

## 客户端 service 列表参考

数据返回格式 JSON:

**success:** true 则表示操作成功; false 表示操作失败。

**message:** 提示信息。当 success=false 时包括错误提示信息

**data:** 返回的具体数据

```
{
  "success": true,
  "message": "Get restaurants info of 1 success!",
  "data": {
    "restid": 1,
    "restname": "嘉苑饭店",
    "restphone": null,
    "restcity": 1,
    "restcityname": "北京",
    "restaddress": "海淀区大柳树路 6 号",
    "restwebsite": "http://www.baidu.com",
    "resttype": 3,
    "resttypeName": "炒菜",
    "restsearchtime": 0,
    "hot": 0,
    "longitude": 116.336172,
    "latitude": 39.953464
  }
}
```

下面说的返回数据是指 data 中的数据类型。

后面附相应的 Controller 的类图和 serviceImpl 的类图。

### 1. user 相关

#### 1) 用户登录

http://localhost:8080/mealplanner/app/login?loginName=minxin&password=12345

返回登录用户的基本信息

#### 2) 用户注册

http://localhost:8080/mealplanner/app/register?username=11&phonenum=11&email=11&password=11

返回注册用户的基本信息

#### 3) 获取用户详细信息

http://localhost:8080/mealplanner/app/userInfo?userId=1

### 2. 好友相关

#### 1) 获取 userId 所有好友信息

http://localhost:8080/mealplanner/app/friend/getAllFriends?userId=1

#### 2) 获取 userId 的好友请求列表

http://localhost:8080/mealplanner/app/friend/getFriendsRequest?userId=6

**3) userId 添加 friendId 为好友**

http://localhost:8080/mealplanner/app/friend/addFriendsRequest?userId=5&friendId=6

**4) userId 接受 friendId 的好友请求**

http://localhost:8080/mealplanner/app/friend/acceptFriendsRequest?userId=6&friendId=5

**5) userId 拒绝 friendId 的好友请求**

http://localhost:8080/mealplanner/app/friend/rejectFriendsRequest?userId=6&friendId=5

### 3. 餐厅及菜单信息

**1) 获取某个餐厅 restId 的基本信息**

http://localhost:8080/mealplanner/app/rest/getRest?restId=1

**2) 获取某个餐厅带菜单的信息**

http://localhost:8080/mealplanner/app/rest/getRestWithMenu?restId=1

**3) 获取所有的餐厅信息(带菜单信息)**

http://localhost:8080/mealplanner/app/rest/getAllRest

**4) 获取从第 start 个开始之后的 limit 个餐厅的信息(带菜单信息)**

http://localhost:8080/mealplanner/app/rest/getSeveralRestWithMenu?start=0&limit=1

**5) 获取从第 start 个开始之后的 limit 个餐厅的基本的基本信息**

http://localhost:8080/mealplanner/app/rest/getRestWithMenu?restId=1

**6) 通过餐厅名称精确匹配获取餐厅信息(带菜单信息)**

http://localhost:8080/mealplanner/app/rest/getRestWithMenuByName?restName=阿里食府

**7) 通过餐厅名称搜索获取餐厅列表**

http://localhost:8080/mealplanner/app/rest/searchByName?restName=阿里食府

**8) 通过餐厅的名称列表(用英文逗号分隔)获取餐厅基本信息(restId, restName, longitude, latitude)**

http://localhost:8080/mealplanner/app/rest/getRestInfoForMaps?restNames=阿里食府,西贝

**9) 根据地图获取的信息向数据库中插入相应的经纬度**

http://localhost:8080/mealplanner/app/rest/insertRestByMap?restName= 呷 哺 呷 哺  
&restAddress=四道口&restType=4&longitude=116.34615800000006&latitude=39.95815

### 4. 菜单相关

**1) 获取某个餐厅的菜单信息**

http://localhost:8080/mealplanner/app/menu/getMenuByRestId?restId=1

**2) 获取某种类型的菜单信息**

http://localhost:8080/mealplanner/app/menu/getMenuByFoodType?foodTypeId=1

**3) 获取所有的菜品类型**

http://localhost:8080/mealplanner/app/menu/getAllFoodType

### 5. 饭局相关

**1) userId 创建饭局**

http://localhost:8080/mealplanner/app/meal/createMeal?restId=1&datetime=2010-10-1&userId=1&friendIds=2,3

**2) userId 创建的所有饭局详细信息(包括好友的回复状态)**

注：饭局状态 all(-1,"所有"),ongoing(0,"正在进行"), success(1,"成功, 所有人已经同意"),

reject(2,"所有人已回馈，但是有人拒绝"),ordered(3,"已下订单啦"); 当 mealStatus 为 ordered 时，未同意/拒绝该请求的好友将无法进行操作且对其不可见。limit 默认值为 5。

http://localhost:8080/mealplanner/app/meal/getMealInfo?userId=1&start=0&limit=2

### 3) userId 创建的某种状态的饭局信息

注：饭局状态 all(-1,"所有"),ongoing(0,"正在进行"), success(1,"成功，所有人已经同意"), reject(2,"所有人已回馈，但是有人拒绝"),ordered(3,"已下订单啦"); limit 默认值为 5

http://localhost:8080/mealplanner/app/meal/getMealInfo?userId=1&start=0&limit=2&status=0

### 4) userId 接受到的饭局邀请

注：若饭局状态 mealStatus 不为 ordered，同时邀请的处理状态不为 waiting 等待处理,则加入返回队列。饭局邀请处理状态：status 的值 all(-1, "所有"), waiting(0,"等待处理中"),accept(1,"已接受"),reject(2,"已拒绝");

http://localhost:8080/mealplanner/app/meal/getMealRequest?userId=4

### 5) userId 所有状态下的饭局邀请

http://localhost:8080/mealplanner/app/meal/getMealRequest?userId=4

### 6) userId 接受饭局邀请 mealId

http://localhost:8080/mealplanner/app/meal/acceptMeal?userId=3&mealId=1

### 7) userId 拒绝饭局邀请 mealId

http://localhost:8080/mealplanner/app/meal/rejectMeal?userId=3&mealId=1

### 8) 获取饭局详细信息(包括好友的回复状态)

http://localhost:8080/mealplanner/app/meal/mealDetail?mealId=2

## 6. 订单相关

### 1) 获取某个人的所有订单

http://localhost:8080/mealplanner/app/order/getOrderByUser?userId=2

### 2) 获取某个人某种状态下的订单

订单状态：all(-1, "all"),commitSuccess(0, "commit success"), commitFailed(1, "commit failed"), confirmed(2, "restaurant confirmed"), canceled(3, "canceled"),finished(4, "finished");

http://localhost:8080/mealplanner/app/order/getOrderByUser?userId=2&status=0

### 3) 用户创建订单（若无 mealId 时为直接创建订单，否则为以 mealId 为基础创建的订单）

注：1. 根据时间,restId,人数获得 freeSeat 的 Id; 2.若没有空闲的 seat，则创建失败；若有空闲的 seat，则占用相应的 seat 同时创建相应的订单；3.若根据 mealId 创建订单，则更改 mealStatus 为 3 ordered.

http://localhost:8080/mealplanner/app/order/createOrder?userId=5&restId=1&mealId=1  
&date=2014-5-20 17:34:20&peopleNum=3&menuIds=1,2,3&phoneNum=12345

### 4) 用户取消订单

注：1. 若用户订单已经被餐厅确认，则用户无法取消订单，只能通过电话取消 2. 若用户可以取消订单，则更新订单状态为取消；同时释放当前座位

http://localhost:8080/mealplanner/app/order/cancelByUser?userId=3&orderId=2

## 7. 排队相关

### 1) 创建排队

注：1. 如果有空座且排队队列为空时，不进入排队程序；2.如果没有空座则进入排队流程：

a) 已在队列中，提示不可排队；b) 不在队列中，进行排队：生成对应的 seqNo、获取排在当前队列未用餐的第一个人的排队号(即下一个进入用餐状态的用户)、获取当前队列排队

的队数、向数据库中插入记录

<http://localhost:8080/mealplanner/app/seq/insertSeq?restId=3&userId=1&peopleNum=6>

## 2) 取消排队

<http://localhost:8080/mealplanner/app/seq/candle?userId=2>

## 3) 获取排队详情

<http://localhost:8080/mealplanner/app/seq/getSeqInfo?userId=1>

# 8. 推送相关

## 1) 手机应用用户账号绑定到百度账号

<http://localhost:8080/mealplanner/app/userBinding?userId=2&baiduUserId=924401985&channelId=4236885180925384783>

## 2) 消息提醒 Title

类别	Title	Description
饭局邀请	饭局邀请	userName + "喊你一起吃饭啦! 快去看看有什么惊喜吧~"
饭局邀请反馈	饭局邀请反馈	userName + "同意/拒绝了你的饭局邀请:) 快去看看这个饭局的进展情况吧!"
订单被餐厅确认	订单已确认	亲, 您的订单 + orderId + "已经被餐厅确认啦, 请不要忘记叫上小伙伴们来用餐哦~"
排队快到了的时候	用餐时间正在靠近	"亲您有 5 人就餐, 为您提供了" + seatType + "人桌, 前面还有 2 位排队" + seatType + "人桌, 请尽快回到餐厅, 以防错过排好"
排队之后开始用餐	用餐时刻	亲, 等了这么久, 终于可以吃上可口的饭菜啦, 还等神马!

class app

#### AppUserInfoController

- userService: UserService

+ login(loginName :String, password :String) : JsonResult<UserInfo>  
+ register(username :String, phonenum :String, email :String, password :String) : JsonResult<UserInfo>  
+ getUserInfo(userId :int) : JsonResult<UserInfo>  
+ userBinding(userId :int, baiduUserId :String, channelId :long) : JsonResult<UserInfo>

#### AppFriendController

- friendService: FriendService

+ getAllFriends(userId :int) : JsonResult<UserInfoWithFriends>  
+ getFriendsRequest(userId :int) : JsonResult<UserInfoWithFriends>  
+ addFriendsRequest(userId :int, friendId :int) : JsonResult<String>  
+ acceptFriendsRequest(userId :int, friendId :int) : JsonResult<String>  
+ rejectFriendsRequest(userId :int, friendId :int) : JsonResult<String>

#### AppRestaurantController

- restaurantService: RestaurantService

+ getAllRestaurants() : JsonResult<List<RestaurantWithMenu>>  
+ getSeveralRestaurantWithMenus(start :int, limit :int) : JsonResult<List<RestaurantWithMenu>>  
+ getRest(restId :int) : JsonResult<RestaurantInfo>  
+ getRestWithMenu(restId :int) : JsonResult<RestaurantWithMenu>  
+ getRestWithMenuByName(restName :String) : JsonResult<RestaurantWithMenu>  
+ searchByName(restName :String) : JsonResult<List<RestaurantInfo>>  
+ getRestInfoForMaps(restNames :String) : JsonResult<List<RestInfoForMap>>  
+ insertRestByMap(restName :String, restAddress :String, restType :int, longitude :double, latitude :double) : JsonResult<String>  
+ getSeveralRest(start :int, limit :int) : JsonResult<List<RestaurantInfo>>

#### AppMenuController

- menuService: MenuService

+ getMenuInfoByRestId(restId :int) : JsonResult<List<MenuInfo>>  
+ getMenuByFoodType(foodTypeId :int) : JsonResult<List<MenuInfo>>  
+ getAllFoodType() : JsonResult<List<FoodType>>

#### AppSequenceController

- sequenceService: SequenceService

+ insertSeq(userId :int, restId :int, peopleNum :int) : JsonResult<SequenceDetailForUser>  
+ cancel(userId :int) : JsonResult<String>  
+ getSeqInfo(userId :int) : JsonResult<SequenceDetailForUser>

#### AppMealController

- mealService: MealService

+ createMeal(restId :int, datetime :String, userId :int, friendIds :String) : JsonResult<String>  
+ getMealInfoByUserId(userId :int, status :int, start :int, limit :int) : JsonResult<List<MealWithDetail>>  
+ getMealRequest(userId :int) : JsonResult<List<MealRequestInfo>>  
+ acceptMeal(mealId :int, userId :int) : JsonResult<String>  
+ rejectMeal(mealId :int, userId :int) : JsonResult<String>  
+ mealDetail(mealId :int) : JsonResult<MealWithDetail>

#### AppOrderController

- orderService: OrderService

+ getOrderByUser(userId :int, status :int) : JsonResult<List<OrderDetail>>  
+ createOrder(userId :int, restId :int, mealId :int, date :String, peopleNum :int, menuIds :String, phoneNum :String) : JsonResult<OrderDetail>  
+ cancelByUser(orderId :int, userId :int) : JsonResult<String>

## class impl

## UserServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- userInfoMapper: UserInfoMapper
- userBindMapper: UserBindMapper

+ setUserInfoMapper(userInfoMapper :UserInfoMapper) : void
+ setUserBindMapper(userBindMapper :UserBindMapper) : void
+ getUserInfoById(id :int) : UserInfo
+ getUserInfoByLogin(loginName :String) : UserInfo
+ register(userInfo :UserInfo) : void
+ addUserChannel(userBind :UserBind) : void
```

## RestaurantServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- restaurantInfoMapper: RestaurantInfoMapper
- menuInfoMapper: MenuInfoMapper
- restUserMapper: RestUserMapper
- restCityMapper: RestCityMapper

+ getAllRestaurantWithMenus() : List<RestaurantWithMenu>
+ getSeveralRestaurantWithMenus(start :int, limit :int) : List<RestaurantWithMenu>
+ getSeveralRest(start :int, limit :int) : List<RestaurantInfo>
+ getRestaurantInfo(restId :int) : RestaurantInfo
+ getRestaurantInfoWithMenu(restId :int) : RestaurantWithMenu
+ getRestInfoByExactName(restName :String) : RestaurantInfo
+ getRestsByName(restName :String) : List<RestaurantInfo>
+ registRestaurant(restaurantInfo :RestaurantInfo) : int
+ mapRestaurantUser(restUser :RestUser) : int
+ getAllCity() : List<RestCity>
+ getRestNameByUser(userInfo :UserInfo) : RestaurantInfo
+ getRestIdByUserId(userId :int) : int
+ getRestInfoForMaps(restNames :String) : List<RestInfoForMap>
+ setRestCityMapper(restCityMapper :RestCityMapper) : void
+ setRestUserMapper(restUserMapper :RestUserMapper) : void
+ setRestaurantInfoMapper(restaurantInfoMapper :RestaurantInfoMapper) : void
+ setMenuInfoMapper(menuInfoMapper :MenuInfoMapper) : void
```

## SeatServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- RESERVED: int = 1
- OCCUPIED: int = 2
- seatInfoMapper: SeatInfoMapper
- seatStatusMapper: SeatStatusMapper

+ setSeatInfoMapper(seatInfoMapper :SeatInfoMapper) : void
+ setSeatStatusMapper(seatStatusMapper :SeatStatusMapper) : void
+ getAvailableSeats(restId :int, dateDay :String, dateClock :int, peopleNum :int) : List<SeatInfo>
+ reserveSeat(restId :int, dateDay :String, dateClock :int, peopleNum :int) : SeatInfo
+ occupySeat(seatId :int, restId :int, dateDay :String, dateClock :int) : void
+ getStateOfSeatWholeDay(seatId :int, restId :int, dateDay :String) : int[]
+ reserveSeatById(seatId :int, restId :int, dateDay :String, dateClock :int) : void
+ freeSeatById(seatId :int, restId :int, dateDay :String, dateClock :int) : void
+ getSeatsByRestId(restId :int) : List<SeatInfo>
+ addSeat(restId :int, seatNo :int, peopleNum :int, description :String) : int
+ updateSeat(seatId :int, restId :int, seatNo :int, peopleNum :int, description :String) : int
+ deleteSeat(seatId :int) : int
```

## MealServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- mealInfoMapper: MealInfoMapper
- mealFriendMapper: MealFriendMapper
- userInfoMapper: UserInfoMapper
- pushService: PushService

+ createMeal(mealInfo :MealInfo, friendIds :String) : int
+ getMealDetailByUserId(userId :int, status :int, start :int, limit :int) : List<MealWithDetail>
+ getMealRequest(userId :int) : List<MealRequestInfo>
+ handleAMeal(mealId :int, userId :int, status :int) : int
+ getMealDetail(mealId :int) : MealWithDetail
+ mealInfo2MealRequestInfo(mealInfo :MealInfo, userId :int, status :int) : MealRequestInfo
+ setMealInfoMapper(mealInfoMapper :MealInfoMapper) : void
+ setMealFriendMapper(mealFriendMapper :MealFriendMapper) : void
+ setUserInfoMapper(userInfoMapper :UserInfoMapper) : void
```

## FriendServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- friendInfoMapper: FriendInfoMapper
- userInfoMapper: UserInfoMapper

+ getFriendRequestsByStatus(friendId :int, status :int) : UserInfoWithFriends
+ getAllFriendInfos(userId :int) : UserInfoWithFriends
+ addFriendRequest(userId :int, friendId :int) : int
+ handleFriendRequest(friendId :int, userId :int, status :int) : int
+ setFriendInfoMapper(friendInfoMapper :FriendInfoMapper) : void
+ setUserInfoMapper(userInfoMapper :UserInfoMapper) : void
```

## MenuServiceImpl

```
- foodTypeMapper: FoodTypeMapper
- menuInfoMapper: MenuInfoMapper
- LOG: Log = LoggerFactory.getL... {readOnly}

+ getMenuInfoByRestId(restId :int) : List<MenuInfo>
+ getMenuInfoByType(typeId :int) : List<MenuInfo>
+ getMenuInfoDetail(menuId :int) : MenuInfo
+ addMenu(menuInfo :MenuInfo) : int
+ updateMenu(menuInfo :MenuInfo) : int
+ deleteMenu(menuId :int) : int
+ getAllFoodTypes() : List<FoodType>
+ addFoodType(foodType :FoodType) : int
+ deleteFoodType(foodTypeId :int) : int
+ setFoodTypeMapper(foodTypeMapper :FoodTypeMapper) : void
+ setMenuInfoMapper(menuInfoMapper :MenuInfoMapper) : void
```

## RestaurantTypeServiceImpl

```
- restTypeMapper: RestTypeMapper

+ setRestTypeMapper(restTypeMapper :RestTypeMapper) : void
+ getAllType() : List<RestType>
```

## SequenceServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- sequenceInfoMapper: SequenceInfoMapper
- userInfoMapper: UserInfoMapper
- restaurantInfoMapper: RestaurantInfoMapper
- seatService: SeatService
- pushService: PushService

+ setUserInfoMapper(userInfoMapper :UserInfoMapper) : void
+ setSequenceInfoMapper(sequenceInfoMapper :SequenceInfoMapper) : void
+ createSequence(sequenceInfo :SequenceInfo) : SequenceDetailForUser
+ cancelSeq(userId :int) : void
+ changeToEating(seqId :int) : void
+ getSequenceInfo(userId :int) : SequenceDetailForUser
+ toSequenceDetailForUser(sequenceInfo :SequenceInfo) : SequenceDetailForUser
+ getDateDay() : String
+ getSeqNo(restId :int) : int
+ getSeqNow(restId :int) : int
+ getSeqBefore(restId :int, seatType :int) : int
+ getSeatType(peopleNum :int) : int
+ getAllSeqInfosByRest(restId :int) : List<SequenceDetailForRest>
+ getPushUserId(sequenceInfo :SequenceInfo) : SequenceInfo
+ setRestaurantInfoMapper(restaurantInfoMapper :RestaurantInfoMapper) : void
```

## OrderServiceImpl

```
- LOG: Log = LoggerFactory.getL... {readOnly}
- orderInfoMapper: OrderInfoMapper
- menuInfoMapper: MenuInfoMapper
- restaurantInfoMapper: RestaurantInfoMapper
- mealInfoMapper: MealInfoMapper
- mealService: MealService
- seatService: SeatService
- pushService: PushService

+ getOrderByUser(userId :int, status :int) : List<OrderDetail>
+ getOrderByRest(restId :int, dateFrom :String, dateTo :String) : List<OrderDetail>
+ createOrder(orderInfo :OrderInfo) : OrderDetail
+ confirmOrder(orderId :int, operationUserId :int) : void
+ cancelOrder(orderId :int, userId :int) : void
+ updateOrder(orderInfo :OrderInfo) : int
+ getOrderDetailByOrder(orderInfo :OrderInfo) : OrderDetail
+ getOrderInfoById(orderId :int) : OrderInfo
+ setOrderInfoMapper(orderInfoMapper :OrderInfoMapper) : void
+ setMenuInfoMapper(menuInfoMapper :MenuInfoMapper) : void
+ setRestaurantInfoMapper(restaurantInfoMapper :RestaurantInfoMapper) : void
+ setMealInfoMapper(mealInfoMapper :MealInfoMapper) : void
```