# 微信小程序O2O商城后台开发梳理

1. 微信后台用户服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的用户信息、用户登录或登出等接口代码，实现用户模块相关功能。

1. 微信小程序商品服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的商品查询、 搜索或获取商品等接口代码，实现商品模块相关功能对接。

1. 微信小程序订单服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的订单售后、订单详情或订单状态等接口代码，实现订单模块相关功能对接。

## 用户登录

JS前端代码

onLaunch: function () {

console.log('App Launch')

wx.login({

success: function (res) {

if (res.code) {

wx.request({

url: "http://localhost:8080/xueqing-client/wechat/onLogin",

data: {

code: res.code

}

})

} else {

console.log('登录失败！' + res.errMsg)

}

}

})

},

Java后端代码

**package** com.xiandian.cloud.web;

@Controller

@RequestMapping("wechat")

**public** **class** WeChatController {

**private** **static** **final** String ***APPID*** = "wx73b63ccf190eb39f";

**private** **static** **final** String ***SECRET*** = "dac4d5358626c7ef0f6773d17c619095";

@ResponseBody

@RequestMapping("onLogin")

**public** String onLogin(@RequestParam(value="code",required=**false**) String code) {

String url = "https://api.weixin.qq.com/sns/jscode2session?appid=" + ***APPID*** + "&secret=" + ***SECRET*** + "&js\_code="

+ code + "&grant\_type=authorization\_code";

RestTemplate restTemplate = **new** RestTemplate();

ResponseEntity<String> responseEntity = restTemplate.exchange(url, HttpMethod.***GET***, **null**, String.**class**);

**if** (responseEntity != **null** && responseEntity.getStatusCode() == HttpStatus.***OK***) {

String data = responseEntity.getBody();

System.***out***.println(data);

}

**return** **null**;

}

}

Java控制台输出

{"session\_key":"SSUHUfo4b\/N6yxNf9Ce4AA==","openid":"oAl8F5iAvHFc\_FRLYlOavEw7\_aRk"}

## 用户信息

微信小程序以前获取用户信息是登录后嵌套获取代码，现在已改为按钮手动获取

简单版前端代码

.wxml文件

<button open-type='getUserInfo' bindgetuserinfo='bindGetUserInfo'>获取用户信息</button>

.js文件

bindGetUserInfo:function(e){

console.log(e.detail.userInfo)

}

微信小程序工具输出结果

{nickName: "…", gender: 1, language: "zh\_CN", city: "Yibin", province: "Sichuan", …}

如果要在小程序中显示

.wxml文件

<text>{{nickName}}</text>

.js文件

Data:{

nickName:“123”

}

bindGetUserInfo:function(e){

var that:this

that.setData({

nickName:e.detail.userInfo.nickName

})

}

## 显示所有商品

Sql代码

/\*  
MySQL Data Transfer  
Source Host: 192.168.0.108  
Source Database: wachat\_mini\_program Target Host: 192.168.0.108  
Target Database: wachat\_mini\_program Date: 2019/5/5 15:41:00  
\*/

SET FOREIGN\_KEY\_CHECKS=0; -- ----------------------------  
-- Table structure for commodity

-- ---------------------------- CREATE TABLE `commodity` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT, `content` varchar(255) DEFAULT NULL,  
`flag` int(11) DEFAULT NULL,

|  |
| --- |
| `img` varchar(255) DEFAULT NULL, `imgs` varchar(255) DEFAULT NULL, `name` varchar(255) DEFAULT NULL, `price` varchar(255) DEFAULT NULL, PRIMARY KEY (`id`)  ) ENGINE=MyISAM AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;  -- ---------------------------- -- Records |
| -- ----------------------------  INSERT INTO `commodity` VALUES ('1', 'Android 云存储客户端开发', '0', '/imgs/android-down.jpg', '/imgs/android-down.jpg,/imgs/android-down.jpg', 'Android 云存储客户端开发', '48.00');  INSERT INTO `commodity` VALUES ('2', 'Docker 容器技术与应用', '0', '/imgs/docker-down.jpg', '/imgs/docker-down.jpg,/imgs/docker-down.jpg', 'Docker 容 器技术与应用', '60.00');  INSERT INTO `commodity` VALUES ('3', 'Hadoop 大数据平台构建和应用', '0', '/imgs/hadoop-down.jpg', '/imgs/android-down.jpg,/imgs/hadoop-down.jpg', 'Hadoop 大数据平台构建和应用', '75.00');  INSERT INTO `commodity` VALUES ('4', ' 云 存 储 技 术 与 应 用 ', '0', '/imgs/ycc-down.jpg', '/imgs/ycc-down.jpg,/imgs/ycc-down.jpg', '云存储技术与应用', '40.00');  INSERT INTO `commodity` VALUES ('5', '软件定义网络(SDN)技术与应用', '0', '/imgs/sdn-down.jpg', '/imgs/sdn-down.jpg,/imgs/sdn-down.jpg', '软件定义网络 (SDN)技术与应用', '56.00'); |

JS前端代码

data: {

goods: [ ]

}

onShow: function () {

this.getlist();

}

getlist: function (e) {

var that=this

wx.request({

url: "http://localhost:8080/xueqing-client/wechat/commodity",

success: function (res) {

let list = res.data.other

that.setData({

goods:list

})

}

})

}

Java后端代码

Commodity.java

**package** com.xiandian.cloud.entity.core;

@Entity

@Table(name="commodity")

**public** **class** Commodity **extends** BaseEntity{

**private** String content;

**private** **int** flag;

**private** String img;

**private** String imgs;

**private** String name;

**private** String price;

**public** String getContent() {

**return** content;

}

**public** **void** setContent(String content) {

**this**.content = content;

}

**public** **int** getFlag() {

**return** flag;

}

**public** **void** setFlag(**int** flag) {

**this**.flag = flag;

}

**public** String getImg() {

**return** img;

}

**public** **void** setImg(String img) {

**this**.img = img;

}

**public** String getImgs() {

**return** imgs;

}

**public** **void** setImgs(String imgs) {

**this**.imgs = imgs;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

}

CommodityDao.java

**package** com.xiandian.cloud.dao.core;

@Repository

**public** **class** CommodityDao **extends** BaseDao<Commodity> {

**public** List<Commodity> getAll(){

String hql="from Commodity";

**return** find(hql);

}

}

CommodityService.java

**package** com.xiandian.cloud.service;

@Service

**public** **class** CommodityService {

@Autowired

**private** CommodityDao commodityDao;

**public** List<Commodity> getAll(){

**return** commodityDao.getAll();

}

}

WechatController.java

**package** com.xiandian.cloud.web;

@Controller

@RequestMapping("wechat")

**public** **class** WeChatController {

@Autowired

**private** CommodityService commodityService;

@ResponseBody

@RequestMapping("commodity")

**public** MessageBean getCommodityList() {

**return** **new** MessageBean(**true**,"",commodityService.getAll());

}

}

微信小程序效果



## 搜索查询商品

.wxml文件

<form bindsubmit='formSubmit'>

<input type='text' name="keyword"></input>

<button formType='submit'>搜索</button>

</form>

.js文件

formSubmit:function(e){

var that=this;

wx.request({

url: "http://localhost:8080/xueqing-client/wechat/search",

data: {

keyword: e.detail.value.keyword

},

success:function(res){

let list=res.data.other

that.setData({

goods:list

})

}

})

}

Java后端代码

CommodityDao.java

**package** com.xiandian.cloud.dao.core;

@Repository

**public** **class** CommodityDao **extends** BaseDao<Commodity> {

**public** List<Commodity> search(String keyword) {

String hql = "from Commodity commodity where commodity.name like '%" + keyword+"%'";

**return** find(hql);

}

}

CommodityService.java

**package** com.xiandian.cloud.service;

@Service

**public** **class** CommodityService {

@Autowired

**private** CommodityDao commodityDao;

**public** List<Commodity> Search(String keyword){

**return** commodityDao.search(keyword);

}

}

**package** com.xiandian.cloud.web;

@Controller

@RequestMapping("wechat")

**public** **class** WeChatController {

@Autowired

**private** CommodityService commodityService;

@ResponseBody

@RequestMapping("search")

**public** MessageBean search(@RequestParam(value = "keyword",required = **false**) String keyword) {

**return** **new** MessageBean(**true**, "", commodityService.Search(keyword));

}

}

微信小程序效果

图片包含 屏幕截图

描述已自动生成

## 订单

订单服务网上没有没有相关资料，但我觉得逻辑上应该和前面的差不多，都是从数据库读取，然后通过小程序和Java的接口获取，最后setData