注：所有的返回数据都是同一个类ResultData.java，不要重新去写。

# 一、用户管理模块

## 1、查询所有用户接口

**接口地址：**

http:127.0.0.1:8081/api/alluser

**请求类型：**

get

**请求参数：**

无

**返回数据格式：**

成功：

{

code: 0,

data: {

users: [{

uName: '张三',

uType: '学生',

uId: '1004002001',

uSex: '0',

csName: '修车191',

uPhone: '12345678910',

uAvaliable: '1'

}, {

uName: '张三',

uType: '教师',

uId: '1004002001',

uSex: '1',

csName: '修车191',

uPhone: '12345678910',

uAvaliable: '1'

}

]

}

}

注意：数据库中没有utype字段，需要先去查询用户身份再组装数据再返回。

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}

## 2、更新用户信息接口

**请求地址：**

http:127.0.0.1:8081/api/updateuser

**请求方式:**

put

**请求参数:**

{

uName:'张三'

,uId:'1004054521'

,uPhone:'13891347495'

,uSex:'0'

,csName:'修车2106'

,uAvaliable:'1'

}

注意：性别0是男，1是女。

**返回数据格式：**

成功：

{

code:0,

data:{ }

}

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}

## 3、重置用户密码接口

**注意：**重置的用户密码和­学号一样

**请求地址：**

http:127.0.0.1:8081/api/resetpas

**请求方式：**

put

**请求参数：**

{

uId:’100405421’

}

**返回数据格式：**

成功：

{

code:0,

data:{ }

}

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}

## 4、删除用户接口

**注意：**删除用户肯定会涉及到级联删除，这一点在后端写，不在数据库中写触发器，所以逻辑要理顺，应该先删除哪些有关的数据，再删哪些，要想清楚。

**请求地址：**

http:127.0.0.1:8081/api/deleteuser

**请求方式：**

delete

**请求参数：**

{

uId:’100405421’

}

**返回数据格式：**

成功：

{

code:0,

data:{ }

}

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}

## 5、新增用户接口

**请求地址**：

<http://127.0.0.1:8081/api/adduser>

**请求方式:**

post

**请求参数:**

{

uName:'王鑫'

,uId:'1740501210601'

,uPhone:'13891347495'

,uSex:'0'

,csName:'软件2106'

}

**返回数据格式：**

成功：

{

code:0,

data:{ }

}

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}

## 6、查询所有班级接口

请求地址：

<http://127.0.0.1:8081/api/allclass>

请求方式:

Get

请求参数：

无

数据返回格式：

成功：

{

code:0,

data:[{

csId:"@integer(60, 100)",

csName:"@string(7, 20)"

},{

csId:"@integer(60, 100)",

csName:"@string(7, 20)"

},{

csId:"@integer(60, 100)",

csName:"@string(7, 20)"

},{

csId:"@integer(60, 100)",

csName:"@string(7, 20)"

},{

csId:"@integer(60, 100)",

csName:"@string(7, 20)"

}]

}

## 7、查询该用户所有的角色

请求地址：

<http://127.0.0.1:8081/api/allrole>

请求方式:

Get

请求参数:

{

uId:’ @string(7, 20)’

}

数据返回格式：

{

code:0,

data:{[{

rId:'@integer(60, 100)',

rName:'@string(7, 20)',

rDesc:'@string(7, 20)',

selected:'@integer(0, 1)'

},{

rId:'@integer(60, 100)',

rName:'@string(7, 20)',

rDesc:'@string(7, 20)',

selected:'@integer(0, 1)'

},{

rId:'@integer(60, 100)',

rName:'@string(7, 20)',

rDesc:'@string(7, 20)',

selected:'@integer(0, 1)'

}]

}

}

## 8、分配角色接口

请求地址:

<http://127.0.0.1:8081/api/distrole>

请求方式：

Post

请求参数：

{

uId:vm.uId,

ids:[123,132,13,45,465]

}

数据返回格式：

成功：

{

code:0,

data:{ }

}

失败：

{

code: -1,

data: { }

}

服务器异常：

{

code:-10,

data:{ }

}