客户名字和logo

餐厅订餐系统- 网络协议设计文档

wanglinglong

日期：2014-12-31

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 |
|  | 2014-12-7 | wanglinglong |  | 修改部分内容 |
|  | 2014-12-08 | LUO ZHI |  | 更新格式 |
|  | 2014-12-8 | Xu Lin |  | 新增获取用户信息接口 |
|  | 2014-12-8 | Xu Lin |  | 更新URL |
|  | 2014-12-9 | Xu Lin |  | 加入测试记录 |
|  | 2014-12-10 | Xu Lin |  | 更新获取用户信息返回信息描述 |
|  | 2014-12-10 | wanglinglong |  | 修改文中的几处错误 |
|  | 2014-12-11 | Xu Lin |  | 更新URL |
| V1.0 | 2014-12-12 | Xu Lin |  | 增加图片地址说明 |
|  | 2014-12-27 | LUO ZHI |  | 增加REST风格说明 |

目录

[1 文档介绍 1](#_Toc407458586)

[1.1 目的 1](#_Toc407458587)

[1.2 范围 1](#_Toc407458588)

[1.3 缩写词列表 1](#_Toc407458589)

[1.4 参考内容 1](#_Toc407458590)

[2 协议说明 2](#_Toc407458591)

[2.1 REST风格的Web-Service 2](#_Toc407458592)

[2.2 JSON数据格式 2](#_Toc407458593)

[2.3 安全性 3](#_Toc407458594)

[3 接口协议 3](#_Toc407458595)

[3.1 获取用户信息 3](#_Toc407458596)

[3.2 获取目前所有分类列表 4](#_Toc407458597)

[3.3 获取某一个分类下面所有菜品列表： 6](#_Toc407458598)

[3.4 查询某个会员的历史订餐记录 8](#_Toc407458599)

[3.5 查询某个订单详情 10](#_Toc407458600)

[3.6 提交订单 12](#_Toc407458601)

# 文档介绍

## 目的

本文档对餐厅订餐系统对外曝露的服务接口进行定义说明。曝露的服务采用REST风格的Web-Service，使用这些服务，可以构建移动客户端，Web客户端，对系统进行外延的扩充。文档指导开发工程师进行Web-Service客户端开发；同时指导测试工程师变成测试用例。

## 范围

本文档仅对餐厅订餐系统Web-Service对外曝露的接口进行详细描述，同时简单阐述了Web-Service在安全性上的设计。对于如何编写客户端代码进行Web-Service调用则不在本文档描述范围内。

## 缩写词列表

|  |  |
| --- | --- |
| 缩写 | 说明 |
| HTTP | 超文本传送协议 (HTTP-Hypertext transfer protocol) |
| JSON | JavaScript对象表示法(JavaScript Object Notation) |
| URL | 统一资源定位符（Uniform Resource Locator） |
| REST | 表述性状态传递（Representational State Transfer） |
|  |  |

## 参考内容

|  |  |
| --- | --- |
| REST-维基百科 | http://zh.wikipedia.org/zh/REST |
| JSON | http://zh.wikipedia.org/wiki/JSON |
|  |  |

# 协议说明

餐厅订餐系统对外Web-Service采用REST风格定义，并使用JSON作为数据格式。

## REST风格的Web-Service

REST风格中文直译是含状态传输（Representational State Transfer），与传统的SOAP和XML-RPC等方式相比更加简洁与容易理解。REST风格扩充使用了HTTP协议中不常被使用的PUT和DELETE这2个动词（相对于GET和POST更为大家熟悉而言）。

REST风格使用URI来定义资源，每一个资源都有唯一的标识符；使用HTTP协议提供的GET/POST/PUT/DELETE四种方法来定义对资源的动作。恰好这四种方法又可以映射到数据库操作的CRUD四种基本操作。下表是一个简单的REST风格举例，

|  |  |  |  |
| --- | --- | --- | --- |
| 资源 | | User | |
| 动作 | GET | <http://host/ws/users> | 获取所有用户集合 |
| GET | <http://host/ws/users/1> | 获取ID为1的用户 |
| POST | <http://host/ws/users> | 创建一个新用户 |
| PUT | <http://host/ws/users/1> | 更新ID为1的用户数据记录 |
| DELETE | <http://host/ws/users/1> | 删除ID为1的用户数据记录 |

## JSON数据格式

JSON是一种类似于XML的数据表示格式，但是比XML更为简洁和轻量，它没有XML众多的Element节点，也不需要进行Validation，尤其JSON与JavaScript的天然集成，有效提高了开发效率。

在JSON的世界里，只有对象。JSON使用花括弧{}和键值对来定义对象，下面是一个使用JSON来表示用户对象的一个简单例子，

*{*

*“username”: “Jerry Gao”,*

*“password” : “12345”,*

*“mobile\_phone”: “13312345678”*

*“roles”: [*

*{*

*“role\_name”: “admin”*

*},*

*{*

*“role\_name”: “cashier”*

*}*

*]*

*}*

## 安全性

使用HTTPS加密通道以及BASE64加密算法对数据进行保护。

# 接口协议

## 获取用户信息

URL：http://192.168.0.100:8080/eorder-ws/rest/users/myuser/13988888888

方法：GET

输入：手机号码

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型 | 说明 | 备注 |
| cellphone | String | 手机号码 | 例如：13988888888 |

输出： 会员信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型 | 说明 | 备注 |
| userId | int | 分类id | 例如： 1 |
| userName | String | 分类名称 | 例如：张三 |
| cellphone | String | 该分类的图片 | 例如：13988888888 |
| levelName | String | 用户等级名称 | 金牌会员 |
| discount | Float | 折扣信息 | 例如：钻石8折、白金8.5折、金牌9折、普通会员9.5折等等。 |

举例：

获取到当前列表为

{

"user":

{

"userId": 1,

"userName": "张三",

"cellphone": “13988888888”,

"levelName": “金牌会员”,

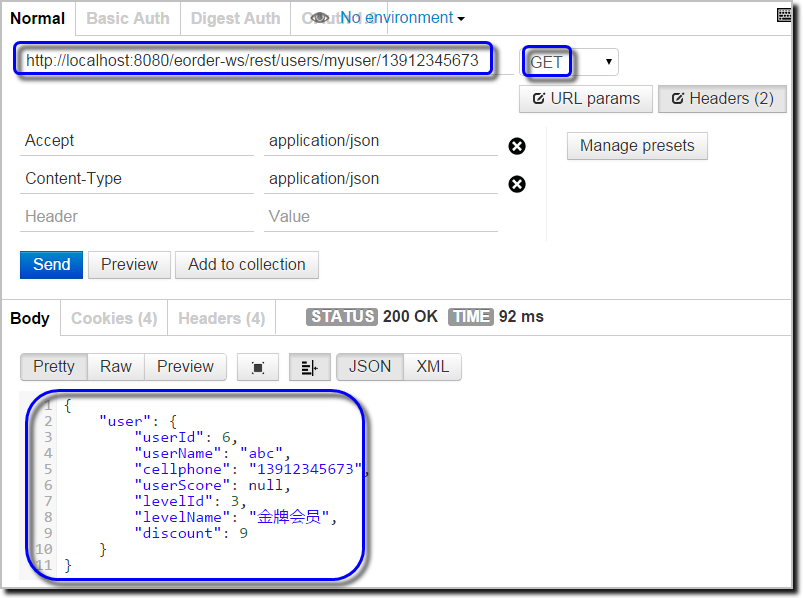
"discount：9

}

}

测试URL：http://localhost:8080/eorder-ws/rest/users/myuser/13912345673

测试结果：



## 获取目前所有分类列表

URL：http://192.168.0.100:8080/eorder-ws/rest/categories

方法：GET

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 |
| categoryId | int | 分类id | 例如： 1 |
| categoryName | String | 分类名称 | 例如：最新 |
| categoryPicture | String | 该分类的图片 | 例如：http://www....33/21.jpg |

图片地址说明：

数据库中存放的是相对地址，例如：/01/0106.png，如果要取图片，需要在前面添加域名和文件夹名称：http://192.168.0.100:8080/eorder-ws/images，最后的图片实际地址为：http://192.168.0.100:8080/eorder-ws/images/01/0106.png

举例：

获取到当前列表为

{

"categories": [

{

"categoryId": 1,

"categoryName": "最热",

"categoryPicture": null

},

{

"categoryId": 2,

"categoryName": "最新",

"categoryPicture": null

},

{

"categoryId": 3,

"categoryName": "湘菜",

"categoryPicture": null

}

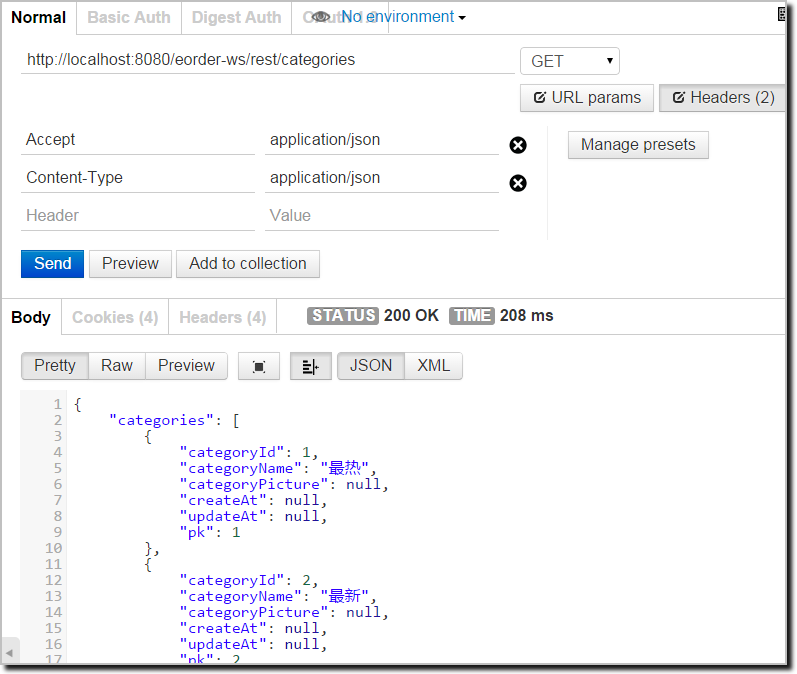
]

}

说明：因为分类不会非常多，一次性下发所有分类。

测试URL：http://localhost:8080/eorder-ws/rest/categories

测试结果：



## 获取某一个分类下面所有菜品列表：

URL：http://localhost:8080/eorder-ws/rest/dishes/mydishes/6

方法：GET

输入：分类id

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. categoryId | 1. int | 1. 分类id | 1. 例如： 1 |

输出：菜品列表，包含菜品名、菜品id字段、菜品价格和菜品图片URL

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. dishId | 1. int | 1. 菜品id | 1. 例如：30001 |
| 1. dishName | 1. String | 1. 菜品名称 | 1. 例如：麻婆豆腐 |
| 1. dishPicture | 1. String | 1. 该菜品的图片 | 1. 例如：http://www....33/31.jpg |
| 1. dishPrice | 1. Float | 1. 该菜品的价格 | 1. 例如：48.5 |

举例：

获取川菜下面菜品列表为：

｛dishes:[

｛＂dishId＂：＂30001＂，＂dishName＂：＂麻婆豆腐＂，＂dishPrice＂：＂18＂，＂dishPicture＂：＂http://www.google.com/2332333/31.jpg＂｝

｛＂dishId＂：＂30002＂，＂dishName＂：＂豆花鱼＂，＂dishPrice＂：＂38＂，＂dishPicture＂：＂http://www.google.com/2332333/32.jpg＂｝

｛＂dishId＂：＂30003＂，＂dishName＂：＂夫妻肺片＂，＂dishPrice＂：＂28.9＂，＂dishPicture＂：＂http://www.google.com/2332333/33.jpg＂｝]

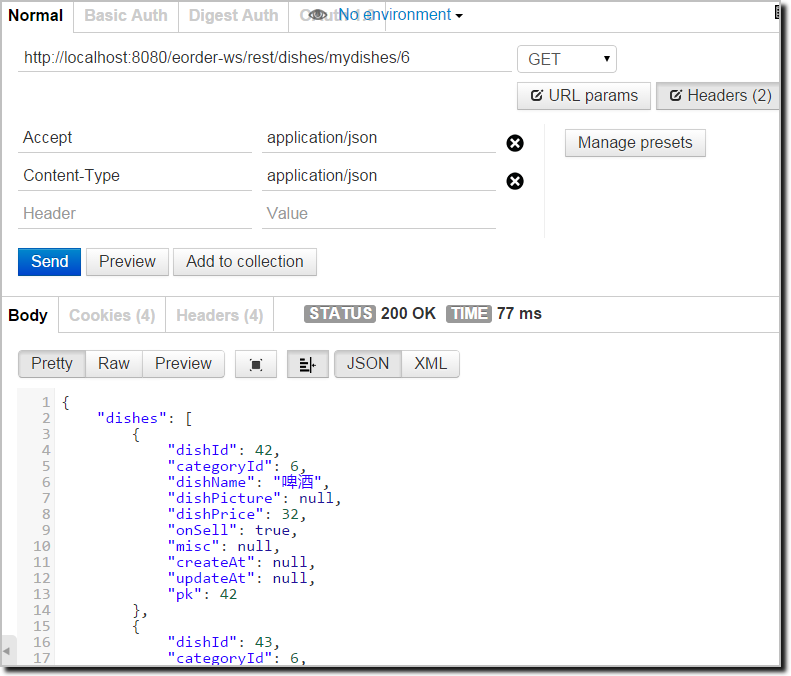
.......

}

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

测试URL：http://localhost:8080/eorder-ws/rest/dishes/mydishes/6

测试结果：



## 查询某个会员的历史订餐记录

URL：http://localhost:8080/eorder-ws/rest/orders/myorders/13912345673

方法：GET

输入：会员手机号码

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. cellphone | 1. String | 1. 手机号码 | 1. 例如： 13988888888 |

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： 1 |
| 1. createAt | 1. String | 1. 订单时间 | 1. 例如：2014-8-1 |
| 1. totalPrice | 1. Float | 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：

｛”orders”:

[ ｛＂orderId＂：＂40001＂，＂createAt＂：＂2014-8-1＂,＂totalPrice＂： "104.5"｝

｛＂orderId＂：＂40002＂，＂createAt＂：＂2014-9-15＂, ＂totalPrice＂："234.5"｝

｛＂orderId＂：＂40003＂，＂createAt＂：＂2014-10-11＂, ＂totalPrice＂："800"｝]

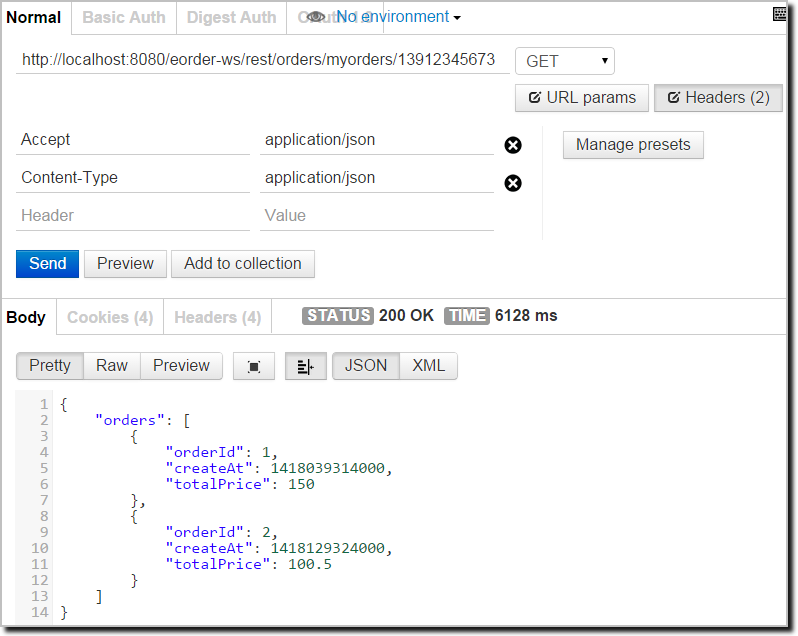
.......

｝

说明：一次性下发，后续考虑是否要按时间段查询下发;

测试URL：http://localhost:8080/eorder-ws/rest/orders/myorders/13912345673

测试结果：



## 查询某个订单详情

URL：http://localhost:8080/eorder-ws/rest/orderitems/myorderitems/{orderId}

方法：GET

输入：订单id

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 名称 | 1. 参数类型 | 1. 说明 | 1. 备注 |
| 1. orderId | 1. int | 1. 菜品id | 1. 例如： 1 |

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. dishId | 1. int | 1. 菜品id | 1. 例如： 1 |
| 1. dishName | 1. String | 1. 菜品名称 | 1. 例如：麻婆豆腐 |
| 1. dishPrice | 1. Float | 1. 菜品的单价 | 1. 例如：18 |
| 1. dishAmount | 1. int | 1. 菜品的数量 | 1. 例如：１ |
| 1. dishPicture | 1. String | 1. 菜品的图片 | 1. 例如：http://www....33/31.jpg |

举例：

获取某个订单的详情为：

｛ “orderitems”：[

｛＂dishId＂：＂30001＂，＂dishName＂：＂麻婆豆腐＂，＂dishPrice＂：＂18＂，　＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2332/31.jpg＂｝

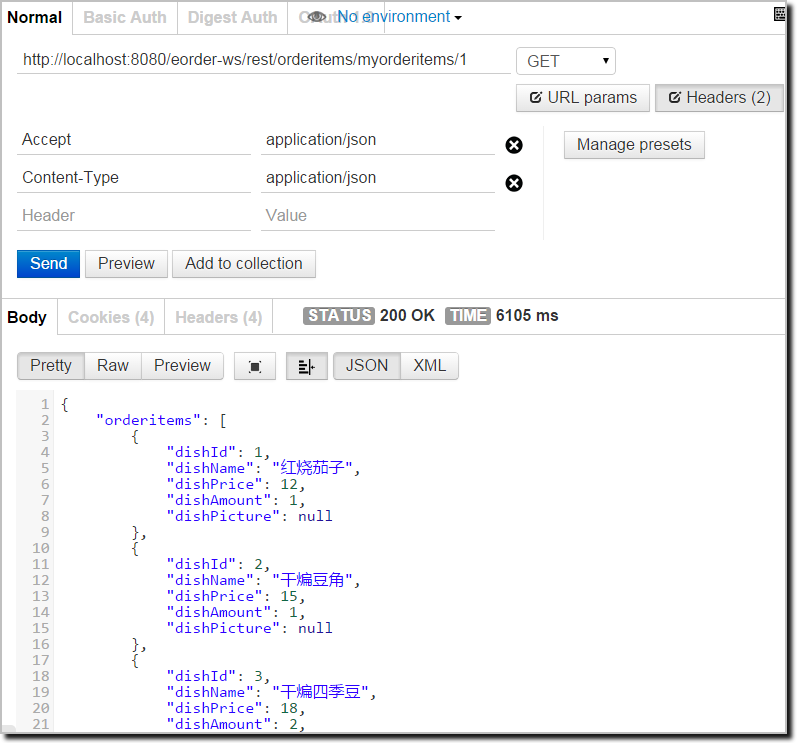
｛＂dishId＂：＂60001＂，＂dishName＂：＂鱼香肉丝＂，＂dishPrice＂：＂32＂，＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2332/88.jpg＂｝

｛＂dishId＂：＂80001＂，＂dishName＂：＂紫菜蛋汤＂，＂dishPrice＂：＂28＂，＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2333/92.jpg＂｝

]｝

测试URL：http://localhost:8080/eorder-ws/rest/orderitems/myorderitems/1

测试结果：



## 提交订单

URL：http://192.168.0.100:8080/eorder-ws/rest/orders/

方法：POST

输入：电话号码，桌号，员工号，总价，菜品列表

输出：ok或者fail

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型 | 说明 | 备注 |
| cellphone | String | 会员手机号码 | 例如：13788886666 |
| tableId | Int | 桌号 | 不可以为空 |
| servantId | String | 员工号 | 二维码扫描 |
| dishPrice | Float | 菜品的总价 | 例如：90.5 |
| dishList | Aarry | 菜品列表 | 例如：见下图示例 |

举例：

提交某个订单的数据为：

｛

＂cellphone＂：＂13788886666＂，

＂tableId＂: ＂2003＂,

＂servantId＂: ＂2014003＂

＂dishPrice＂：＂90＂，

＂dishList＂：[

｛＂dishId＂：＂30001＂，＂dishName＂：＂麻婆豆腐＂，＂price＂：＂18＂，＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2332/31.jpg＂｝

｛＂dishId＂：＂60001＂，＂dishName＂：＂鱼香肉丝＂，＂dishPrice＂：＂32＂，＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2332/88.jpg＂｝

｛＂dishId＂：＂80001＂，＂dishName＂：＂紫菜蛋汤＂，＂dishPrice＂：＂28＂，＂dishAmount＂：＂１＂，＂dishPicture＂：＂http://www.google.com/2333/92.jpg＂｝

｛＂dishId＂：＂90002＂，＂dishName＂：＂米饭＂，＂dishPrice＂：＂4＂，＂dishAmount＂：＂3＂，＂dishPicture＂：＂http://www.google.com/2333/022.jpg＂｝

]

｝