**民宿小程序接口API文档**

**(内部资料, 注意保密)**

**文档概述**,此文档主要用于描述给花啦民宿小程序提供的接口说明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本编号 | 更新内容 | 操作人 | 更新时间 |
| V1.0 | 初稿 | 王集权 | 2019年5月31日 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[一、 新手指南 3](#_Toc10647412)

[1. 简介 3](#_Toc10647413)

[2. 名词解释 3](#_Toc10647414)

[二、 对接前准备工作说明 4](#_Toc10647415)

[1. 准备工作 4](#_Toc10647416)

[2. 开始对接 4](#_Toc10647417)

[3. 对接信息 4](#_Toc10647418)

[4. 测试单处理 4](#_Toc10647419)

[三、 业务及系统接口对应说明 5](#_Toc10647420)

[四、 Sign获取及其刷新 5](#_Toc10647421)

[五、 接口说明 8](#_Toc10647422)

[1. API请求拼装规则 8](#_Toc10647423)

[2. 系统参数 9](#_Toc10647424)

[3. 系统请求全局响应公共部分 9](#_Toc10647425)

[六、 接口明细 10](#_Toc10647426)

[1. 首页及其他页面各个位置的推荐广告位 10](#_Toc10647427)

[1.1 获取广告图/推荐图/轮播图 10](#_Toc10647428)

[1.2 获取推荐房型 11](#_Toc10647429)

[2. 房源列表及详情 12](#_Toc10647430)

[2.1 获取搜索结果页房源列表(分页) 12](#_Toc10647431)

[2.2 房源详情页数据获取 12](#_Toc10647432)

[2.3 房东信息 15](#_Toc10647433)

[2.4 获取评论列表按照 房东/房源 15](#_Toc10647434)

[2.5 获取评论列表按照 评论用户 16](#_Toc10647435)

[2.6 用户进行评论 17](#_Toc10647436)

[2.7 获取一个房型的房源日历 18](#_Toc10647437)

[2.8 获取默认选择的日历和费用明细 19](#_Toc10647438)

[2.9 获取用户选择的入离时间和费用明细 20](#_Toc10647439)

[3 订单相关 21](#_Toc10647440)

[3.1 用户下单 21](#_Toc10647441)

[3.2 获取订单列表(按 状态, 用户, 房东 组合获取) 22](#_Toc10647442)

[3.3 获取单个订单信息(按订单号ordercode获取) 23](#_Toc10647443)

[4. 用户相关 23](#_Toc10647444)

[4.1 获取OpenId, 通过js\_code 23](#_Toc10647445)

[4.2 登陆/注册时发送短信验证码 24](#_Toc10647446)

[4.3 登录/注册 (拿到短信验证码后) 25](#_Toc10647447)

[4.4 重新导入订单 (线上商户) 26](#_Toc10647448)

[4.5 获取在线下单的订单类型(已测) 27](#_Toc10647449)

[4.6 自动创建一个订单编号(已测) 27](#_Toc10647450)

[4.7 线上商户在线下单 28](#_Toc10647451)

[5. 订单公共相关接口 29](#_Toc10647452)

[5.1 获取 订单详情 29](#_Toc10647453)

[5.2 获取 单个订单的 操作日志 30](#_Toc10647454)

[5.3 修改收货信息 30](#_Toc10647455)

[5.4 修改物流信息 31](#_Toc10647456)

[5.5 获取 日期范围 简要列表 32](#_Toc10647457)

[5.6 获取 全部订单(管理员) 32](#_Toc10647458)

[6. 商品及供货相关接口 33](#_Toc10647459)

[6.1 获取 商品总库列表 33](#_Toc10647460)

[6.2 获取 线下商户供货商品列表 34](#_Toc10647461)

[6.3 线下商户 添加/移除供货 35](#_Toc10647462)

[6.4 线下商户 修改供货报价和库存 36](#_Toc10647463)

[7. 商户/员工 相关信息接口 37](#_Toc10647464)

[7.1 商户综合信息展示(已测) 37](#_Toc10647465)

[7.2 解绑账号/退出登陆(已测) 38](#_Toc10647466)

[7.3 获取用户相关设置 (已测) 38](#_Toc10647467)

[7.4 设置用户的配置项目(结合7.3使用) (已测) 39](#_Toc10647468)

[8. 数据统计相关接口 40](#_Toc10647469)

[8.1 数据罗盘(已测) 40](#_Toc10647470)

# 新手指南

## 简介

花啦民宿系统(即乐住坊)是一个民宿预定平台, 类似于途家,爱彼迎等, 地位于各个景区周边的民宿预定,将会首先给出 用户端1.0微信小程序版本, 后续将增加 房东管理端的小程序版本

## 名词解释

用户: 即C端用户,可进行在线房源的下单预定 (下单需登陆)

房东: 房源/房型的管理者/运营者

管理员: 即平台的管理方,可对房东进行管理, 可对房源进行审核

OpenId: 每个打开小程序或者公众号的微信用户 都有自己的openid, 用于平台识别是哪个微信用户(同一个微信用户在我方公众号, 花啦民宿小程序上的openid都不同)

sign: 用户令牌,每个微信用户的令牌都不同 且一个用户的令牌只在当天有效, 过期需要重新获取 (部分接口请求时是需要sign的, 有些则不需要)

# 对接前准备工作说明

## 准备工作

需要通过面对面或者远程视频的方式将大概流程和注意事项提前沟通一遍, 并将相关业务逻辑在pc端实际演示一遍

## 开始对接

前端按照接口进行对接开发,遇到问题两端联调

## 对接信息

|  |  |  |
| --- | --- | --- |
| **名称** | **数值** | **说明** |
| appfrom |  | 分配给应用的key，用于标记一个唯一的接入方（对接时给出） |
| sign |  | 用户颁发的动态令牌（会专门提供对应接口来获取） |
|  |  |  |
| 接口域名 |  | 开发前管理员会给出 |

**注：**

## 测试单处理

由于目前相关业务都还没有上线, 所以在小程序正式上线运行前所有的订单都默认为测试订单

# 业务及系统接口对应说明

1. **订单流转流程**

**正常流程** : 用户浏览下单->付款->房东确认->待入住->已入住->已离店

异常流程1: 下单->付款->在确认前后的取消->退款

异常流程其他: 具体参看订单状态来确定

1. **接口调用流程**
2. 的
3. 对于不需要权限校验(sign)的接口直接传参调用即可
4. 对于需要权限校验(sign)的接口需要首先获取到openid,然后跳转到登陆页面需要用户手机号验证码进行登录
5. 用户登陆后可以获取sign
6. 成功绑定后,继续根据sign调用其他接口
7. 接下来需要用到上面的sign的,每次拿到的sign只在当天有效

# Sign获取及其刷新

**1.小程序通过js\_code 获取 openid (get请求方式)**

**请参考 接口明细中的4.1**

**2.拿到openid后可以和用户输入的手机号联合调用触发发送验证码**

**请参考 接口明细中的4.2**

**3.根据openid, 手机号, 验证码 可以获取用户信息 及sign**

**请参考 接口明细中的4.3 (需要说明的是返回的数据中UPwd就是sign,)**

**4.默认其他接口都按如下格式拼接**

Get请求的接口格式:

[**https://接口域名/api/hotel/**](https://接口域名/api/hotel/)**请求路径?参数拼接**

Post请求路径

[**https://接口域名/api/hotel/**](https://接口域名/api/hotel/)**请求路径**

**实际对接中红字替换为 自己微信的openid 和 管理员给您分配的appfrom**

**已成功绑定后台账号的微信用户可以通过此接口获取到对应的账户信息和sign,**

**sign在返回的实体信息UPwd字段中,sign的过期时间在返回实体信息UpdateTime字段中**

**sign获取频次限制: 该接口目前设置的调用频次为每个openid , 12次/天, 超频调用可能会触发风控引发被封号, 需要开发者将获取到的openid在本地缓存**

* **返回结果示例**

{

"respCode": "**2000**",

"respMsg": "获取商户信息成功",

"respData": {

"Id": 3,

"UName": "王集权",

"UPhone": "15313815000",

"UMail": "15313815000@test.com",

"**UPwd**": "FDVIm0oOentNTUnC5U/155+USPZmMJ0M",

"UImg": "",

"UInfo": "xx1测试商户,勿接单",

"AParentId": 0,

"AccRoleId": 7,

"AStatus": 2,

"CreateTime": "2018-10-20 20:11:50",

"SignTime": "2019-02-21 00:48:45",

"SignIp": "111.14.85.27",

"SignCount": 10,

"WXOpenId": "",

"WXName": "",

"Remark": "",

"**UpdateTime**": "2019-02-23 00:00:00",

"IsDelete": 0,

"RoleName": "线下商户 外部",

"RoleGroup": 3,

"RoleGrade": 2,

"RoleDes": "线下商户组 中级 ",

"RStatus": 1

},

"respDataExt": "",

"respTime": "2019-02-22 15:32:35",

"respVerify": "fcd100597e89f74a0f80062939930205"

}

**没有绑定时返回的响应结果**

{

"respCode": "**1001**",

"respMsg": "未找到绑定的账号或账号已关闭,请联系管理员处理",

"respData": "",

"respDataExt": "",

"respTime": "2019-02-22 15:37:24",

"respVerify": "9d55a3b1087a701e492564ad5d630765"

}

**如果没有返回对应的响应结果,请自动跳转到绑定页面(如果已在绑定页面了就不用跳转了),要求用户输入用户名和密码进行绑定, 绑定时候请调用如下接口**

https://接口域名/api/main/account/BindOpenIdToAcc

{

"openid": "自己的openid",

"wximg": "微信头像地址",

"wxname": "微信昵称",

"uname": "商户自己的系统登录名",

"pwd": "商户自己的密码",

"uip": "当前微信用户的ip或者经纬度"

}

请求方式为post请求

* **返回结果示例**

{

"respCode": "**2000**",

"respMsg": "绑定成功",

"respData": "",

"respDataExt": "",

"respTime": "2019-02-22 15:37:24",

"respVerify": "9d55a3b1087a701e492564ad5d630765"

}

**或**

{

"respCode": "**1001**",

"respMsg": "错误或者警告信息",

"respData": "",

"respDataExt": "",

"respTime": "2019-02-22 15:37:24",

"respVerify": "9d55a3b1087a701e492564ad5d630765"

}

# 接口说明

## API请求拼装规则

所有请求地址, 都是由于 接口域名前缀+接口路径 组成的, 请求方式有GET 和 POST两种

通常情况下, 接口路径中 **最后一个单词以get开头的话**, 则此接口是已get方式进行请求的

请求举例:

$.ajax({

type: "GET",

url: "请求地址,

data: {

parm1: $("#parm1").val(),

parm2: $("#parm2").val()

},

contentType: "application/text;charset=utf-8",

beforeSend: function(request) {

request.setRequestHeader("appfrom", $myappfrom );

request.setRequestHeader("sign", $mysign );

},

success: function(data) {

//bind data or alert error

}

});

接口路径中 **最后一个单词不是以get开头的话** 则此接口默认是已post方式进行请求的

请求举例:

$.ajax({

type: "POST",

url: "请求地址,

data: {

parm1: $("#parm1").val(),

parm2: $("#parm2").val()

},

contentType: "application/json;charset=utf-8",

beforeSend: function(request) {

request.setRequestHeader("appfrom", $myappfrom );

request.setRequestHeader("sign", $mysign );

},

success: function(data) {

//bind data or alert error

}

});

## 系统参数

**除非特殊说明,否则所有的请求中都须在 header中加入{ appfrom: $myAppFrom }**

**除非特殊说明,所有需要sign的接口中,都须在header中加入{ appfrom: $myAppFrom , sign: $MySign }**

| **名称** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| appfrom | String | 是 | 对接口管理员给到的appfrom, 可以写死到小程序配置文件中 |
| sign | String | 是 | 通过接口返回的sign,注意他的过期时间,每个微信用户的sign都不同 |

## 系统请求全局响应公共部分

**暂时全局响应码给出简单几种, 可以配合提示信息进行识别**

|  |  |  |
| --- | --- | --- |
| **响应码** | **响应描述** | **解决方案** |
| 2000 | 请求得到成功执行,无错误 | 此时可以根据具体接口查看respData是否有数据需要绑定 |
| 1000 | 非法请求 | 遇到此种情况可能是原来的sign过期了,需要重新获取sign |
| 1001 | 遇到错误或警告 | 此时可以根据respMsg判断错误或警告信息,亦可弹出信息 |

**全局响应参数通常为如下,如有特殊会在具体接口上说明**

|  |  |  |
| --- | --- | --- |
| **参数** | **参数描述** | **补充说明** |
| respCode | 请求响应码 |  |
| respMsg | 响应说明,对响应码的说明 | 如果响应respCode不是2000,则可以直接吐司弹出此信息 |
| respData | 响应内容 | 请求成功时会有具体的返回内容 |
| respDataExt | 响应内部的补充和扩展内容 | 对响应内容的补充说明 |
| respVerify | 响应校验值 | 主要用户校验响应的真实性, 非后端请求请忽略此值 |
| respTime | 响应时间 |  |

* **响应示例**

{

"respCode": "**1001**",

"": "错误或者警告信息",

"respData": "",

"respDataExt": "",

"respTime": "2019-02-22 15:37:24",

"respVerify": "9d55a3b1087a701e492564ad5d630765"

}

# 接口明细

## 首页及其他页面各个位置的推荐广告位

### 1.1 获取广告图/推荐图/轮播图

* **API****路径：**Recommend/GetAdList
* **API概述:** 获取各个位置的推荐图,例如首页 轮播图和底部推荐图
* **API建议: 用户切换城市的时候需要刷新页面**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| dicTypeId | Int | Y | 1. 小程序首页顶部轮播推荐 2. 小程序首页底部广告位推荐 |
| cityName | String | N | 城市名称, 默认空字符串 |

* **返回结果中的respData**

请参考数据实体表中的 **RecommendAD**实体, 注意返回的是**实体集合**

### 1.2 获取推荐房型

* **API路径：**Recommend/GetRoomList
* **API概述:** 本接口主要是用于查询各个页面不同位置的推荐房源列表
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| dicTypeId | Int | Y | 3. 首页精选推荐房源 4. 首页热门入住 |
| cityName | String | N | 城市名称, 默认空字符串 |

* **返回结果中的respData**
* 请参考数据实体表中的 **View\_RoomMainInfo**实体, 注意返回的是**实体集合**

## 房源列表及详情

### 2.1 获取搜索结果页房源列表(分页)

* **API路径：**ViewRoomMainInfo/GetListPageSearch
* **API概述:** 本接口主要获取房源搜索结果列表
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| orderBy | Int |  | 先默认写个1 |
| searchName | String | Y | 查询关键词, 默认传空字符串即可 |
| pageSize | Int | Y | 每页展示几条信息 默认写15 |
| pageIndex | Int | Y | 查询第几页 默认写1 |
| accountId | Int |  | 房东id, 默认写0 |
| stayInDate | String |  | 入住日期, 格式为 yyyy-MM-dd |
| stayOutDate | String |  | 离店日期, 格式为 yyyy-MM-dd |

* **返回结果中的respData**

请参考数据实体表中的 **MsgNotice** 实体, 注意返回的是**实体集合**

### 2.2 房源详情页数据获取

* **API路径：**/Room/GetRoomForDetailPage
* **API概述:** 本接口主要是用于房型详情页的数据绑定
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| roomAccId | int | Y | 从上个页面传过来的RoomAccId |

* **返回结果中的respData, 请参考下图**



### 2.3 房东信息

* **API路径：**AccInfo/GetModelByAccId
* **API概述:** 用于各个位置专门获取房东信息的
* **API建议: 例如 房东详情页**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| accountid | String | Y |  |
| from | int | Y | 请求位置 : 1.从房东详情页 |

* **返回结果中的RespData**

请参考数据实体表中的 **AccInfo**实体, 注意返回的是**单个实体**

### 2.4 获取评论列表按照 房东/房源

* **API路径：**RoomEvaluate/GetListPageByAccOrRoom
* **API概述:** 本接口主要是按房东或者房源获取对应的评论信息
* **API建议:**
* **请求方式：GET**
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| accountid | Int | Y | 房东的id, 默认传0 |
| roomAccId | int |  | 房型/房源的id, 默认传0 |
| pageSize | Int | Y | 每页展示几条信息 默认写15 |
| pageIndex | Int | Y | 查询第几页 默认写1 |

* **返回结果中的RespCode**

请参考数据实体表中的 **RoomEvaluate**实体, 注意返回的是**实体集合**

* **返回结果中的RespDataExt**

totalCount: 总结过有几条, currCount:当前返回信息条数 ; totalPage:共计多少页 , currPage: 当前是第几页

### 2.5 获取评论列表按照 评论用户

* **API路径：**RoomEvaluate/GetListPageByUser
* **API概述:** 本接口主要是用户获取对应的评论信息
* **API建议:**
* **请求方式：GET**
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| userId | Int | Y | 用户的id, 登陆后可以获取到 |
| pageSize | Int | Y | 每页展示几条信息 默认写15 |
| pageIndex | Int | Y | 查询第几页 默认写1 |

* **返回结果中的RespCode**

请参考数据实体表中的 **RoomEvaluate**实体, 注意返回的是**实体集合**

* **返回结果中的RespDataExt**

totalCount: 总结过有几条, currCount:当前返回信息条数 ; totalPage:共计多少页 , currPage: 当前是第几页

### 用户进行评论

* **API路径：**RoomEvaluate/AddModelEvaluate
* **API概述:** 本接口主要是用于用户入住/离店后对房源的评价
* **API建议: 需要用户登陆后才可以的呀(即header里需要有sign )**
* **请求方式：**POST
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| userId | Int | Y | 用户id |
| ordercode | String | Y | 订单编号 |
| msg | String | Y | 评论内容, 500字以内 |
| score | Decimal(18,1) | Y | 评分 1到5分, 例如 4.8 |
| imgA | String | Y | 评论图片A , 选填 |
| imgB |  |  | 评论图片 选填 |
| imgC |  |  | 评论图片 选填 |
| imgD |  |  | 评论图片 选填 |
| imgE |  |  | 评论图片 选填 |
|  |  |  |  |

* **返回结果中的RespCode**

操作成功时RespCode=2000, RespMsg=”评论成功”

操作失败时RespCode=1001, RespMsg=”错误或警告信息”

### 2.7 获取一个房型的房源日历

* **API路径：**Room/GetRoomDayForUser
* **API概述:** 本接口主要是用户获取一个房型的房源日历
* **API建议:**
* **请求方式：GET**
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| roomAccId | Int | Y |  |

* **返回结果中的RespCode**

请参考数据实体表中的 **自定义房源日历** 实体, 注意返回的是**实体集合** ,以下是解释

|  |  |  |
| --- | --- | --- |
| **属性** | **类型** | **描述** |
| RoomDayId | Int | 房源日历表的id,可以用来计算用户选择入离时间后 住几晚 |
| DayOf | String | 一个月里的几号,用于展示的 |
| DayDes | String | 副标题描述, 例如儿童节, 满房 |
| WeekOf | Int | 星期几, 0周日, 1周一 2周二…….以此类推 |
| CanSelect | Bool | 是否允许用户选择 |
| RemainingRoom | Int | 剩余几间房 |
| DayPrice | Decimal | 当日房价 |

### 2.8 获取默认选择的日历和费用明细

* **API路径：**Room/GetRoomDayDefaultSel
* **API概述:** 本接口主要是用户进入 搜索列表或者提交订单页面时默认选中的入离日期
* **API建议:**
* **请求方式：GET**
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| roomAccId | Int | Y |  |

* **返回结果中的RespCode**

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **描述** |
| totalFee | Decimal | 总费用 (押金+住宿费) |
| totalStayFee | Decimal | 住宿总费用 |
| onlineDeposit | Decimal | 在线支付的押金 |
| stayList | 字典 | 每晚的住宿费用列表 |
| SelDay | roomDay实体列表 | 个实体, 第一个实体是入住时间的实体, 第二个是离店日期的实体 |
| orderMaxRoomCount | Int | 所选时间段内可选的最大房间套数 |
| orderMaxUserCount | Int | 所选时间段内可选的最多入住人数 |
|  |  |  |

### 2.9 获取用户选择的入离时间和费用明细

* **API路径：**Room/GetRoomDayUserSel
* **API概述:** 本接口主要是用户进入 搜索列表或者提交订单页面时默认选中的入离日期
* **API建议:**
* **请求方式：GET**
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| startRoomDayId | Int | Y | 入住时间的roomday的id |
| endRoomDayId | Int | Y | 离店时间的roomday的id |

* **返回结果中的RespData**

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **描述** |
| totalFee | Decimal | 总费用 (押金+住宿费) |
| totalStayFee | Decimal | 住宿总费用 |
| onlineDeposit | Decimal | 在线支付的押金 |
| stayList | 字典 | 每晚的住宿费用列表 |
| SelDay | roomDay实体列表 | 个实体, 第一个实体是入住时间的实体, 第二个是离店日期的实体 |
| orderMaxRoomCount | Int | 所选时间段内可选的最大房间套数 |
| orderMaxUserCount | Int | 所选时间段内可选的最多入住人数 |
| roomAccId | Int |  |
| orderMaxRoomCount | Int | 在此时间段内最多可以预定多少套所选房型的房间 |
| orderMaxUserCount | Int | 在此时间段内所选房型最多可以入住多少人 |
| stayDes | Int | 入离描述, 格式例如 06-28到06-29 |
| stayInTime | Datetime | 入住时间 |
| stayOutTime | Datetime | 离店时间 |
| nightCount | Int | 共几晚 |

## 订单相关

### 用户下单

* **API路径：**OrderMain/ AddOrderHotel
* **API概述:** 本接口主要是用于用户下单预订房间
* **API建议:**
* **请求方式：**POST
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| userId | Int | Y |  |
| startRoomDayId | Int | Y | 用户选择的入住时间的 roomday的id |
| endRoomDayId | Int | Y | 用户选择的离店时间的 roomday的id |
| usermsg | String | Y | 用户留言, 下单备注 |
| userName | String | Y | 入住用户姓名 |
| userPhone | String | Y | 用户联系电话 |
| userIdCard | String |  | 入住人员身份证, 选填 |
| roomCount | Int | Y | 预订房间套数 |
| personCount | Int | Y | 入住人数量 |

* **返回结果中的RespCode**

操作成功时RespCode=2000, RespMsg=”下单成功”

操作失败时RespCode=1001, RespMsg=”错误或警告信息”

### 获取订单列表(按 状态, 用户, 房东 组合获取)

* **API路径：**OrderMain/GetListPageByUserOrAcc
* **API概述:** 本接口主要是用于获取用户或者房东对应的订单
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| stateId | Int |  | 订单状态id, 从订单状态表获取 |
| userId | Int |  | Userid和accountid至少传其中一项 |
| accountId | Int |  | Userid和accountid至少传其中一项 |
| pageSize | Int | Y | 每页展示几条信息 默认写15 |
| pageIndex | Int | Y | 查询第几页 默认写1 |

* **返回结果中的RespData**

请参考数据实体表中的 **OrderMain**实体, 注意返回的是**实体集合**

* **返回结果中的RespDataExt**

totalCount: 总结过有几条, currCount:当前返回信息条数 ; totalPage:共计多少页 , currPage: 当前是第几页

### 获取单个订单信息(按订单号ordercode获取)

* **API路径：**OrderMain/GetOrderByCode
* **API概述:**
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| ordercode | Int |  | 订单编号 |

* **返回结果中的RespData**

请参考数据实体表中的 **OrderMain**实体, 注意返回的是**单个实体**

## 用户相关

### 获取OpenId, 通过js\_code

* **API路径：**https://接口域名/api/weixin/WXAppAuth/GetOpenIdByJSCode
* **API概述:** 本接口主要是用于小程序获取openid
* **API建议: 本接口的地址可能和其他的地址有轻微的区别, 注意分辨**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| js\_code | String | Y | 小程序中获取到的js\_code |
| From | Int | Y | 默认传2即可 (表示来自民宿小程序) |

* **返回结果中的RespData**

请自行分析返回的结果进行使用

**2.拿到openid后可以和用户输入的手机号联合调用触发发送验证码**

https://接口域名/api/hotel/userwx/GetVerifyCode?openId=YourOpenId&phone=InputPhone

### 登陆/注册时发送短信验证码

* **API路径：**UserWX/RegistOrLoginUser
* **API概述:** 本接口主要是用于在用户登陆时输入手机号后发送短信验证码
* **API建议: 1.暂时先不加如图形校验码; 2.返回响应后开始60秒倒计时**
* **请求方式：**POST
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| phone | String | Y | 手机号码 |
| openid | String | Y | 通过接口4.1能获取到 |
| checkcode | String |  | 图形校验码, 暂时就传一个空字符串就行 |
| from | String | Y | 来源, 1.来自民宿小程序 //2.来自微店网页端 |

* **返回结果**

操作成功时RespCode=2000, RespMsg=” 验证码已下发” ,

操作失败时RespCode=1001, RespMsg=”提示信息,可以toast出来” ,

### 登录/注册 (拿到短信验证码后)

* **API路径：**UserWX/ RegistOrLoginUser
* **API概述:** 获取到短信验证码后输入并点击登录
* **API建议: 1.返回的数据中UPwd即为sign, 请存储, 部分接口需要在 header中携带sign**

**2.拿到响应的实体后,将实体信息中UPwd也就是sign和 Id存储到本地**

**3.sign的过期时间就是实体信息中的UpdateTime**

* **请求方式：**POST
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| phone | String | Y | 手机号码 |
| openid | String | Y | 通过接口4.1能获取到 |
| verifycode | String | Y | 短信验证码 |
| from | String | Y | 来源, 1.来自民宿小程序 //2.来自微店网页端 |
| nickname | String | Y | 微信昵称 |
| headimg | String | Y | 用户头像链接 |

* **返回结果**

操作成功时RespCode=2000, RespMsg=” 提示信息,可以toast出来” , RespData=UserWX实体(**请参考实体信息表中的UserWX**)

操作失败时RespCode=1001, RespMsg=”提示信息,可以toast出来” ,

### 已登陆用户sign过期重新获取

* **API路径：**UserWX/GetUserSign
* **API概述:** 重新获取用户信息及其sign
* **API建议:**
* **请求方式：**GET
* **业务参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| uPhone | String | Y | 手机号码 |
| openId | String | Y |  |

* **返回结果**

请参考数据实体表中的 **UserWX**实体, 注意返回的是**单个实体**

**1.返回的数据中UPwd即为sign, 请存储, 部分接口需要在 header中携带sign**

**2.拿到响应的实体后,将实体信息存储到本地**

**3.sign的过期时间就是实体信息中的UpdateTime**