# 接口文档

## 登陆

|  |  |
| --- | --- |
| 接口说明 |  |
| 请求地址 | /admin/wechat\_login |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/wechat\_login',  data: {  code  } |
| 参数说明 | Code微信登陆返回的code值 |
| 返回说明 |  |
| 成功返回 | 首次登陆：  {  status: 201,  message: '登录成功'  }  登陆成功：  {  status: 200,  message: '登录成功'  }  登陆失败：  {  status: -1,  message: '登录失败'  } |

## 设置用户信息

|  |  |
| --- | --- |
| 接口说明 | wx.getUserInfo的返回值userInfo保存起来，需要一个user\_id, |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/user\_info',  data,  method: 'post'  } |
| 参数说明 | Data是wx.getUserInfo的返回值res.userInfo,包含avatarUrl，  nickName，  city,  gender(性别 0：未知、1：男、2：女),  province,  city,  country。  后台保留avatarUrl，nickName，city。保存用户信息的时候，要有一个user\_id, |
| 返回说明 |  |
| 成功返回 | 成功{  status: 200,  message: '设置用户信息成功'  }  失败{  status: -1,  message: '获取用户信息失败'  } |

## 添加收获地址地址

|  |  |
| --- | --- |
| 接口说明 | 用户管理自己的收获地址，添加一个收获地址，保存到数据库, |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/address',  method: 'post',  data: {  phone: data.phone,  name: data.name,  title: data.addressInfo.title,  address: data.addressInfo.address,  gender: data.gender,  house\_number: this.data.houseNumber,  lng: data.addressInfo.location && data.addressInfo.location.lng,  lat: data.addressInfo.location && data.addressInfo.location.lat,  }  } |
| 参数说明 | Title：地址名称，address：地址,需要 id主键，需要一个字段来表示是哪个用户的地址。user\_id |
| 返回说明 |  |
| 成功返回 | {  status: 200,  success: '添加收货地址成功'  }  填写的参数有问题：  {  status: '-1',  message: '添加地址失败，参数有误'  }  后端有问题{  status: -1,  type: 'ADD\_ADDRESS\_FAILED',  message: '添加收货地址失败'  } |

## 获取用户所有的收获地址

|  |  |
| --- | --- |
| 接口说明 | 用户在订单选择地址的时候获取用户的所有地址以供选择/用户在添加新的收获地址前获取所有的收获地址来显示 |
| 请求地址 |  |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/all\_address'  } |
| 参数说明 |  |
| 返回说明 | 成功获取的地址是address是一个地址list，包含  {  [id,  phone:  name: ,  title:,  address:,  gender:,  house\_number:,  lng: ,  lat:  ]，  ….  } |
| 成功返回 | {  status: 200,  address: address,  message: '获取地址成功'  }  {  status: -1,  message: '获取收货地址失败',  } |

## 删除某个收获地址

|  |  |
| --- | --- |
| 接口说明 | 用户管理地址，删除一个收获地址 |
| 请求地址 |  |
| 请求方式 | DELETE |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/address',  method: 'delete',  data: {  address\_id: this.data.addressList[index].id  }  } |
| 参数说明 | address\_id是某个地址的主键。 |
| 返回说明 |  |
| 成功返回 | {  status: 200,  message: '删除收获地址成功'  }  {  status: -1,  message: '删除地址失败，参数有误'  }  {  status: -1,  message: '删除收获地址失败',  } |

管理地址（待定）address\_manage页面

|  |  |
| --- | --- |
| 接口说明 |  |
| 请求地址 |  |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | url: /admin/all\_address/,  data: {} |
| 参数说明 |  |
| 返回说明 |  |
| 成功返回 |  |

获取用户的所有地址

|  |  |
| --- | --- |
| 接口说明 | 获取用户的全部收货地址，在订单上显示返回的地址列表的第一个 |
| 请求地址 |  |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: '/admin/all\_address',  } |
| 参数说明 |  |
| 返回说明 | 成功获取的地址是address是一个地址list，包含  {  [id,  phone:  name: ,  title:,  address:,  gender:,  house\_number:,  lng: ,  lat:  ]，  ….  } |
| 成功返回 | {  status: 200,  address: address,  message: '获取地址成功'  }  {  status: -1,  message: '获取收货地址失败',  } |

提交订单

|  |  |
| --- | --- |
| 接口说明 | 用户选好菜之后，提交订单保存 |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/wxorder',  method: 'post',  data: {  restaurant\_id: data.restaurantInfo.id,  foods: JSON.stringify(data.foodsList),  address\_id: data.addressInfo.id  } |
| 参数说明 | restaurant\_id：店铺id,  foods:购买的菜品的列表的JSON 字符串,字段有  {  [  price,菜品价格  name,菜品名称  pic\_url,菜品图片地址  num,菜品数量  ]，  …..}  address\_id:用户所选择的地址id |
| 返回说明 | order\_id：订单id, total\_price：总价 |
| 成功返回 | {  status: 200,  order\_id: values[3],  total\_price: values[0].total\_price.toFixed(2),  }  {  status: -1,  message: '下订单失败，参数有误'  }  {  status: -1,  message: '提交订单失败'  } |

更新token

|  |  |
| --- | --- |
| 接口说明 | 微信需要更新token,token有效日期为2小时 |
| 请求地址 |  |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: '/service/uploadtoken'  } |
| 参数说明 |  |
| 返回说明 | Uptoken: AppID和AppSecret调用接口来获取access\_token |
| 成功返回 | {  uptoken: token,  message: '获取上传凭证成功',  status: 200  }  {  status: -1,  message: '获取上传凭证失败'  } |

查询店铺信息

|  |  |
| --- | --- |
| 接口说明 | 根据店铺id查询店铺信息 |
| 请求地址 |  |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/order/' + id  } |
| 参数说明 | id :restaurant\_id |
| 返回说明 | data:查询的店铺信息。里面的数据为是一个restaurantInfo，restaurantInfo里的数据是  {  pic\_url,商店的图片地址  name 商店名称  } |
| 成功返回 | {  status: 200,  data: order,  message: '获取指定订单成功'  }  {  status: -1,  message: '获取指定订单失败，参数有误!'  }  {  status: -1,  message: '该订单不存在'  }  {  status: -1,  message: '获取指定订单失败'  } |

提交评论

|  |  |
| --- | --- |
| 接口说明 | 订单结束填写评论提交 |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/comment',  method: 'post',  data: {  order\_id: data.order\_id,  comment\_data: data.commentData,  food\_score: this.data.food\_score,  pic\_url: JSON.stringify(data.images)  } |
| 参数说明 | order\_id：订单id  comment\_data:评论内容  food\_score：菜品评分（未确定，应该是小于5的float）  pic\_url：评论的图片json字符串:采用的是七牛云存储，所以是七牛云存储后的图片地址 |
| 返回说明 |  |
| 成功返回 | {  status: 200,  message: '评论成功'  }  {  status: -1,  message: '评价失败，该订单不能评论!'  }  {  status: -1,  message: '评论失败，参数有误'  } |

## 1、根据lat，lng获取详情位置

|  |  |
| --- | --- |
| 接口说明 | 根据lat，lng获取详情位置 |
| 请求地址 | /v1/detailLocation |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/detailLocation',  data: {  location,  } |
| 参数说明 | Location：  {  lat,  lng  }  Lat：维度；lng:经度 |
| 返回说明 | 根据经纬度获取详细地址信息  message:  {  address,  location  }  其中  address：获取到的地址，String  location：  {  lat,  lng  } |
| 成功返回 | { status: 200,  message: "获取位置信息成功",  data }  后端返回的，其中data为返回说明中的address和location |

## 2、获取多家餐馆

|  |  |
| --- | --- |
| 接口说明 | 获取多家餐馆 |
| 请求地址 | /v1/restaurants |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | data: {  lng: lng || this.data.location.lng,  lat: lat || this.data.location.lat,  offset: (this.data.pageIndex - 1) \* this.data.pageSize,  limit: this.data.pageSize  } |
| 参数说明 | Lat：维度；lng:经度 |
| 返回说明 | 返回一个对象List吧  Message:  {  {  id: Number,  user\_id:Number, //餐馆所属用户id  name: String,  pic\_url: String, //店家图片地址  month\_sales:Number, //月售几笔  month\_sales\_tip: String, //月售几笔  wm\_poi\_score: Number, //商家评分  delivery\_score:Number, //配送评分  quality\_score:Number, //质量评论  food\_score:Number, //口味评分  pack\_score:Number, //包装评分  distance: String, //距离  delivery\_time\_tip: String, //送达时间提示  shipping\_fee\_tip: String, //配送费提示  min\_price\_tip: String, //最低价起送提示  average\_price\_tip: String, //平均费  third\_category: String, //中式简餐  discounts2: [], //折扣  shipping\_time: String, //经营时间  shopping\_time\_start:String,//开始经营时间  shopping\_time\_end:String, //结束经营时间  shipping\_fee: Number, //配送费  avg\_delivery\_time: Number, //平均送达时间  min\_price: Number, //最低价起送价格  bulletin: String, //公告  app\_delivery\_tip: String, //配送信息提示  poi\_back\_pic\_url: String, //头部背景虚拟图  log\_field: {  poi\_type\_icon\_type: Number,  recommend\_type: Number,  average\_delivery\_time: Number,  search\_log\_id: String  },  comment\_number: Number, //评价数量  address:String, //地址  call\_center:String, //电话  lng:String, //纬度  lat:String, //经度  created\_at:{type:Date,default:new Date()}  }，  {}  …  } |
| 成功返回 | status: 200,  message: '获取餐馆列表成功',  data: restaurants |

3、关键字搜索地方

|  |  |
| --- | --- |
| 接口说明 | 关键字搜索地方，搜索一个地方，地址位置搜索，然后系统出周边的一些地址 |
| 请求地址 | /v1/suggestion |
| 请求方式 | GET |
| 前置条件 |  |
| 参数示例 | {  url: baseURL + '/v1/suggestion',  data: {keyword: this.data.keyword},  success: function (res) {  console.log('address-res', res.data.data.data);  vm.setData({  addressList: res.data.data.data  })  }  } |
| 参数说明 | Keyword：位置关键字 |
| 返回说明 | data:建议位置列表  {  [  title，地点名称，String  address 地点的地址，String  ],  ………  } |
| 成功返回 | { status: 200,  message: "获取位置信息成功",  data } |

4、

# 接口文档

## 获取用户所有的订单

|  |  |
| --- | --- |
| 接口说明 | 获取订单列表 |
| 请求地址 | /v1/orders |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/orders'  } |
| 参数说明 |  |
| 返回说明 |  |
| 成功返回 | {  status: 200,  data: [{  restaurant:{  pic\_url:”店铺图片地址”,  id:”店铺ID”,  name:”店铺名字”  },  id:”订单详情ID”,  food:[  {  name:”食物名字”,  num:”食物数量”  },{},…  ],  total\_price:”总价格”,  has\_comment:”是否有评价”  },{},…  ]  message: '获取我的订单列表成功'  } |

## 查询一个订单信息

|  |  |
| --- | --- |
| 接口说明 | 根据order\_id获取指定订单 |
| 请求地址 | '/v1/order/'+ id |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/order/'+ id  } |
| 参数说明 |  |
| 返回说明 | data里面的数据  {  restaurantInfo:{  id:店铺ID”，  pic\_url:”店铺图片地址”,  name:”店铺名字”  },  foodlist:{  pic\_url:”食物图片地址”,  name:”食物名字”,  price:”食物价格”,  num:”食物数量”  },  address:{  name:”收货人员”,  phone:”收货号码”,  address:”配送地址”  },  orderInfo:{  id:”订单号码”,  create\_time:”订单时间”,  total\_price:”订单总价”  }  } |
| 成功返回 | {  status: 200,  data:  message: '获取指定订单成功'  } |

## 根据关键字搜索商店。

|  |  |
| --- | --- |
| 接口说明 | 搜索商家 |
| 请求地址 | baseURL + '/v1/search/restaurant' |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/search/restaurant',  data: {  keyword: value  } |
| 参数说明 |  |
| 返回说明 | Data是个列表，表示搜索到的商家信息。  【{  id:”商家ID”,  pic\_url:”商家店铺图片地址”,  name:”商家名字”,  delivery\_time\_tip:”预计送达时间”  }，  …  】, |
| 成功返回 | {  status: 200,  data:  message: '搜索餐馆成功'  } |

## 获取指定餐馆信息

|  |  |
| --- | --- |
| 接口说明 | 根据restaurant\_id获取指定餐馆信息 |
| 请求地址 | '/v1/restaurant/' + id |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/restaurant/' + id,  data: {}  } |
| 参数说明 |  |
| 返回说明 | Data存有商家信息  restaurantInfo:{  pic\_url:”店铺图片地址”,  min\_price\_tip:”XXX”,  shipping\_fee\_tip:”运费”，  delivery\_time\_tip:”预计送达时间”，  bulletin:”XXX”,  wm\_poi\_score:”XXX”,  food\_score:”口味评分”,  pack\_score:”包装评分”，  delivery\_score:”配送评分”，  address:”商检地址”,  shopping\_time\_start:”配送开始时间”,  shopping\_time\_end:”配送结束时间”  } |
| 成功返回 | {  status: 1,  message: '获取指定餐馆信息成功',  data: |

## 获取某个餐馆的食物列表)

|  |  |
| --- | --- |
| 接口说明 | 根据restaurant\_id获取指定餐馆食物列表 |
| 请求地址 | '/v1/food/' + id |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/food/' + id,  data: {},  } |
| 参数说明 | Id是restaurant\_id |
| 返回说明 | data是一个列表，表示该商家的所有菜 [  {  name:”分类名字”，  spus:{  pic\_url:”XXX”,  name:”XXX”,  month\_saled:”月售”,  price:”价格”  }  },{},…  ]  } |
| 成功返回 | {  status: 200,  message: '获取食物列表成功',  data:  } |

## 获取某个餐馆的所有评论

|  |  |
| --- | --- |
| 接口说明 | 根据restaurant\_id获取餐馆评论 |
| 请求地址 | '/v1/comment/' |
| 请求方式 | get |
| 前置条件 |  |
| 参数示例 | {  url: '/v1/comment/',  data: {  restaurant\_id: id,  offset: 0,  limit: 200  }  } |
| 参数说明 |  |
| 返回说明 | Data是一个评论列表  数据如下  [  {  avatar:”XXX”,  user\_name:”用户名字”,  comment\_time:”评论时间”,  food\_score:”评分”,  comment\_data:”XXX”,  pic\_url:”评论图片地址”,  add\_comment\_list:[  {  content:”回复内容”  },{},…  ]  },{},…  ] |
| 成功返回 | {  status: 200,  message: '获取餐馆评论成功',  data:  } |

|  |  |
| --- | --- |
| 接口说明 |  |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 |  |
| 参数说明 |  |
| 返回说明 |  |
| 成功返回 |  |

|  |  |
| --- | --- |
| 接口说明 |  |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 |  |
| 参数说明 |  |
| 返回说明 |  |
| 成功返回 |  |

|  |  |
| --- | --- |
| 接口说明 |  |
| 请求地址 |  |
| 请求方式 | POST |
| 前置条件 |  |
| 参数示例 |  |
| 参数说明 |  |
| 返回说明 |  |
| 成功返回 |  |