# 开发规范api指南

[开发规范api指南 1](#_Toc486944707)

[一.数据地址及表结构 3](#_Toc486944708)

[二.添加用户和组操作的实现 4](#_Toc486944709)

[2.1前端实现规范 4](#_Toc486944710)

[2.2后端实现规范 8](#_Toc486944711)

[三.一般查询操作的实现 10](#_Toc486944712)

[3.1前端实现规范 10](#_Toc486944713)

[3.2后端实现规范 12](#_Toc486944714)

## 一.数据地址及表结构

**数据库地址**：***jdbc:mysql://111.207.189.4:3306/gcxcszw\_crud***

**数据表**：

表1. 组表DEMO\_GROUP

CREATE TABLE `DEMO\_GROUP` (

`group\_Id` int(11) NOT NULL AUTO\_INCREMENT,

`group\_name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`group\_Id`)

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8;

表2. 用户表DEMO\_USER

CREATE TABLE `DEMO\_USER` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(255) DEFAULT NULL,

`user\_pwd` varchar(255) DEFAULT NULL,

`user\_sex` varchar(255) DEFAULT NULL,

`user\_age` int(11) DEFAULT NULL,

`user\_group\_fk` int(11) DEFAULT NULL,

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=utf8;

**源码（含前后台）svn地址**：***https://111.207.189.4/svn/gcx/creditwebs/citygovcreditcrudapi***

## 二.添加用户和组操作的实现

### 2.1前端实现规范

注截图参考代码请参考svn地址查找源码。

#### 2.1.1方式一：逐个提取dom数据

##### 简述

逐个提取dom节点数据，组装后台参数，调用ajax实现交互。

##### 页面 demo\_user\_and\_group.html：



##### 步骤1：引入js

文件开头引入

<script src="../js/jquery-1.7.1.min.js"></script>

<script src="../js/json2.js"></script>

文件尾部引入

##### 步骤2：前端参数提取及封装

//后台参数对象

**var** args = {};

//实例化组对象信息

**var** group = {};

//页面取值：组-组名

group.groupName = $("[name=groupName]").val();

//组装后台参数对象-赋值组对象

args.group = group;

//实例化用户对象信息

**var** user = {};

//页面取值：用户-用户名

user.userName = $("[name=userName]").val();

//页面取值：用户-密码

user.userPwd = $("[name=userPwd]").val();

//页面取值：用户-性别

user.userSex = $("[name=userSex]").val();

//页面取值：用户-年龄

user.userAge = $("[name=userAge]").val();

//组装后台参数对象-赋值用户对象

args.user = user;

//后台参数转换成json串

**var** jsonArgs = JSON.stringify(args);

//打印验证json串，查看是否符合后台需求，出差可以将该json串从控制台打印copy-paste to 后台开发人员

console.log('args = '+ args);

##### 步骤3：ajax调用

（注：响应处理部分仅供参考，具体处理视需求而定！）

$.ajax({

url:'/crud/user/addUserByGroup',

type:'POST',

contentType:'application/json;charset=utf-8',

data:jsonArgs,

async:**true**,

dataType:'json',

success:**function**(result){

//打印回调的相应正文

console.log('response text = '+JSON.stringify(result));

**var** msg = result.data.code;

**if**(msg==200){

alert("添加成功...");

}**else** **if**(msg==500){

alert("添加失败...");

}

//延时刷新页面

setTimeout("window.location.reload()", 1000);

}

});

完整js交互代码截图如下



#### 2.1.2方式二：表单萃取数据

##### 简述

通过form萃取数据调用ajax实现交互。

##### 页面 demo2\_user\_and\_group.html：



##### 步骤1：引入js （注：缺一不可，顺序不能颠倒）

<script src=*"../js/jquery-1.7.1.min.js"*></script>

<script src=*"../js/jquery.form.js"*></script>

<script src=*"../js/json2.js"*></script>

<script src=*"../js/set.js"*></script>

<script src=*"../js/map.js"*></script>

<script src=*"../js/extraData.js"*></script>

##### 步骤2：form表单提取数据

/\*\*通过form表单提取数据\*\*/

**var** targetJson = fromFormToJson('userForm');

console.log('targetJson='+targetJson);

##### 步骤3：ajax请求

注：响应处理部分仅供参考，具体处理视需求而定！

$.ajax({

url:'/crud/user/addUserByGroup',

type:'POST',

contentType:'application/json;charset=utf-8',

data:targetJson,

async:**true**,

dataType:'json',

success:**function**(result){

//打印回调的相应正文

console.log('response text = '+JSON.stringify(result));

**var** msg = result.data.code;

**if**(msg==200){

alert("添加成功...");

}**else** **if**(msg==500){

alert("添加失败...");

}

//延时刷新页面

setTimeout("window.location.reload()", 5000);

}

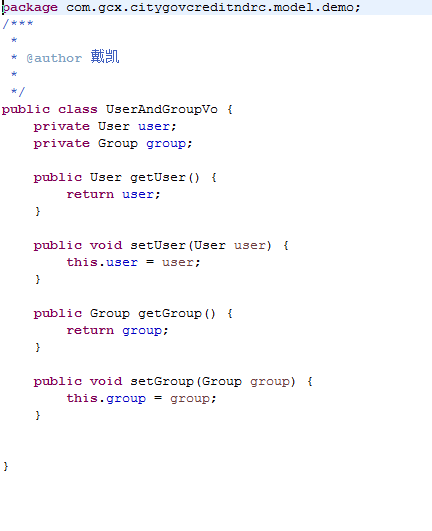
});

完整js交互代码截图如下



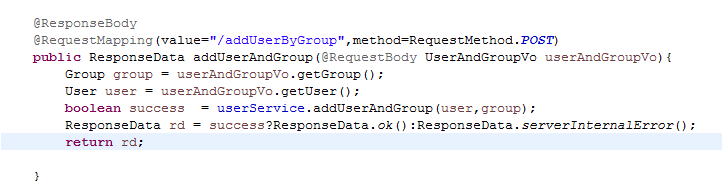
### 2.2后端实现规范

#### 步骤1. Vo代码设计如下



注：UserAndGroupVo引用对象均系代码生成代码。

#### 步骤2.Controller代码实现如下



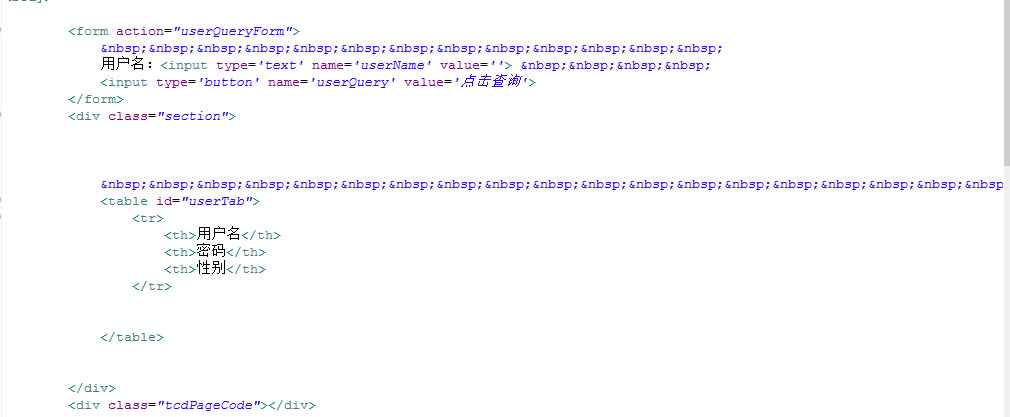
注：@ResponseBody @RequestMapping @RequestBody注解一个不能少，注意位置。

## 三.一般查询操作的实现

注：具体视需求而定，代码仅供参考

### 3.1前端实现规范

#### 页面 demo\_query\_users.html：



#### 步骤1：引入js css

<link rel=*"stylesheet"* type=*"text/css"* href=*"../css/jquery.page.css"*>

<script src="../js/util.1.0.js"></script>

<script src=*"../js/jquery-1.7.1.min.js"*></script>

<script src=*"../js/jquery.page.js"*></script>

<script src=*"../js/json2.js"*></script>

#### 步骤2：js的实现部分

**function** load\_list(){

**var** userName\_ = $('[name=userName]').val();

$.ajax({

url:'/crud/user/queryEntry',

type:'GET',

dataType:'json',

data:{pageNo:pagenow,pageSize:pagesize,userName:userName\_},

success:**function**(result){

console.log(JSON.stringify(result));

**if**(pageflag){

//生成分页栏

pageall = Math.ceil(result.data.total/pagesize);

loadpages();

}

**var** list = result.data.result;

$("#userTab").empty();

**var** dataHeader ="<tr><th>用户名</th><th>密码</th><th>性别</th><th>年龄</th></tr>";

$("#userTab").append(dataHeader);

**for**(**var** i=0; i<list.length; i++){

**var** record = list[i];

**var** dataItem = "";

dataItem += "<tr>";

dataItem += "<td>"+isnull(record.userName)+"</td>";

dataItem += "<td>"+isnull(record.userPwd)+"</td>";

dataItem += "<td>"+isnull(record.userSex)+"</td>";

dataItem += "<td>"+isnull(record.userAge)+"</td></tr>";

$("#userTab").append(dataItem);

}

},

error: **function**(){

//console.log("error");

}

});

}

**function** isnull(a){

**if**(a==**null**||a==undefined ||a==""){

**return** "";

}**else**{

**return** a;

}

}

//以下两个参数为初始化参数，可根据具体要求修改

//当前页码

**var** pagenow = 1;

//单页显示条数

**var** pagesize = 5;

//单行显示列数

**var** colsize = 3;

//总页数

**var** pageall = 1;

//分页开关

**var** pageflag = **true**;

//获取总页码，添加分页

**function** loadpages(){

//加载分页

$(".tcdPageCode").createPage({

pageCount:pageall,//总页数

current:pagenow,//初始化页码为1， 代表第一页

backFn:**function**(p){//p当前页

pagenow = p;

load\_list();

}

});

}

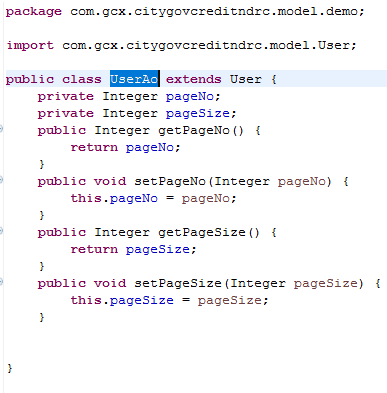
$(document).ready(**function**(){

load\_list();

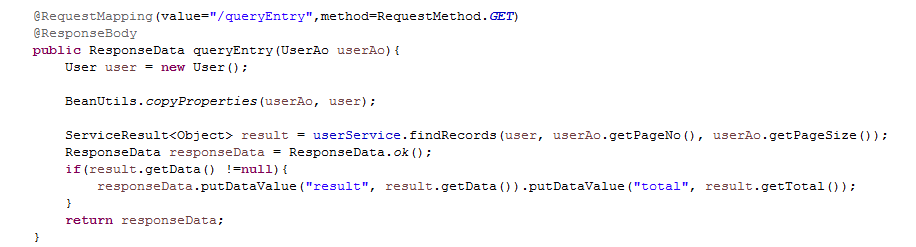
});

### 3.2后端实现规范

#### 步骤1：Ao的写法



#### 步骤2：controller的写法



## 四 修改操作的实现

### 4.1前端实现规范

#### 页面 demo\_modify\_user.html：



#### 步骤一 修改页面引入的js脚本

注：顺序不能颠倒

<script src=*"../js/jquery-1.7.1.min.js"*></script>

<script src=*"../js/util.1.0.js"*></script>

<script src=*"../js/jquery.page.js"*></script>

<script src=*"../js/json2.js"*></script>

<script src=*"../js/set.js"*></script>

<script src=*"../js/map.js"*></script>

<script src=*"../js/extraData.js"*></script>

#### 步骤二 查询列表项页面html事件源.

注: 参考demo\_query\_users.html页面



#### 步骤三 查询列表项页面事件源的js定义

注: 参考demo\_query\_users.html页面



注：参数传递含两种情况，缓存可用，否则静态页面直接通过？参数名=参数值拼接传递

#### 步骤四 修改页面加载数据的js

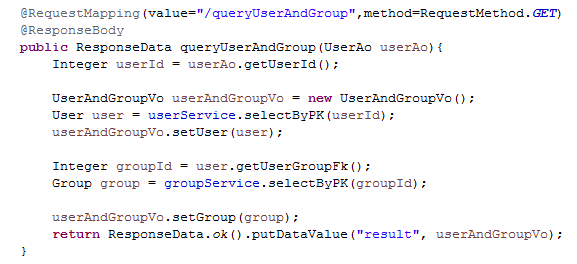


#### 步骤五 提取修改数据做前后台交互

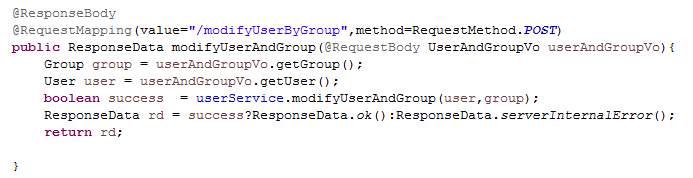


### 4.2后端实现规范

#### 步骤一 查询controller实现



#### 步骤二 修改controller实现



## 五 详情页面的实现

### 5.1 前端实现规范

#### 页面 demo\_detail\_user.html



#### 步骤一 详情页面引入的js脚本

注：顺序不能颠倒

<script src=*"../js/jquery-1.7.1.min.js"*></script>

<script src=*"../js/util.1.0.js"*></script>

<script src=*"../js/jquery.page.js"*></script>

<script src=*"../js/json2.js"*></script>

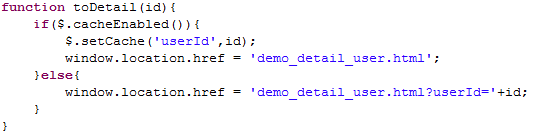
#### 步骤二 查询列表项页面html事件源.

注: 参考demo\_query\_users.html页面



#### 步骤三 查询列表项页面事件源的js定义

注: 参考demo\_query\_users.html页面



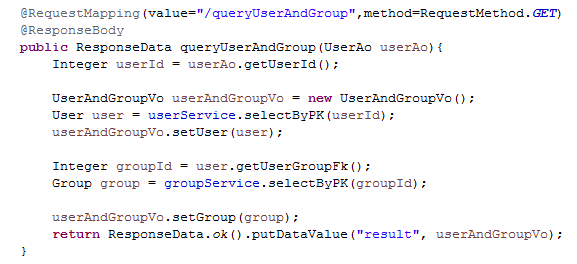
注：参数传递含两种情况，缓存可用，否则静态页面直接通过？参数名=参数值拼接传递

#### 步骤四 详情页面加载数据的js



### 5.2 后端实现规范

#### 步骤一 查询controller的实现



## 六 多文件上传的实现

注:仅供参考，不作为标准推广

#### 6.1前端实现规范

页面multiFileUpload.html



步骤一 引入js脚本

步骤二 JS实现

#### 6.2后端实现规范