**参考：**[**http://www.cnblogs.com/xdp-gacl/p/4261895.html**](http://www.cnblogs.com/xdp-gacl/p/4261895.html)

**只为成功找方法，不为失败找借口！**

[**MyBatis学习总结(一)——MyBatis快速入门**](http://www.cnblogs.com/xdp-gacl/p/4261895.html)

**一、Mybatis介绍**

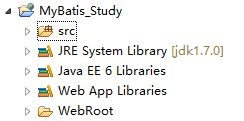


　　MyBatis是一个支持***普通SQL查询***，***存储过程***和***高级映射***的优秀**持久层框架**。MyBatis消除了几乎所有的JDBC代码和参数的手工设置以及对结果集的检索封装。MyBatis可以使用简单的**XML或注解**用于配置和原始映射，将接口和Java的PO映射成数据库中的记录。

**二、mybatis快速入门**

**2.1、准备开发环境**

 1、创建测试项目，普通java项目或者是JavaWeb项目均可，如下图所示：



2、添加相应的jar包

**【mybatis】**

       　　mybatis-3.1.1.jar

**【MYSQL驱动包】**  
　　　　mysql-connector-java-5.1.7-bin.jar

3、创建数据库和表，针对MySQL数据库

　　SQL脚本如下：

create database mybatis;

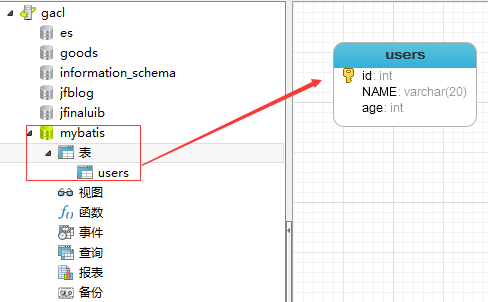
use mybatis;

CREATE TABLE users(id INT PRIMARY KEY AUTO\_INCREMENT, NAME VARCHAR(**20**), age INT);

INSERT INTO users(NAME, age) VALUES('孤傲苍狼', **27**);

INSERT INTO users(NAME, age) VALUES('白虎神皇', **27**);

　　将SQL脚本在MySQL数据库中执行，完成创建数据库和表的操作，如下：

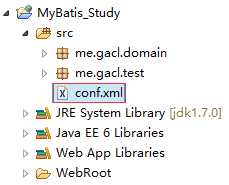


　　到此，前期的开发环境准备工作全部完成。

**2.2、使用MyBatis查询表中的数据**

**1、添加Mybatis的配置文件conf.xml**

　　在src目录下创建一个conf.xml文件，如下图所示：



　　conf.xml文件中的内容如下：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

3 <configuration>

4 <environments default="development">

5 <environment id="development">

6 <transactionManager type="JDBC" />

7 <!-- 配置数据库连接信息 -->

8 <dataSource type="POOLED">

9 <property name="driver" value="com.mysql.jdbc.Driver" />

10 <property name="url" value="jdbc:mysql://localhost:3306/mybatis" />

11 <property name="username" value="root" />

12 <property name="password" value="XDP" />

13 </dataSource>

14 </environment>

15 </environments>

16

17 </configuration>

[复制代码](javascript:void(0);)

**2、定义表所对应的实体类，如下图所示：**

　　User类的代码如下：

[复制代码](javascript:void(0);)

1 package me.gacl.domain;

2

3 /\*\*

4 \* @author gacl

5 \* users表所对应的实体类

6 \*/

7 public class User {

8

9 //实体类的属性和表的字段名称一一对应

10 private int id;

11 private String name;

12 private int age;

13

14 public int getId() {

15 return id;

16 }

17

18 public void setId(int id) {

19 this.id = id;

20 }

21

22 public String getName() {

23 return name;

24 }

25

26 public void setName(String name) {

27 this.name = name;

28 }

29

30 public int getAge() {

31 return age;

32 }

33

34 public void setAge(int age) {

35 this.age = age;

36 }

37

38 @Override

39 public String toString() {

40 return "User [id=" + id + ", name=" + name + ", age=" + age + "]";

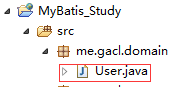
41 }

42 }

[复制代码](javascript:void(0);)

　　3、定义操作users表的sql映射文件userMapper.xml

　　创建一个me.gacl.mapping包，专门用于存放sql映射文件，在包中创建一个userMapper.xml文件，如下图所示：



userMapper.xml文件的内容如下：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8" ?>

2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

3 <!-- 为这个mapper指定一个唯一的namespace，namespace的值习惯上设置成包名+sql映射文件名，这样就能够保证namespace的值是唯一的

4 例如namespace="me.gacl.mapping.userMapper"就是me.gacl.mapping(包名)+userMapper(userMapper.xml文件去除后缀)

5 -->

6 <mapper namespace="me.gacl.mapping.userMapper">

7 <!-- 在select标签中编写查询的SQL语句， 设置select标签的id属性为getUser（类似dao中定义一个getUser的方法，但方法可以重载），id属性值必须是唯一的，不能够重复

8 使用parameterType属性指明查询时使用的参数类型，resultType属性指明查询返回的结果集类型

9 resultType="me.gacl.domain.User"就表示将查询结果封装成一个User类的对象返回

10 User类就是users表所对应的实体类

11 -->

12 <!--

13 根据id查询得到一个user对象

14 -->

15 <select id="getUser" parameterType="int"

16 resultType="me.gacl.domain.User">

17 select \* from users where id=#{id}

18 </select>

19 </mapper>

[复制代码](javascript:void(0);)

　　4、在conf.xml文件中注册userMapper.xml文件

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

3 <configuration>

4 <environments default="development">

5 <environment id="development">

6 <transactionManager type="JDBC" />

7 <!-- 配置数据库连接信息 -->

8 <dataSource type="POOLED">

9 <property name="driver" value="com.mysql.jdbc.Driver" />

10 <property name="url" value="jdbc:mysql://localhost:3306/mybatis" />

11 <property name="username" value="root" />

12 <property name="password" value="XDP" />

13 </dataSource>

14 </environment>

15 </environments>

16

17 <mappers>

18 <!-- 注册userMapper.xml文件，

19 userMapper.xml位于me.gacl.mapping这个包下，所以resource写成me/gacl/mapping/userMapper.xml-->

20 <mapper resource="me/gacl/mapping/userMapper.xml"/>

21 </mappers>

22

23 </configuration>

[复制代码](javascript:void(0);)

　　5、编写测试代码：执行定义的select语句

　　创建一个Test1类，编写如下的测试代码：

[复制代码](javascript:void(0);)

package me.gacl.test;

import java.io.IOException;

import java.io.InputStream;

import java.io.Reader;

import me.gacl.domain.User;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class Test1 {

public static void main(String[] args) throws IOException {

//mybatis的配置文件

String resource = "conf.xml";

//使用类加载器加载mybatis的配置文件（它也加载关联的映射文件）

InputStream is = Test1.class.getClassLoader().getResourceAsStream(resource);

//构建sqlSession的工厂

SqlSessionFactory sessionFactory = new SqlSessionFactoryBuilder().build(is);

//使用MyBatis提供的Resources类加载mybatis的配置文件（它也加载关联的映射文件）

//Reader reader = Resources.getResourceAsReader(resource);

//构建sqlSession的工厂

//SqlSessionFactory sessionFactory = new SqlSessionFactoryBuilder().build(reader);

//创建能执行映射文件中sql的sqlSession

SqlSession session = sessionFactory.openSession();

/\*\*

\* 映射sql的标识字符串，

\* me.gacl.mapping.userMapper是userMapper.xml文件中mapper标签的namespace属性的值，

\* getUser是select标签的id属性值，通过select标签的id属性值就可以找到要执行的SQL

\*/

String statement = "me.gacl.mapping.userMapper.getUser";//映射sql的标识字符串

//执行查询返回一个唯一user对象的sql

User user = session.selectOne(statement, 1);

System.out.println(user);

}

}