MyBatis的CRUD

一、配置文件

|  |
| --- |
| <?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">  <!--  namespace:命名空间，类似java的包机制  -->  <mapper namespace=*"com.yk.pojo.User"*>  <!-- 查询所有-->  <select id=*"selectUser"* resultType=*"com.yk.pojo.User"*>  select \* from user  </select>    <!-- 根据ID查询一条信息  id:唯一的标识，一个映射文件中不允许出现两个相同的id  parameterType:传入参数的类型  resultType:返回结果类型  sql中利用#{属性名}的形式来传递参数，此时会在参数值前后自动拼接单引号,在某些情况下，你不希望自动给我们的参数值前后添加单引号，可以使用${属性名}来处理，一般这种写法不常用  -->  <select id=*"selectById"* parameterType=*"int"* resultType=*"com.yk.pojo.User"*>  select \* from user where id=#{id}  </select>    <!-- 插入 -->    <insert id=*"insertUser"* parameterType=*"com.yk.pojo.User"*>  insert into user(  username,  userpassword  ) values(  #{username},  #{userpassword}  )  </insert>    <!-- 修改 -->  <update id=*"updateUser"* parameterType=*"com.yk.pojo.User"*>  update user set username=#{username},userpassword=#{userpassword} where id=#{id}  </update>    <!-- 删除 -->  <delete id=*"deleteUser"* parameterType=*"java.lang.Integer"*>  delete from user where id=#{id}  </delete>  </mapper> |

二、测试编码

|  |
| --- |
| **public** **class** Test {  **public** **static** **void** main(String[] args) **throws** IOException {  //1.设置配置文件路径，从src目录下开始，  String resource="mybatis-config.xml";    //2.使用ibatis自带的Resources类获取配置文件的字符流  Reader reader=Resources.*getResourceAsReader*(resource);    //3.使用SqlSessionFactroyBuilder()实例的builder()方法来创建一个SqlSessionFactroy工厂  SqlSessionFactory sqlSessionFactory=**new** SqlSessionFactoryBuilder().build(reader);    //4.通过工厂类来打开一次数据库的会话  SqlSession sqlSession=sqlSessionFactory.openSession();      **try** {    //---================================查询  //1.获取所有的新闻集合  //实体类+方法名  /\*List<User> lists=sqlSession.selectList("com.yk.pojo.User.selectUser");  for (User user:lists) {  System.out.println(user);  }\*/      //2.获取传入ID的实体  /\*User user=sqlSession.selectOne("com.yk.pojo.User.selectById",3);  System.out.println(user);\*/    //-----------=========================修改  //3.插入信息    /\*User user=new User();  user.setUsername("test");  user.setUserpassword("123");  int result=sqlSession.insert("com.yk.pojo.User.insertUser",user);  //mybatis中,默认不开启事务的自动提交,所以你insert了,但是系统认为你的一连串操作没有完成,为了保障数据的一致性,系统不会加入这条新的数据,只有当你手动提交的时候,数据库才会加入数据  sqlSession.commit(); //提交事务  if(result>0){  System.out.println("插入成功");  }\*/        //-----------========================修改    /\*User user=new User();  user.setId(3);  user.setUsername("test2");  user.setUserpassword("456");  int results=sqlSession.update("com.yk.pojo.User.updateUser",user);  sqlSession.commit();  if(results>0){  System.out.println("修改成功");  }\*/      //----------==========================删除  **int** results=sqlSession.delete("com.yk.pojo.User.deleteUser",9);  sqlSession.commit();  **if**(results>0){  System.***out***.println("删除成功");  }    } **catch** (Exception e) {  //事务回滚  sqlSession.rollback();  e.printStackTrace();  }**finally** {  //关闭会话  sqlSession.close();  }  }  } |