1. 介绍

       Ibatis是开源的持久层框架。它的核心是SqlMap，将实体Bean跟关系数据库进行映射，将业务代码和SQL语句的书写进行分开，方便管理。Ibatis是“半自动”的ORM持久层框架。这里的“半自动化”，是相对Hibernate等提供了全面的数据库封装机制的“全自动化”ORM 实现而言，“全自动”ORM 实现了 POJO 和数据库表之间的映射，以及 SQL 的自动生成和执行。而iBATIS 的着力点，则在于POJO 与 SQL之间的映射关系。也就是说，iBATIS并不会为程序员在运行期自动生成 SQL 执行。具体的 SQL 需要程序员编写，然后通过映射配置文件，将SQL所需的参数，以及返回的结果字段映射到指定 POJO。

2. 前提

    1) 安装了[MySql](http://lib.csdn.net/base/14)数据库；

    2) 将以下Jar包加入工程的classpath：commons-logging-1.0.4.jar、ibatis-2.3.0.677.jar、mysql-connector-java-5.0.3-bin.jar。

3. 实例

 3.1 在MySql数据库中创建数据库

**[sql]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

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**);

 3.2 书写实体Bean：Student.java

**[java]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

1. **package** com.lanp.beans;
3. /\*\*
4. \* Student Bean
5. \* @author LanP
6. \* @since 2011-11-27 15:36
7. \* @version V1.0
8. \*/
9. **public** **class** Student {
10. **private** String name;
11. **private** String psw;
12. **private** Boolean enabled;
14. **public** String getName() {
15. **return** name;
16. }
17. **public** **void** setName(String name) {
18. **this**.name = name;
19. }
20. **public** String getPsw() {
21. **return** psw;
22. }
23. **public** **void** setPsw(String psw) {
24. **this**.psw = psw;
25. }
26. **public** Boolean getEnabled() {
27. **return** enabled;
28. }
29. **public** **void** setEnabled(Boolean enabled) {
30. **this**.enabled = enabled;
31. }
32. }

 3.3 配置数据库的属性文件：mysql.properties

**[html]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

1. mysql.driver = com.mysql.jdbc.Driver
2. mysql.url = jdbc:mysql://localhost:3306/MYDB
3. mysql.username = root
4. mysql.password = 157891

 3.4 SqlMap配置文件：sqlMapConfig.xml

**[html]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

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. <!-- Properties属性配置文件，加载数据库连接信息 -->
7. **<properties** resource="mysql.properties"**/>**
9. **<settings**
10. cacheModelsEnabled="true"
11. enhancementEnabled="true"
12. lazyLoadingEnabled="true"
13. errorTracingEnabled="true"
14. maxRequests="32"
15. maxSessions="10"
16. maxTransactions="5"
17. useStatementNamespaces="false"
18. **/>**
19. <!-- 配置Ibatis事务管理，使用JDBC事务类型，数据源使用Simple类型 -->
20. **<transactionManager** type="JDBC"**>**
21. **<dataSource** type="SIMPLE"**>**
22. **<property** name="JDBC.Driver" value="${mysql.driver}"**/>**
23. **<property** name="JDBC.ConnectionURL" value="${mysql.url}"**/>**
24. **<property** name="JDBC.Username" value="${mysql.username}"**/>**
25. **<property** name="JDBC.Password" value="${mysql.password}"**/>**
26. **</dataSource>**
27. **</transactionManager>**
28. <!-- 配置Ibatis要使用的SqlMap文件信息 -->
29. **<sqlMap** resource="com/lanp/beans/student.xml"**/>**
30. **</sqlMapConfig>**

 3.5 student实体Bean的映射配置：student.xml

**[html]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

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>**

 3.6 测试类：TestStudent.java

**[java]** [view plain](http://blog.csdn.net/lanpiao_87/article/details/7017404) [copy](http://blog.csdn.net/lanpiao_87/article/details/7017404)

[print](http://blog.csdn.net/lanpiao_87/article/details/7017404)[?](http://blog.csdn.net/lanpiao_87/article/details/7017404)

1. **package** com.lanp.beans;
3. **import** java.io.Reader;
4. **import** java.util.List;
6. **import** com.ibatis.common.resources.Resources;
7. **import** com.ibatis.sqlmap.client.SqlMapClient;
8. **import** com.ibatis.sqlmap.client.SqlMapClientBuilder;
10. /\*\*
11. \* 测试Ibatis
12. \* @author LanP
13. \* @since 2011-11-27 15:36
14. \* @version V1.0
15. \*/
16. **public** **class** TestStudent {
18. **public** **static** **void** main(String[] args) {
20. String resource = "sqlMapConfig.xml";
21. **try** {
22. //读取配置文件
23. Reader reader = Resources.getResourceAsReader(resource);
24. //得到SqlMapClient
25. SqlMapClient sqlMap = SqlMapClientBuilder.buildSqlMapClient(reader);
27. //新增学生信息
28. insertStudent(sqlMap);
30. //查看全部的学生
31. queryAllStudents(sqlMap);
33. //查看特定的学生
34. queryStudentByName(sqlMap);
35. } **catch**(Exception e){}
36. }
38. /\*\*
39. \* 根据学生的名字查询特定的学生信息
40. \* @param sqlMap
41. \*/
42. **private** **static** **void** queryStudentByName(SqlMapClient sqlMap) {
43. **try** {
44. System.out.println("--------------查询特定的学生信息-------------------------");
45. Student stu = (Student)sqlMap.queryForObject("queryStudentById", "lanp");
46. **if**(**null** != stu) {
47. System.out.println("== 学生名字： " + stu.getName() + " ,学生密码： " + stu.getPsw() + " ==");
48. }
49. } **catch**(Exception e){}
50. }
52. /\*\*
53. \* 显示所有学生的信息
54. \* @param sqlMap
55. \*/
56. @SuppressWarnings("unchecked")
57. **private** **static** **void** queryAllStudents(SqlMapClient sqlMap) {
58. **try** {
59. System.out.println("--------------查询所有的学生信息-------------------------");
60. List<Student> students = sqlMap.queryForList("queryAllStudents");
61. **if**(**null** != students && students.size()>0) {
62. **for**(**int** i=0; i<students.size(); i++) {
63. Student student = students.get(i);
64. System.out.println("== 学生名字： " + student.getName() + " ,学生密码： " + student.getPsw() +" ==");
65. }
66. }
67. } **catch**(Exception e){}
68. }


72. /\*\*
73. \* 新增学生信息
74. \* @param sqlMap
75. \*/
76. **private** **static** **void** insertStudent(SqlMapClient sqlMap) {
77. **try** {
78. System.out.println("--------------新增学生信息-------------------------");
79. Student student = **new** Student();
80. student.setName("xinh");
81. student.setPsw("123");
82. student.setEnabled(**true**);
83. //开始Ibatis事务
84. sqlMap.startTransaction();
85. sqlMap.insert("insertStudent", student);
86. //结束IBatis事务
87. sqlMap.commitTransaction();
88. } **catch**(Exception e){}
89. }
91. }