**Spring与Mybatis整合环境搭建**

**本文主要介绍Spring与Mybatis的常用整合方法。**

**目录**

[一、准备 1](#_Toc320794774)

[二、配置文件 2](#_Toc320794775)

[1. 数据库连接基本信息jdbc.properties 2](#_Toc320794776)

[2.在数据库中创建表与及相应的序列 2](#_Toc320794777)

[3.创建相应的JavaBean 2](#_Toc320794778)

[4.准备表的映射文件UserTable.xml 3](#_Toc320794779)

[5. 总配置文件sqlMapConfig.xml 4](#_Toc320794780)

[6.Spring配置文件beans.xml 4](#_Toc320794781)

[7.相应的DAO 6](#_Toc320794782)

[7.1 IUserDAO.java 6](#_Toc320794783)

[7.2UseDAOImpl 7](#_Toc320794784)

[三、测试 8](#_Toc320794785)

# 一、准备

1.mybatis-spring-1.1.1.jar或者更高版本 下载地址: <http://code.google.com/p/mybatis/>

在这个jar包里有spring和mybatis的jar包需要一同导入。如果其他版本没有的话，可以去他们的官方下载。

2.如果需要对数据库操作可能还需要引用到的jar包有：commons-dbcp.jar,commons-pool.jar,commons-io.jar以及对应的数据库驱动，我这里使用的是oracle数据库，所以jar包ojdbc6.jar

3.JAVA环境

需要安装JRE或者JDK，版本最好1.6或者以上。设置好环境变量。

4.IDE工具（Eclipse 当然也可以用其他的）

# 二、配置文件

## 1. 数据库连接基本信息jdbc.properties

连接数据库的一些基本信息，放在src下

|  |
| --- |
| driverClassName=oracle.jdbc.driver.OracleDriver  url=jdbc:oracle:thin:@127.0.0.1:1521:orcl  username=admin  password=admin  maxActive=255  maxIdle=20  maxWait=100 |

具体的一些信息，你可以根据自己的数据库情况进行一定的修改。

## 2.在数据库中创建表与及相应的序列

|  |
| --- |
| Create table usertable  (  Userid int primary key,  Username varchar2(32) not null,  Password varchar2(32)  );  Create sequence usertable\_seq start with 1 increment by 1 nocahe;  Commit; |

## 3.创建相应的JavaBean

|  |
| --- |
| **package** com.springmybatis.po;  **public** **class** UserTable  {  **private** **int** userId;  **private** String userName;  **private** String password;  **public** UserTable()  {  }  **public** **int** getUserId()  {  **return** userId;  }  **public** **void** setUserId( **int** userId )  {  **this**.userId = userId;  }  **public** String getUserName()  {  **return** userName;  }  **public** **void** setUserName( String userName )  {  **this**.userName = userName;  }  **public** String getPassword()  {  **return** password;  }  **public** **void** setPassword( String password )  {  **this**.password = password;  }  } |

## 4.准备表的映射文件UserTable.xml

|  |
| --- |
| <?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=*"com.springmybatis.po.UserTable"*>  <resultMap type=*"UserTable"* id=*"userMap"*>  <id property=*"userId"* column=*"userid"*></id>  <id property=*"userName"* column=*"username"*></id>  <id property=*"password"* column=*"password"*></id>  </resultMap>  <select id=*"getUser"* parameterType=*"UserTable"* resultMap=*"userMap"*>  select userId,userName from usertable where userId=#{userId}  </select>  <insert id=*"addUser"* parameterType=*"UserTable"*>  insert into usertable(userid,username,password) values(userid\_seq.nextval,#{userName},#{password})  </insert>  <update id=*"updateUser"* parameterType=*"UserTable"*>  update usertable set username=#{userName},password=#{password} where userid=#{userId}  </update>  </mapper> |

## 5. 总配置文件sqlMapConfig.xml

对于熟悉Mybatis这个orm框架的肯定知道，我们一般都需要一个对数据库操作的总配置文件。

|  |
| --- |
| <?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>  <!-- 别名 -->  <typeAliases>  <typeAlias type=*"com.springmybatis.po.UserTable"* alias=*"UserTable"*/>  </typeAliases>  <mappers>  <mapper resource=*"com/springmybatis/po/UserTable.xml"* />  </mappers>  </configuration> |

## 6.Spring配置文件beans.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd"*>  <!-- config db property file location -->  <bean id=*"propertyConfigurer"* class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>    <property name=*"locations"*>  <list>  <value>jdbc.properties</value>  </list>  </property>  </bean>    <bean id=*"dataSource"* destroy-method=*"close"* class=*"org.apache.commons.dbcp.BasicDataSource"* >  <property name=*"driverClassName"*>  <value>${driverClassName}</value>  </property>  <property name=*"url"* >  <value>${url}</value>  </property>  <property name=*"username"* >  <value>${username} </value>  </property>  <property name=*"password"* >  <value>${password}</value>  </property>  <property name=*"maxActive"*>  <value>${maxActive}</value>  </property>  <property name=*"maxIdle"*>  <value>${maxIdle}</value>  </property>  <property name=*"maxWait"*>  <value>${maxWait}</value>  </property>  <property name=*"validationQuery"*>  <value>select count(1) from Dual</value>  </property>  <!-- test when get connection -->  <property name=*"testOnBorrow"*>  <value>true</value>  </property>  <!-- test when return connection to pool -->  <property name=*"testOnReturn"*>  <value>true</value>  </property>  <property name=*"testWhileIdle"*>  <value>true</value>  </property>  </bean>    <!-- 创建SqlSessionFactory，同时指定数据源-->  <bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"* />  <property name=*"configLocation"* value=*"classpath:sqlMapConfig.xml"*></property>  </bean>    <bean id=*"userDAO"* class=*"com.springmybatis.dao.UserDAOImpl"*>  <property name=*"sqlSessionFactory"* ref=*"sqlSessionFactory"*></property>  </bean>    </beans> |

## 7.相应的DAO

### 7.1 IUserDAO.java

|  |
| --- |
| **package** com.springmybatis.dao;  **import** com.springmybatis.po.UserTable;  **public** **interface** IUserDAO  {  /\*\*  \* add user into db  \* **@param** userTable necessary user information  \* **@return** add if succeed  \*/  **public** **boolean** addUser(UserTable userTable);    /\*\*  \* validate user  \* **@param** userTable  \* **@return**  \*/  **public** **boolean** checkUser(UserTable userTable);      **public** **boolean** updateUser( UserTable userTable );  } |

### 7.2UseDAOImpl

|  |
| --- |
| package com.springmybatis.dao;  import org.apache.ibatis.session.SqlSession;  import org.apache.ibatis.session.SqlSessionFactory;  import com.springmybatis.po.UserTable;  public class UserDAOImpl implements IUserDAO  {  private SqlSessionFactory sqlSessionFactory;  public SqlSessionFactory getSqlSessionFactory()  {  return sqlSessionFactory;  }  public void setSqlSessionFactory( SqlSessionFactory sqlSessionFactory )  {  this.sqlSessionFactory = sqlSessionFactory;  }  public boolean addUser( UserTable userTable )  {  SqlSession session = this.getSqlSessionFactory().openSession();  int i1 = session.insert( "addUser", userTable );  session.close();  return i1 != 0 ;  }  public boolean updateUser( UserTable userTable )  {  SqlSession session = this.getSqlSessionFactory().openSession();  int i = session.update( "updateUser", userTable );  return i != 0;  }  public boolean checkUser( UserTable userTable )  {  return false;  }  } |

# 三、测试

|  |
| --- |
| package com.springmybatis.dao;  import org.springframework.beans.BeansException;  import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import com. springmybatis.po.UserTable;  public class ApplicationContextBean  {  private static ApplicationContext context = null;  private final static String CONFIG\_FILE\_NAME = "beans.xml";  static  {  context = new ClassPathXmlApplicationContext( CONFIG\_FILE\_NAME );  }  public static IUserDAO getUserDAO()  {  try  {    return (IUserDAO)context.getBean( "userDAO" );  }  catch ( BeansException e )  {  return null;  }  }    public static void main( String[] args )  {  IUserDAO dao = getUserDAO();  UserTable userTable = new UserTable();  userTable.setPassword( "qqqr123" );  userTable.setUserName( "test" );  boolean result = dao.addUser( userTable );  System.out.println(result);  }  } |

如果没有什么问题的话，应该会打印出true. 祝你成功。