点餐系统

接口使用说明书

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ]正式发布  [√] 正在修改 | 文件标识： | 001 |
| 当前版本： | X.Y |
| 作 者： | 魏玉凤 |
| 完成日期： |  |

**目 录**

[一、 接口概述 3](#_Toc17373)

[1.1 概述 3](#_Toc12052)

[1.2 基本规则 3](#_Toc25498)

[1.3 域名 3](#_Toc1652)

[1.4 编写目的 3](#_Toc13267)

[二、 接口定义 3](#_Toc9313)

[2.1 用户管理 3](#_Toc31074)

[2.1.1 添加用户 3](#_Toc5198)

[2.1.2 登录 4](#_Toc3051)

[2.1.3 修改密码 5](#_Toc18788)

[2.1.4 查询所有用户 5](#_Toc27583)

[2.1.5 验证用户名是否重复 6](#_Toc23884)

[2.2 菜单管理 6](#_Toc1015)

[2.2.1 增加菜单 6](#_Toc20080)

[2.2.2 修改菜单 7](#_Toc25154)

[2.2.3 删除菜单 7](#_Toc264)

[2.2.4 查询全部菜单 8](#_Toc14901)

[2.2.5 上传图片 8](#_Toc8611)

[2.3 订餐管理 9](#_Toc30759)

[2.2.1 订餐 9](#_Toc15648)

[2.2.2 取消订餐 9](#_Toc2011)

[2.2.3 产生打扫人员 10](#_Toc16676)

[2.2.4 获取今日订餐人员信息 10](#_Toc21895)

[2.2.5 统计今日订餐人数 10](#_Toc29078)

[2.2.2 获取个人订餐信息 11](#_Toc1011)

## 接口概述

### 概述

接口用于点餐系统和外界系统进行数据交互以完成某种业务流程。

### **基本规则**

### **域名**

http://www.\*\*\*.com

### **编写目的**

预期读者：

对接口行为和目的有一定了解的人。

背景说明 

软件系统名称：接口前端接入服务器。 

描述 <接口> 接收不同商家的接入，接收数据转发给主机服务器，并同步/异步将 返回数据发给接入商家的行为。

## 接口定义

### 2.1 用户管理

#### 2.1.1 添加用户

1. 请求API

http://192.168.2.251:8080/api/User/AddUser/{user}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| user | 用户的信息 |
| user.UserId | 员工号 |
| user.UserName | 用户名 |
| user.UserPwd | 用户密码 |
| user.Deptno | 部门号 |
| user.UserType | 用户类型(1管理员，2普通用户) |
| user.Gender | 性别 |

3. 请求参数示例

http://192.168.2.251:8080/api/User/AddUser?user={

"UserId":44444,

"UserName":"zhangsan",

"UserPwd":"456789",

"Deptno":1,

"UserType":2,

"Gender":"ddd"}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | true | 增加成功 |
| 500 | “系统繁忙” | 增加失败 |

#### 2.1.2 登录

1. 请求API

http://192.168.2.251:8080/api/User/Login?userId={userId}&pwd={pwd}&type={type}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| userId | 用户工号 |
| pwd | 用户密码 |
| type | 用户类型 |

3. 请求参数示例

{

“userId”:1001,

“pwd”:”123”,

“type”:1

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | UserInfo | 登录成功，返回User对象 |
| 404 | “用户名或密码错误” | 用户名密码不匹配 |
| 500 | “系统繁忙” | 登录失败 |

#### 2.1.3 修改密码

1. 请求API

http://192.168.2.251:8080/api/User/MotifyPassword?userId={userId}&newPassword={newPassword}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| userId | 用户编号 |
| newPassword | 新密码 |

3. 请求参数示例

{

“userId”:1,

“newPassword”:”123”

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | true | 修改成功 |
| 500 | “系统繁忙” | 修改失败 |

#### 2.1.4 查询所有用户（分页）

1. 请求API

http://192.168.2.251:8080/api/User/GetAllUser?rows={rows}&page={page}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| rows | 每页显示的行数 |
| page | 当前页数 |

3. 请求参数示例

{

“rows”:5,

“page”:”2”

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | Page | 查询成功，返回分页对象 |
| 200 | [] | 当前没有用户 |
| 500 | “系统繁忙” | 查询失败 |

#### 2.1.5 删除用户

1. 请求API

http://192.168.2.251:8080/api/User/DelUser?userId={userId}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| userId | 用户编号 |

3. 请求参数示例

{

“userId”:1

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | true | 删除成功 |
| 200 | false | 删除失败 |
| 500 | “系统繁忙” | 删除失败 |

### 2.2 部门管理

#### 2.2.1 查询所有部门

1. 请求API

http://192.168.2.251:8080/api/Dept/GetAllDept

2. 请求参数描述

无

3. 请求参数示例

无

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | List<DeptInfo> | 部门查询成功，返回List集合 |
| 200 | [] | 部门为空 |
| 500 | “系统繁忙” | 菜单查询失败 |

### 2.3 订餐管理

#### 2.2.1 订餐

1. 请求API

http://192.168.2.251:8080/api/Order/AddOrder?order

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| order | 订餐信息 |
| order.UserId | 用户编号 |
| order.Remark | 订餐备注 |

3. 请求参数示例

http://192.168.2.251:8080/api/Order/AddOrder?order={

"UserId":1005,

"Remark":"ddd"}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | “OK” | 订餐成功 |
| 500 | “系统繁忙” | 订餐失败 |
| 404 | “已超过订餐时间” | 订餐失败 |

#### 2.2.2 取消订餐

1. 请求API

http://192.168.2.251:8080/api/Order/CancelOrder?userId={userId}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| userId | 用户编号 |

3. 请求参数示例

{

“userId”:1

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | “OK” | 取消订餐成功 |
| 500 | “系统繁忙” | 取消订餐失败 |
| 404 | “已超过取消订餐时间” | 取消订餐失败 |

#### 2.2.3 产生打扫人员

1. 请求API

http://192.168.2.251:8080/api/Order/ProduceSweep

2. 请求参数描述

无

3. 请求参数示例

无

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | “OK” | 打扫人员产生成功 |
| 500 | “系统繁忙” | 打扫人员产生失败 |
| 404 | “未到订餐截止时间” | 打扫人员产生失败 |
| 303 | “无人订餐” | 打扫人员产生失败 |

#### 2.2.4 获取今日订餐人员信息（分页、按部门）

1. 请求API

http://192.168.2.251:8080/api/Order/GetOrderPeople?rows={rows}&page={page}&deptId={deptId}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| rows | 每页显示的行数 |
| page | 当前页数 |
| deptId | 部门ID |

3. 请求参数示例

{

“rows”:5,

“page”:”2”,

“deptId”:1

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | Page | 今日订餐人员分页 |
| 200 | [] | 今日还未有人订餐 |
| 500 | “系统繁忙” | 查询失败 |

#### 2.2.5 统计今日订餐人数

1. 请求API

http://192.168.2.251:8080/api/Order/ConutOrderNumber

2. 请求参数描述

无

3. 请求参数示例

无

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | number | 统计成功，返回int类型 |
| 500 | “系统繁忙” | 统计失败 |

#### 2.2.6 今日是否产生打扫人员

1. 请求API

http://192.168.2.251:8080/api/Order/WetherProduce

2. 请求参数描述

无

3. 请求参数示例

无

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | true | 打扫人员已产生 |
| 200 | false | 打扫人员未产生 |
| 500 | “系统繁忙” | 判断失败 |

#### 2.2.7 用户今日是否订餐

1. 请求API

http://192.168.2.251:8080/api/Order/UserIdOrder?userid={userid}

2. 请求参数描述

|  |  |
| --- | --- |
| **参数** | **说明** |
| userId | 用户编号 |

3. 请求参数示例

{

“userId”:1

}

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | true | 该用户今日已订餐 |
| 200 | false | 该用户今日未订餐 |
| 500 | “系统繁忙” | 获取失败 |

#### 2.2.8 获取打扫人员名字

1. 请求API

http://192.168.2.251:8080/api/Order/GetClearName

2. 请求参数描述

无

3. 请求参数示例

无

4. 返回状态码说明

|  |  |  |
| --- | --- | --- |
| **状态码** | **结果** | **描述** |
| 200 | IList<UserInfo> | 打扫人员集合 |
|  | [] | 还未产生打扫人员 |
| 500 | “系统繁忙” | 获取失败 |