随笔- 94  文章- 9  评论- 303

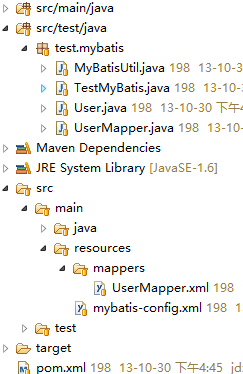
[MyBatis使用示例](http://www.cnblogs.com/zhanjindong/p/3397828.html)

原文链接：http://www.cnblogs.com/magialmoon/archive/2013/10/30/3397828.html

下面是一个简单的MyBatis使用DEMO。

**整体结构**

整体代码大致如下：



**POM依赖**

需要引用两个jar包，一个是mybatis，另一个是mysql-connector-java，如果是maven工程的话，pom里如下添加依赖就可以了。

[复制代码](javascript:void(0);)

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.2.3</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.26</version>

</dependency>

[复制代码](javascript:void(0);)

**数据准备**

在MySQL中造点数据进行测试用：

[复制代码](javascript:void(0);)

CREATE DATABASE mybatis\_test;

CREATE TABLE user

(

age INTEGER NOT NULL,

name VARCHAR(**64**) NOT NULL DEFAULT ''

);

insert user values(**18**,'zhanjindong');

insert user values(**20**,'zhangsan');

[复制代码](javascript:void(0);)

**配置文件**

需要两类配置文件，一类是MyBatis的配置文件mybatis-config.xml，示例中是一个很简单配置，详细的配置网上有很多说明。

[复制代码](javascript:void(0);)

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

<!DOCTYPE configuration

PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<settings>

<!-- changes from the defaults for testing -->

<setting name="cacheEnabled" value="false" />

<setting name="useGeneratedKeys" value="true" />

<setting name="defaultExecutorType" value="REUSE" />

</settings>

<typeAliases>

<typeAlias alias="User" type="test.mybatis.User"/>

</typeAliases>

<environments default="development">

<environment id="development">

<transactionManager type="jdbc"/>

<dataSource type="POOLED">

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

<property name="url" value="jdbc:mysql://192.168.71.38:3306/mybatis\_test"/>

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

<property name="password" value="123456"/>

</dataSource>

</environment>

</environments>

<mappers>

<mapper resource="mappers/UserMapper.xml" />

</mappers>

</configuration>

[复制代码](javascript:void(0);)

另一类是数据访问接口映射文件：示例中是UserMapper.xml。这个文件一般放在src/main/resource下面或子目录下mybatis就可以找到了，在mybatis-config.xml中由mappers/mapper节点的resource指定。

[复制代码](javascript:void(0);)

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

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="test.mybatis.UserMapper">

<!-- 这里namespace必须是UserMapper接口的路径” -->

<insert id="insertUser" parameterType="User">

insert into user(name,age) values(#{name},#{age})

<!-- 这里sql结尾不能加分号，否则报“ORA-00911”的错误 -->

</insert>

<!-- 这里的id必须和UserMapper接口中的接口方法名相同 -->

<select id="getUser" resultType="User" parameterType="java.lang.String">

select \* from user where name=#{name}

</select>

</mapper>

[复制代码](javascript:void(0);)

对应这个映射文件就是test.mybatis这个命名空间下的UserMapper这个接口，只是定义了访问数据表的接口：

[复制代码](javascript:void(0);)

package test.mybatis;

public interface UserMapper {

public void insertUser(User user);

public User getUser(String name);

}

[复制代码](javascript:void(0);)

需要一个POJO:User.java

[复制代码](javascript:void(0);)

package test.mybatis;

public class User {

private String name;

private Integer age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

public User(String name, Integer age) {

super();

this.name = name;

this.age = age;

}

public User() {

super();

}

}

[复制代码](javascript:void(0);)

**测试**

通过MyBatis操作数据库用到的是一个叫SqlSession的类，这个类是通过SqlSessionFactory产生的，一般建议在全局维护一个SqlSessionFactory就可以了。

TestMyBatis.java

[复制代码](javascript:void(0);)

package test.mybatis;

import java.io.IOException;

import java.io.Reader;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class MyBatisUtil {

private final static SqlSessionFactory sqlSessionFactory;

static {

String resource = "mybatis-config.xml";

Reader reader = null;

try {

reader = Resources.getResourceAsReader(resource);

} catch (IOException e) {

System.out.println(e.getMessage());

}

sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);

}

public static SqlSessionFactory getSqlSessionFactory() {

return sqlSessionFactory;

}

}

[复制代码](javascript:void(0);)

测试代码如下：

TestMyBatis.java

[复制代码](javascript:void(0);)

package test.mybatis;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

public class TestMyBatis {

static SqlSessionFactory sqlSessionFactory = null;

static {

sqlSessionFactory = MyBatisUtil.getSqlSessionFactory();

}

public static void main(String[] args) {

testAdd();

getUser();

}

public static void testAdd() {

SqlSession sqlSession = sqlSessionFactory.openSession();

try {

UserMapper userMapper = sqlSession.getMapper(UserMapper.class);

User user = new User("lisi", new Integer(25));

userMapper.insertUser(user);

sqlSession.commit();// 这里一定要提交，不然数据进不去数据库中

} finally {

sqlSession.close();

}

}

public static void getUser() {

SqlSession sqlSession = sqlSessionFactory.openSession();

try {

UserMapper userMapper = sqlSession.getMapper(UserMapper.class);

User user = userMapper.getUser("zhangsan");

System.out.println("name: " + user.getName() + "|age: "

+ user.getAge());

} finally {

sqlSession.close();

}

}

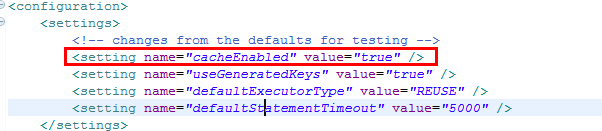
}

[复制代码](javascript:void(0);)

**注意事项**

1、mybatis会用log4j记录日志，但是开启debug模式貌似对性能影响非常厉害。

2、mybatis的查询缓存对性能的影响非常大，启用和不启用差距非常大



http://images.cnitblog.com/i/288950/201406/171417031143100.png

**注意**：mapper文件中一定要加上cache这一行，否则不生效。

**示例代码下载**

[下载示例代码。](http://files.cnblogs.com/magialmoon/MyBatisTest.rar)