闲置宝测试文档

一.Mapper层测试

1.EvaluationsMapper测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Evaluations;

import edu.fzu.se.backend.mapper.EvaluationsMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

@SpringBootTest

class EvaluationsMapperTest {

@Autowired

private EvaluationsMapper evaluationsMapper;

@Test

void selectAll() {

List<Evaluations> evaluations = evaluationsMapper.selectAll();

assertNotNull(evaluations);

}

@Test

void selectByTradeId() {

Long tradeId = 1000000001L;

Evaluations evaluations = evaluationsMapper.selectByTradeId(tradeId);

assertNotNull(evaluations);

assertEquals(tradeId, evaluations.getTrade\_ID());

}

@Test

void selectBySellerId() {

Long sellerId = 100000005L;

List<Evaluations> evaluations = evaluationsMapper.selectBySellerId(sellerId);

assertNotNull(evaluations);

assertTrue(evaluations.size() > 0);

}

@Test

void selectByBuyerId() {

Long buyerId = 100000002L;

List<Evaluations> evaluations = evaluationsMapper.selectByBuyerId(buyerId);

assertNotNull(evaluations);

assertTrue(evaluations.size() > 0);

}

@Test

void selectByBuyerAndSellerId() {

Long buyerId = 100000005L;

Long sellerId = 100000004L;

List<Evaluations> evaluations = evaluationsMapper.selectByBuyerAndSellerId(buyerId, sellerId);

assertNotNull(evaluations);

assertTrue(evaluations.size() > 0);

}

@Test

void deleteById() {

Long tradeId = 1000000001L;

int rowsAffected = evaluationsMapper.deleteById(tradeId);

assertEquals(1, rowsAffected);

}

@Test

void insertEvaluation() {

Evaluations evaluation = new Evaluations();

evaluation.setTrade\_ID(1000000001L);

evaluation.setBuyer\_Evaluation("好评");

evaluation.setSeller\_Evaluation("非常好评");

int rowsAffected = evaluationsMapper.insertEvaluation(evaluation);

assertEquals(1, rowsAffected);

}

@Test

void updateById() {

Evaluations evaluation = new Evaluations();

evaluation.setTrade\_ID(1000000001L);

evaluation.setBuyer\_Evaluation("好评");

evaluation.setSeller\_Evaluation("非常好评");

int rowsAffected = evaluationsMapper.updateById(evaluation);

assertEquals(1, rowsAffected);

}

}

测试结果截图：

文本

描述已自动生成

2.FavoritesMapper层的测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Favorites;

import edu.fzu.se.backend.bean.Goods;

import edu.fzu.se.backend.mapper.FavoritesMapper;

import edu.fzu.se.backend.mapper.GoodsMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

//import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

class FavoritesMapperTest {

@Autowired

private FavoritesMapper favoritesMapper;

private GoodsMapper goodsMapper;

@Test

void selectAll() {

List<Favorites> favorites = favoritesMapper.selectAll();

assertNotNull(favorites);

assertTrue(!favorites.isEmpty());

}

@Test

void selectById() {

Long favoriteId = 1000000001L;

Favorites favorite = favoritesMapper.selectById(favoriteId);

assertNotNull(favorite);

assertEquals(favoriteId, favorite.getFavorite\_ID());

}

@Test

void selectByUserId() {

Long userId = 100000001L;

List<Favorites> favorites = favoritesMapper.selectByUserId(userId);

assertNotNull(favorites);

}

@Test

void selectByKeyword() {

String keyword = "台";

List<Favorites> favorites = favoritesMapper.selectByKeyword(keyword);

assertNotNull(favorites);

assertTrue(!favorites.isEmpty());

}

@Test

void deleteById() {

Long favoriteId = 1000000001L;

int result = favoritesMapper.deleteById(favoriteId);

assertEquals(1, result);

}

@Test

void insertFavorite() {

Favorites favorite = new Favorites();

favorite.setFavorite\_ID(1000000001L);

favorite.setUser\_ID(100000001L);

favorite.setGoods\_ID(1000000001L);

favorite.setInsert\_Time("2022-01-01 00:00:00");

int result = favoritesMapper.insertFavorite(favorite);

assertEquals(1, result);

}

@Test

void updateById() {

Favorites favorite = new Favorites();

favorite.setFavorite\_ID(1000000002L);

favorite.setUser\_ID(100000002L);

favorite.setGoods\_ID(1000000002L);

favorite.setInsert\_Time("2022-01-01 00:00:00");

int result = favoritesMapper.updateById(favorite);

assertEquals(1, result);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序

描述已自动生成

3.Goods\_Images\_ConnsMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Goods\_Images\_Conns;

import edu.fzu.se.backend.mapper.Goods\_Images\_ConnsMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

class Goods\_Images\_ConnsMapperTest {

@Autowired

private Goods\_Images\_ConnsMapper goodsImagesConnsMapper;

@Test

void selectByGoodsAndImageIds() {

Long goodsId = 1000000005L;

Long imageId = 1L;

Goods\_Images\_Connsresult=goodsImagesConnsMapper.selectByGoodsAndImageIds(goodsId, imageId);

assertEquals(goodsId, result.getGoods\_ID());

assertEquals(imageId, result.getImage\_ID());

}

@Test

void selectByGoodsId() {

Long goodsId = 1000000005L;

List<Goods\_Images\_Conns> resultList = goodsImagesConnsMapper.selectByGoodsId(goodsId);

assertNotNull(resultList);

assertTrue(!resultList.isEmpty());

}

@Test

void insertConnection() {

Goods\_Images\_Conns connection = new Goods\_Images\_Conns();

connection.setGoods\_ID(1000000004L);

connection.setImage\_ID(1L);

int rowsAffected = goodsImagesConnsMapper.insertConnection(connection);

assertTrue(rowsAffected > 0);

}

@Test

void deleteConnection() {

Long goodsId = 1000000004L;

Long imageId = 1L;

int rowsAffected = goodsImagesConnsMapper.deleteConnection(goodsId, imageId);

assertTrue(rowsAffected > 0);

}

}

测试结果截图：

文本

描述已自动生成

4.GoodsMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Goods;

import edu.fzu.se.backend.bean.Images;

import edu.fzu.se.backend.mapper.GoodsMapper;

import edu.fzu.se.backend.mapper.ImagesMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

class GoodsMapperTest {

@Autowired

private GoodsMapper goodsMapper;

private ImagesMapper imagesMapper;

@Test

void selectAll() {

List<Goods> goodsList = goodsMapper.selectAll();

assertNotNull(goodsList);

assertTrue(!goodsList.isEmpty());

}

@Test

void deleteById() {

Long goodsId = 1000000014L;

int result = goodsMapper.deleteById(goodsId);

assertEquals(1, result);

}

@Test

void selectImagesByGoodsId() {

Long goodsId = 1000000001L;

List<Images> images = goodsMapper.selectImagesByGoodsId(goodsId);

assertNotNull(images);

assertTrue(!images.isEmpty());

}

@Test

void getHomePageGoods() {

int searchid =5;

List<Goods> actualGoods = goodsMapper.getHomePageGoods(searchid);

assertNotNull(actualGoods);

assertEquals(5, actualGoods.size());

}

@Test

void selectById() {

Long goodsId = 1000000001L;

Goods goods = goodsMapper.selectById(goodsId);

assertNotNull(goods);

assertEquals(goodsId, goods.getGoods\_ID());

}

@Test

void selectBySellerId() {

Long sellerId = 100000001L;

List<Goods> goodsList = goodsMapper.selectBySellerId(sellerId);

assertNotNull(goodsList);

assertTrue(!goodsList.isEmpty());

}

@Test

void selectByPriceRange() {

int minPrice = 10;

int maxPrice = 100;

List<Goods> goodsList = goodsMapper.selectByPriceRange(minPrice, maxPrice);

assertNotNull(goodsList);

assertTrue(!goodsList.isEmpty());

for (Goods goods : goodsList) {

assertTrue(goods.getGoods\_Price() >= minPrice && goods.getGoods\_Price() <= maxPrice);

}

}

@Test

void selectByKeyword() {

String keyword = "红";

List<Goods> goodsList = goodsMapper.selectByKeyword(keyword);

assertNotNull(goodsList);

}

@Test

void selectGoodsByUserName() {

String userName = "小美";

List<Goods> goodsList = goodsMapper.selectGoodsByUserName(userName);

assertNotNull(goodsList);

}

@Test

void insertGood() {

Goods goods = new Goods();

goods.setGoods\_ID(1000000015L);

goods.setGoods\_Name("乒乓球");

goods.setSeller\_ID(100000005L);

goods.setGoods\_Price(250.00);

goods.setGoods\_Description("这是一个测试商品");

goods.setClassification("交通工具");

goods.setRelease\_Time("2023-11-11 00:00:00");

int result = goodsMapper.insertGood(goods);

assertEquals(1, result);

}

@Test

void updateById() {

Goods goods = new Goods();

goods.setGoods\_ID(1000000004L);

goods.setSeller\_ID(100000004L);

goods.setGoods\_Name("测试商品");

goods.setGoods\_Price(100.00);

goods.setGoods\_Description("这是一个测试商品");

goods.setClassification("文娱用品");

goods.setRelease\_Time("2023-11-12 12:07:52");

int result = goodsMapper.updateById(goods);

assertEquals(1, result);

}

}

测试结果截图：

文本

描述已自动生成

5.HistoryMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.History;

import edu.fzu.se.backend.mapper.HistoryMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

//import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

@SpringBootTest

class HistoryMapperTest {

@Autowired

private HistoryMapper historyMapper;

@Test

void selectAll() {

List<History> histories =historyMapper.selectAll();

assertNotNull(histories);

assertTrue(!histories.isEmpty());

}

@Test

void selectById() {

Long searchID = 1000000001L;

History history = historyMapper.selectById(searchID);

assertNotNull(history);

assertEquals(searchID,history.getSearch\_ID());

}

@Test

void deleteById() {

Long searchID = 1000000014L;

int result =historyMapper.deleteById(searchID);

assertEquals(1,result);

}

@Test

void insertHistory() {

History history = new History();

history.setUser\_ID(100000005L);

history.setSearch\_ID(1000000016L);

history.setKeyword("足球");

int result = historyMapper.insertHistory(history);

assertEquals(1,result);

}

@Test

void updateById() {

History history =new History();

history.setSearch\_ID(1000000001L);

history.setUser\_ID(100000002L);

history.setKeyword("篮球");

int result = historyMapper.updateById(history);

assertEquals(1,result);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序

描述已自动生成

6.ImagesMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Evaluations;

import edu.fzu.se.backend.bean.Images;

import edu.fzu.se.backend.mapper.ImagesMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

//import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

@SpringBootTest

class ImagesMapperTest {

@Autowired

private ImagesMapper imagesMapper;

@Test

void testSelectAll() {

List<Images> images = imagesMapper.selectAll();

assertNotNull(images);

System.out.println(images);

assertTrue(!images.isEmpty());

}

@Test

void testSelectById() {

Long imageId = 7L;

Images images = imagesMapper.selectById(imageId);

assertNotNull(images);

assertEquals(imageId, images.getImage\_ID());

}

@Test

void InsertImage() {

Images images = new Images();

images.setImage\_ID(6L);

images.setImage\_Name("用户头像1.jpg");

images.setImage\_Data("base128编码的图片数据");

images.setImage\_Class("平台");

int result = imagesMapper.insertImage(images);

assertEquals(1, result);

}

@Test

void testUpdateImageById() {

Images images = new Images();

images.setImage\_ID(1L);

images.setImage\_Name("用户头像1.jpg");

images.setImage\_Data("base128编码的图片数据");

images.setImage\_Class("平台");

int result = imagesMapper.updateImageById(images);

assertEquals(1, result);

}

@Test

void deleteById() {

Long searchID = 9L;

int result =imagesMapper.deleteById(searchID);

assertEquals(1,result);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序

描述已自动生成

7.MessagesMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Messages;

import edu.fzu.se.backend.mapper.MessagesMapper;

import edu.fzu.se.backend.mapper.ShoppingCartMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

class MessagesMapperTest {

@Autowired

private MessagesMapper messagesMapper;

@Test

void selectAll() {

List<Messages> messages = messagesMapper.selectAll();

assertNotNull(messages);

assertTrue(!messages.isEmpty());

}

@Test

void selectById() {

Long Message\_ID = 1000000001L;

Messages message =messagesMapper.selectById(Message\_ID);

assertNotNull(message);

}

@Test

void selectBySenderAndReceiverIds() {

List<Messages> result = messagesMapper.selectBySenderAndReceiverIds(100000001L,100000002L);

assertEquals(1, result.size());

}

@Test

void deleteById() {

Long Message\_ID = 1000000006L;

int result = messagesMapper.deleteById(Message\_ID);

assertEquals(1,result);

}

@Test

void insertMessage() {

Messages message = new Messages();

message.setMessage\_ID(1000000006L);

message.setSender\_ID(100000001L);

message.setReceiver\_ID(100000002L);

message.setMessage\_Content("你的铅笔还有吗？");

message.setTime("2022-11-11 00:00:00");

message.setIsRead(true);

int result = messagesMapper.insertMessage(message);

assertEquals(1,result);

}

@Test

void updateById() {

Messages message = new Messages();

message.setMessage\_ID(1000000007L);

message.setSender\_ID(100000002L);

message.setReceiver\_ID(100000001L);

message.setMessage\_Content("你的水彩笔还有吗？");

message.setTime("2022-11-11 00:00:00");

message.setIsRead(true);

int result = messagesMapper.updateById(message);

assertEquals(1,result);

}

}

测试结果截图：

图形用户界面, 文本

描述已自动生成

8.ShoppingCartMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.ShoppingCart;

import edu.fzu.se.backend.mapper.ShoppingCartMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

class ShoppingCartMapperTest {

@Autowired

private ShoppingCartMapper shoppingCartMapper;

@Test

void selectByUserIdAndGoodsId() {

ShoppingCart result = shoppingCartMapper.selectByUserIdAndGoodsId(100000001L, 1000000001L);

assertNotNull(result);

assertEquals(100000001L, result.getUser\_ID());

assertEquals(1000000001L, result.getGoods\_ID());

}

@Test

void selectByUserId() {

List<ShoppingCart> result = shoppingCartMapper.selectByUserId(100000004L);

assertNotNull(result);

assertTrue(result.size() > 0);

for (ShoppingCart cart : result) {

assertEquals(100000004L, cart.getUser\_ID());

}

}

@Test

void insertShoppingCart() {

ShoppingCart newShoppingCart = new ShoppingCart();

newShoppingCart.setUser\_ID(100000005L);

newShoppingCart.setGoods\_ID(1000000005L);

newShoppingCart.setCount(1);

int result = shoppingCartMapper.insertShoppingCart(newShoppingCart);

assertEquals(1, result);

}

@Test

void updateShoppingCartCount() {

ShoppingCart updatedShoppingCart = new ShoppingCart();

updatedShoppingCart.setUser\_ID(100000004L);

updatedShoppingCart.setGoods\_ID(1000000005L);

updatedShoppingCart.setCount(2);

int result = shoppingCartMapper.updateShoppingCartCount(updatedShoppingCart);

assertEquals(1, result);

}

@Test

void deleteShoppingCart() {

int result = shoppingCartMapper.deleteShoppingCart(100000005L, 1000000005L);

assertEquals(1, result);

}

@Test

void clearShoppingCart() {

int result = shoppingCartMapper.clearShoppingCart(100000005L);

assertTrue(result >= 0);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

9.TradesMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.Trades;

import edu.fzu.se.backend.mapper.TradesMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

//import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

@SpringBootTest

class TradesMapperTest {

@Autowired

private TradesMapper tradesMapper;

@Test

void selectAll() {

List<Trades> trades = tradesMapper.selectAll();

assertNotNull(trades);

}

@Test

void selectById() {

Long tradeId = 1000000001L;

Trades trade = tradesMapper.selectById(tradeId);

assertNotNull(trade);

assertEquals(tradeId, trade.getTrade\_ID());

}

@Test

void selectByBuyerAndSellerIds() {

Long buyerId = 100000002L;

Long sellerId = 100000001L;

List<Trades> trades = tradesMapper.selectByBuyerAndSellerIds(buyerId, sellerId);

assertNotNull(trades);

}

@Test

void deleteById() {

Long tradeId = 1000000006L;

int result = tradesMapper.deleteById(tradeId);

assertEquals(1, result);

}

@Test

void insertTrade() {

Trades trade = new Trades();

trade.setTrade\_ID(1000000006L);

trade.setBuyer\_ID(100000001L);

trade.setSeller\_ID(100000005L);

trade.setTrade\_Time("2022-11-11 00:00:00");

trade.setPaid\_Amount(100.0);

int result = tradesMapper.insertTrade(trade);

assertEquals(1, result);

}

@Test

void updateById() {

Trades trade = new Trades();

trade.setTrade\_ID(1000000005L);

trade.setTrade\_Time("2022-11-11 00:00:00");

trade.setPaid\_Amount(60.0);

int result = tradesMapper.updateById(trade);

assertEquals(1, result);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序

描述已自动生成

10.WxUserMapper层测试

package edu.fzu.se.backend.Mapper;

import edu.fzu.se.backend.bean.WxUser;

import edu.fzu.se.backend.mapper.WxUserMapper;

import jakarta.annotation.Resource;

import org.apache.ibatis.annotations.Mapper;

import org.junit.jupiter.api.Test;

//import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.context.TestConfiguration;

import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.junit.jupiter.api.Assertions.\*;

import org.springframework.context.annotation.Configuration;

//@SpringJUnitConfig(locations = "classpath:spring-boot-test.xml")

@SpringBootTest

class WxUserMapperTest {

@Autowired

private WxUserMapper wxUserMapper;

@Test

void selectAll() {

List<WxUser> users = wxUserMapper.selectAll();

assertNotNull(users);

}

@Test

void selectById() {

Long userId = 100000001L;

WxUser user = wxUserMapper.selectById(userId);

assertNotNull(user);

assertEquals(userId, user.getUser\_ID());

}

@Test

void selectByFzuKey() {

String fzuKey = "102101519";

WxUser user = wxUserMapper.selectByFzuKey(fzuKey);

assertNotNull(user);

assertEquals(fzuKey, user.getFZU\_Key());

}

@Test

void selectByKeyword() {

String keyword = "小";

List<WxUser> users = wxUserMapper.selectByKeyword(keyword);

assertNotNull(users);

}

@Test

void deleteById() {

Long userId = 100000011L;

int result = wxUserMapper.deleteById(userId);

assertEquals(1, result);

}

@Test

void insertUser() {

WxUser user = new WxUser();

user.setUser\_Name("小易");

user.setUser\_Key("password8");

user.setFZU\_Key("102101526");

user.setHead\_Portrait("6");

user.setDelivery\_Address("福州大学1号楼909室");

int result = wxUserMapper.insertUser(user);

assertEquals(1, result);

}

@Test

void updateById() {

WxUser user = new WxUser();

user.setUser\_ID(100000001L);

user.setUser\_Name("小何");

user.setUser\_Key("password14");

user.setHead\_Portrait("8");

user.setDelivery\_Address("福州大学1号楼110室");

int result = wxUserMapper.updateById(user);

assertEquals(1, result);

}

}

测试结果截图：

图形用户界面, 文本, 应用程序

描述已自动生成

后端接口截图：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

一．购物车控制器

1. 获取所有购物车项

图形用户界面, 文本

中度可信度描述已自动生成

1. 更新购物车项数量

背景图案

中度可信度描述已自动生成

1. 删除购物车项

图形用户界面

描述已自动生成

1. 插入新的购物车项

背景图案

描述已自动生成

1. 根据用户id查询购物车列表

背景图案

描述已自动生成

1. 清空购物车

背景图案

描述已自动生成

二．评估控制器

1.根据交易id查询评价记录

背景图案

描述已自动生成

2.根据交易id更新评价记录

电脑萤幕的截图

描述已自动生成

3.根据交易id删除评价记录

背景图案

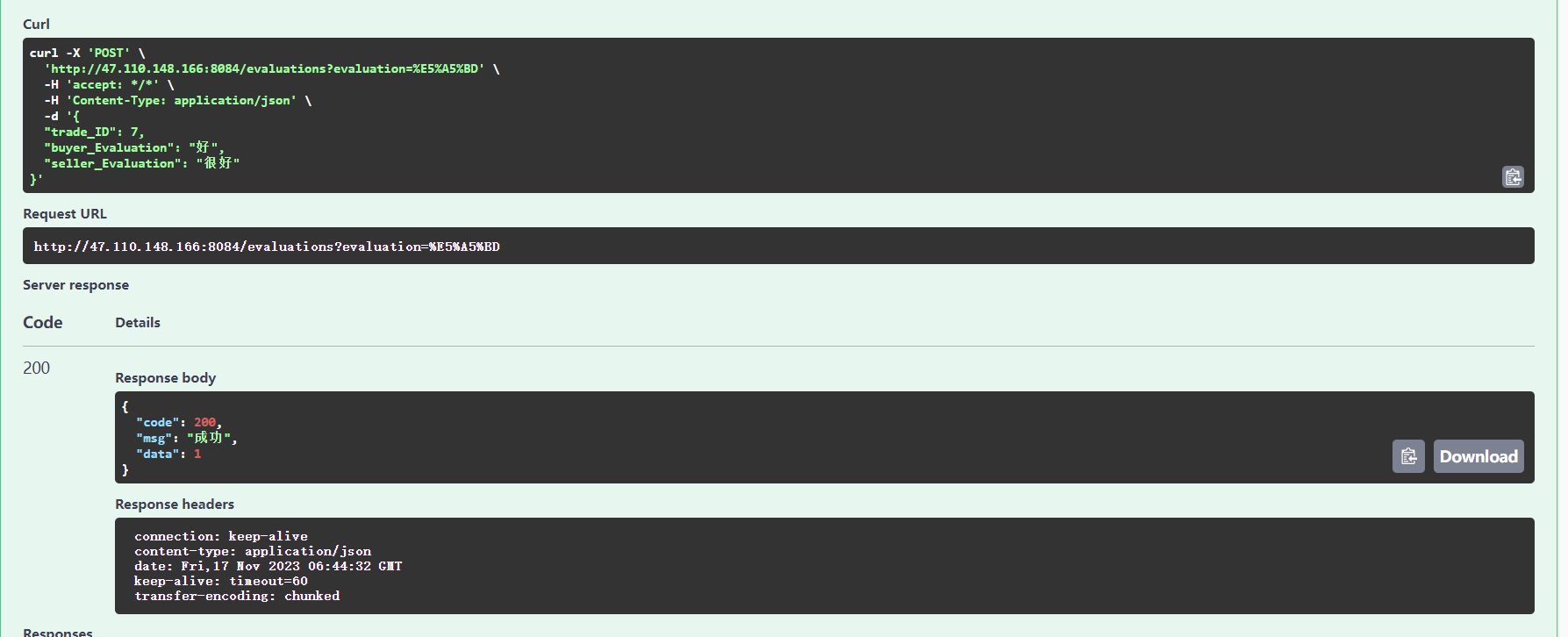
描述已自动生成

4.获取所有评价记录

文本

描述已自动生成

1. 插入评价记录



6.根据卖家id查询评价记录

图形用户界面, 背景图案

中度可信度描述已自动生成

1. 根据买家id查询评价记录

图形用户界面, 应用程序

描述已自动生成

8.根据买家和卖家id查询评价记录

背景图案

中度可信度描述已自动生成

三．图片上传控制器

1.上传图片



四．收藏夹控制器

1.根据收藏id进行查询

背景图案

描述已自动生成

2.根据收藏id更新收藏信息

背景图案

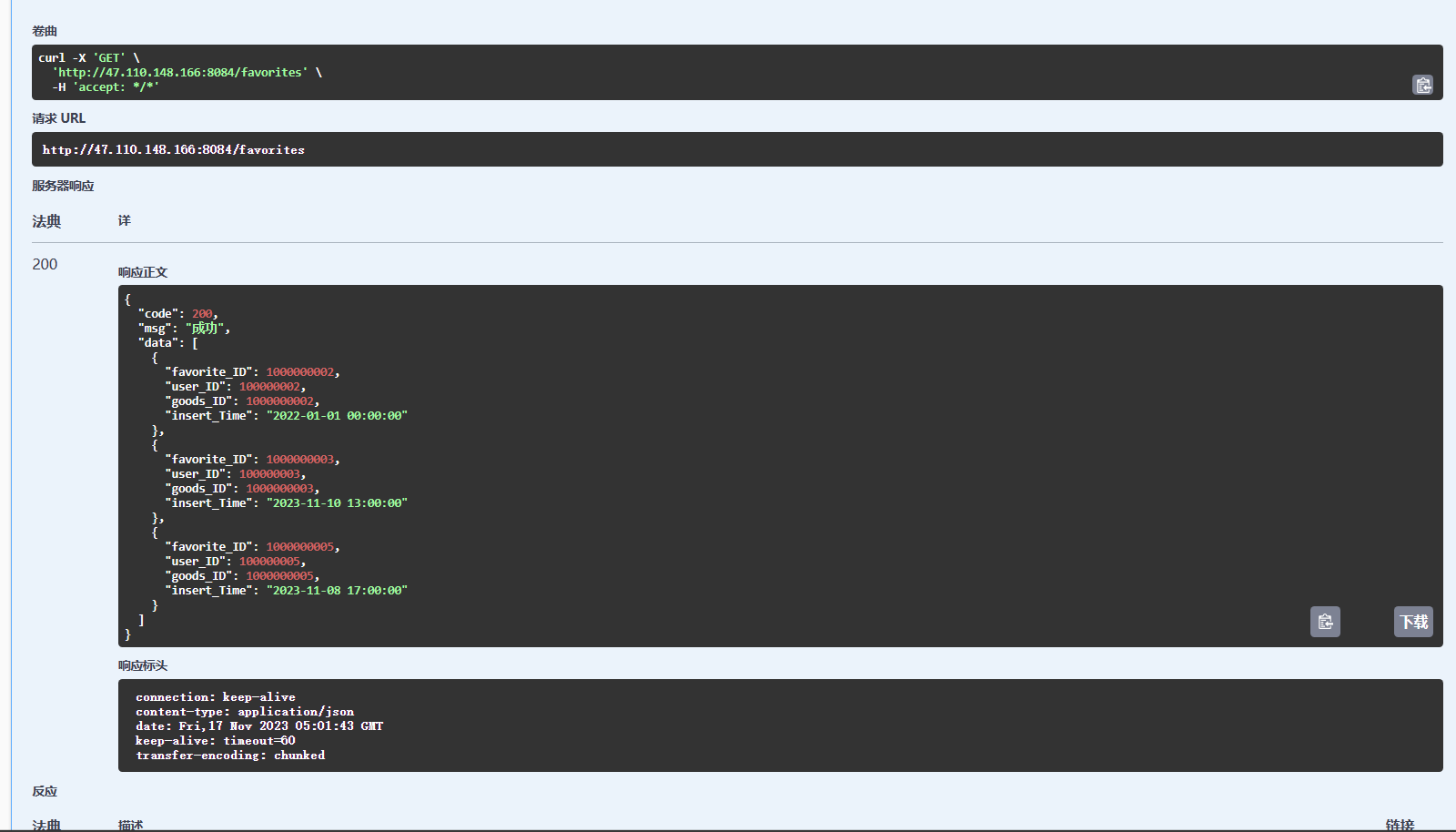
描述已自动生成

3.根据收藏id删除收藏

背景图案

描述已自动生成

4.获取所有收藏信息



5.插入收藏信息

背景图案

描述已自动生成

6.根据用户id进行查询

背景图案

描述已自动生成

7.根据收藏商品的关键字进行模糊匹配查询

背景图案

描述已自动生成

五．历史控制器

1.更新历史记录

背景图案

描述已自动生成

1. 插入历史记录



3.根据用户ID获取历史记录

背景图案

描述已自动生成

4.根据用户id获取最新的10条历史记录

背景图案

低可信度描述已自动生成

5.根据id获取历史记录

背景图案

描述已自动生成

6.根据id删除历史记录

背景图案

描述已自动生成

7.获取所有历史记录

文本

描述已自动生成

六．交易控制器

1.发布商品



七．Wxuser控制器

1.注册



2.登录

图形用户界面

低可信度描述已自动生成