"**/>**
25. **</property>**
26. **</bean>**
28. **</beans>**

9. 测试类，TestStudent：

**[java]** [view plaincopyprint?](http://blog.csdn.net/lanpiao_87/article/details/7036409)

1. **package** com.lanp.beans;
3. **import** org.springframework.context.ApplicationContext;
4. **import** org.springframework.context.support.ClassPathXmlApplicationContext;
6. **import** com.lanp.dao.StudentDao;
8. /\*\*
9. \* 测试Ibatis
10. \* @author LanP
11. \* @since 2011-11-27 15:36
12. \* @version V1.0
13. \*/
14. **public** **class** TestStudent {
16. **public** **static** **void** main(String[] args) {
18. //1.初始化beans.xml文件
19. ApplicationContext ctx = **new** ClassPathXmlApplicationContext("SQLMapClient.xml");
20. //2.获取MYDB数据库Student表中的内容
21. StudentDao studentDao = (StudentDao)ctx.getBean("studentDao");
22. **if**(**null** != studentDao) {
23. Student student = studentDao.getStudent("ph");
24. **if**(**null** != student) {
25. System.out.println("== 学生名字：" + student.getName() + ",学生密码：" + student.getPsw());
26. } **else** {
27. System.out.println("== 没有该学生信息！");
28. }
29. } **else** {
30. System.out.println("== StudentDao注入失败！");
31. }
33. }
35. }