客户名字和logo

餐厅订餐系统- 网络协议设计文档

wanglinglong

日期：2014-12-04

文档版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订人 | 审核人 | 变更内容 |
| V0.1 | 2014-11-19 | wanglinglong |  | 初始化 |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

目录 2

1 文档介绍 3

1.1 目的 3

1.2 范围 3

1.3 缩写词列表 3

1.4 参考内容 3

2 项目概述 3

3 客户端实现方案 3

3.1 菜单浏览 3

3.2 订单查询 5

3.3 系统设置 6

4 附录 6

# 文档介绍

## 目的

为客户端开发工程师和服务端开发工程师共同开发订餐系统提供指导，本文档特别规定了服务端和客户端的通讯协议。

## 范围

餐厅订餐系统详细设计阶段，代码编写阶段，供服务端开发工程师和客户端开发工程师参考使用。

## 缩写词列表

无

## 参考内容

无

# 项目概述

为餐厅设计的整套订餐系统，包括前端App订餐客户端软件，Web端订餐软件，Web端后台管理软件。

# 协议说明

基于http协议，根据接口的需要选择get或者post请求，服务返回byte数据流。交互的数据格式为json格式。

1.get请求：

所有请求参数拼接在请求的url链接中；服务器可以根据url做数据静态化缓存。

说明：可以通过shandle指定服务器返回的数据是否gizp压缩；默认gzip压缩。

2.post请求：

采用post表单方式提交请求参数。

说明：请求提交的post表单共三个参数项：handle,data,shandle

## 业务对象json数据单元

模块请求参数单元

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型 | 说明 | 备注 |
| moduleId | int | 模块id | 分为虚拟id和真实id两种；  <100000的id都是虚拟id |
| pageid | int | 请求的页码 | 首次请求传1;  如果想一次获取列表所有数据请传0; |

# 接口协议

业务对象json数据单元

1. 获取目前所有分类列表：

post请求，请求url地址：

http://xxxxxxx

输入：无

输出：分类列表，包含分类名，分类id字段，分类图片url;

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型 | 说明 | 备注 |
| moduleId | int | 模块id | 例如：20001 |
| moduleName | String | 模块名称 | 例如：最新 |
| icon | String | 该模块的图标 | 例如：http://www....33/21.jpg |
|  |  |  |  |

举例：

获取到当前列表为

｛

｛＂moduleId＂：＂20001＂，＂moduleName＂：＂最新＂，＂icon＂：＂http://www.google.com/2332333/21.jpg＂｝

｛＂moduleId＂：＂20002＂，＂moduleName＂：＂最热＂，＂icon＂：＂http://www.google.com/2332333/22.jpg＂｝

｛＂moduleId＂：＂20003＂，＂moduleName＂：＂川菜＂，＂icon＂：＂http://www.google.com/2332333/23.jpg＂｝

｛＂moduleId＂：＂20004＂，＂moduleName＂：＂湘菜＂，＂icon＂：＂http://www.google.com/2332333/24.jpg＂｝

｛＂moduleId＂：＂20005＂，＂moduleName＂：＂粤菜＂，＂icon＂：＂http://www.google.com/2332333/25.jpg＂｝

.......

}

说明：因为分类不会非常多，一次性下发所有分类.

2. 获取某一个分类下面所有菜品列表：

post请求，请求url地址：

Url = http://xxxxxxx

请求命令：　Url + “moduleId =” + moduleId

输入：分类id

输出：菜品列表，包含菜品名，菜品id字段，菜品价格，菜品图片url;

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. goodsId | 1. int | 1. 商品id | 1. 例如：30001 |
| 1. goodsName | 1. String | 1. 商品名称 | 1. 例如：麻婆豆腐 |
| 1. icon | 1. String | 1. 该商品的图标 | 1. 例如：http://www....33/31.jpg |
| 1. price | 1. Double | 1. 该商品的价格 | 1. 例如：48.5 |

举例：

获取川菜下面菜品列表为：

｛

｛＂goodsId＂：＂30001＂，＂goodsName＂：＂麻婆豆腐＂，＂price＂：＂18＂，＂icon＂：＂http://www.google.com/2332333/31.jpg＂｝

｛＂goodsId＂：＂30002＂，＂goodsName＂：＂豆花鱼＂，＂price＂：＂38＂，＂icon＂：＂http://www.google.com/2332333/32.jpg＂｝

｛＂goodsId＂：＂30003＂，＂goodsName＂：＂夫妻肺片＂，＂price＂：＂28＂，＂icon＂：＂http://www.google.com/2332333/33.jpg＂｝

.......

}

说明：因为某个分类下面商品不会非常多，所以一次性下发完成，不用增加page页．

３．查询某个会员的历史订餐记录：

post请求，请求url地址：

Url = http://xxxxxxx

请求命令：　Url + “userId =” + userId

输入：会员电话号码

输出：会员历史订单记录列表, 包含订单id,　订单日期，总价;

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. orderId | 1. int | 1. 订单id | 1. 例如：40001 |
| 1. time | 1. String | 1. 订单时间 | 1. 例如：2014-8-1 |
| 1. allPrice | 1. Double | 1. 订单的总价 | 1. 例如：104.5 |
|  |  |  |  |

举例：

获取某个会员的订单列表为：

｛

｛＂orderId＂：＂40001＂，＂time＂：＂2014-8-1＂,＂allPrice＂： "104.5"｝

｛＂orderId＂：＂40002＂，＂time＂：＂2014-9-15＂, ＂allPrice＂："234.5"｝

｛＂orderId＂：＂40003＂，＂time＂：＂2014-10-11＂, ＂allPrice＂："800"｝

.......

｝

说明：某个会员的订单应该还好，不会特别多，数据还好，我觉得可以一次性下发;

后续考虑是否要按时间段查询下发;

4．查询某个订单详情：

post请求，请求url地址：

Url = http://xxxxxxx

请求命令：　Url + “orderId =” + orderId

输入：订单id

输出：订单详情，包括订单号，订单日期，订单总价，订单菜品列表，单项菜品数量;

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. goodsId | 1. int | 1. 商品id | 1. 例如：30001 |
| 1. goodsName | 1. String | 1. 商品名称 | 1. 例如：麻婆豆腐 |
| 1. price | 1. Double | 1. 商品的单价 | 1. 例如：18 |
| 1. count | 1. int | 1. 商品的数量 | 1. 例如：１ |
| 1. icon | 1. String | 1. 商品的图标 | 1. 例如：http://www....33/31.jpg |

举例：

获取某个订单的详情为：

｛

｛＂goodsId＂：＂30001＂，＂goodsName＂：＂麻婆豆腐＂，＂price＂：＂18＂，　＂count＂：＂１＂，＂icon＂：＂http://www.google.com/2332/31.jpg＂｝

｛＂goodsId＂：＂60001＂，＂goodsName＂：＂鱼香肉丝＂，＂price＂：＂32＂，＂count＂：＂１＂，＂icon＂：＂http://www.google.com/2332/88.jpg＂｝

｛＂goodsId＂：＂80001＂，＂goodsName＂：＂紫菜蛋汤＂，＂price＂：＂28＂，＂count＂：＂１＂，＂icon＂：＂http://www.google.com/2333/92.jpg＂｝

｝

６．提交订单：

post请求，请求url地址：

Url = http://xxxxxxx

请求命令：　Url

输入：菜品id，菜品数量，会员id．

输出：ok或者fail