[**C#开发微信门户及应用(18)-微信企业号的通讯录管理开发之成员管理**](http://www.cnblogs.com/wuhuacong/p/3995484.html)

在上篇随笔《[C#开发微信门户及应用(17)-微信企业号的通讯录管理开发之部门管理](http://www.cnblogs.com/wuhuacong/p/3992230.html)》介绍了通讯录的部门的相关操作管理，通讯录管理包括部门管理、成员管理、标签管理三个部分，本篇主要介绍成员的管理操作，包括创建、删除、更新、获取、获取部门成员几个操作要点。

**1、成员的创建操作**

为了方便，我们可以创建一个部门组织结构，这是开发的前提，因为我们通讯录管理，也是基于一个组织机构下的，如上篇介绍的组织结构层次一样。我这里创建一个广州爱奇迪的根结构，然后在其中在创建一些组织机构，如下图所示。



在后台可以通过功能操作添加人员，本篇主要介绍如何调用微信企业号API进行人员管理的操作。

创建人员的API定义如下所示。

* 请求说明

Https请求方式: POST

<https://qyapi.weixin.qq.com/cgi-bin/user/create?access_token=ACCESS_TOKEN>

请求包结构体为:

{

"userid": "zhangsan",

"name": "张三",

"department": [1, 2],

"position": "产品经理",

"mobile": "15913215421",

"gender": 1,

"tel": "62394",

"email": "zhangsan@gzdev.com",

"weixinid": "zhangsan4dev"

}

* 参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 员工UserID。对应管理端的帐号，企业内必须唯一。长度为1~64个字符 |
| name | 是 | 成员名称。长度为1~64个字符 |
| department | 否 | 成员所属部门id列表。注意，每个部门的直属员工上限为1000个 |
| position | 否 | 职位信息。长度为0~64个字符 |
| mobile | 否 | 手机号码。企业内必须唯一，mobile/weixinid/email三者不能同时为空 |
| gender | 否 | 性别。gender=0表示男，=1表示女。默认gender=0 |
| tel | 否 | 办公电话。长度为0~64个字符 |
| email | 否 | 邮箱。长度为0~64个字符。企业内必须唯一 |
| weixinid | 否 | 微信号。企业内必须唯一 |

* 权限说明

管理员须拥有“操作通讯录”的接口权限，以及指定部门的管理权限。

* 返回结果

{

"errcode": 0,

"errmsg": "created"

}

 我们在C#里面，需要定义对应给的接口，然后根据需要构造对应的传递实体信息。

这里我把人员管理的接口全部定义好，接口定义如下所示。

[复制代码](javascript:void(0);)

#region 部门成员管理

/// <summary>

/// 创建成员

/// </summary>

CommonResult CreateUser(string accessToken, CorpUserJson user);

/// <summary>

/// 更新成员

/// </summary>

CommonResult UpdateUser(string accessToken, CorpUserUpdateJson user);

/// <summary>

/// 删除成员

/// </summary>

CommonResult DeleteUser(string accessToken, string userid);

/// <summary>

/// 根据成员id获取成员信息

/// </summary>

CorpUserGetJson GetUser(string accessToken, string userid);

/// <summary>

/// 获取部门成员

/// </summary>

CorpUserListJson GetDeptUser(string accessToken, int department\_id, int fetch\_child = 0, int status = 0);

#endregion

[复制代码](javascript:void(0);)

然后根据信息定义，创建一个承载人员信息的CorpUserJson实体对象，创建人员的实现操作代码如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 创建成员

/// </summary>

public CommonResult CreateUser(string accessToken, CorpUserJson user)

{

string urlFormat = "https://qyapi.weixin.qq.com/cgi-bin/user/create?access\_token={0}";

var data = new

{

userid = user.userid,

name = user.name,

department = user.department,

position = user.position,

mobile = user.mobile,

gender = user.gender,

tel = user.tel,

email = user.email,

weixinid = user.weixinid

};

var url = string.Format(urlFormat, accessToken);

var postData = data.ToJson();

return Helper.GetCorpExecuteResult(url, postData);

}

[复制代码](javascript:void(0);)

**2、成员的更新操作**

成员的数据更新和创建操作类似，它的企业号定义如下所示。

* 请求说明

Https请求方式: POST

<https://qyapi.weixin.qq.com/cgi-bin/user/update?access_token=ACCESS_TOKEN>

请求包示例如下（如果非必须的字段未指定，则不更新该字段之前的设置值）:

{

"userid": "zhangsan",

"name": "李四",

"department": [1],

"position": "后台工程师",

"mobile": "15913215421",

"gender": 1,

"tel": "62394",

"email": "zhangsan@gzdev.com",

"weixinid": "lisifordev",

"enable": 1

}

由于它的操作数据类似，因此它的实现代码也差不多，如下所示就是。

[复制代码](javascript:void(0);)

/// <summary>

/// 更新成员

/// </summary>

public CommonResult UpdateUser(string accessToken, CorpUserUpdateJson user)

{

string urlFormat = "https://qyapi.weixin.qq.com/cgi-bin/user/update?access\_token={0}";

//string postData = user.ToJson();

var data = new

{

userid = user.userid,

name = user.name,

department = user.department,

position = user.position,

mobile = user.mobile,

gender = user.gender,

tel = user.tel,

email = user.email,

weixinid = user.weixinid,

enable = user.enable

};

var url = string.Format(urlFormat, accessToken);

var postData = data.ToJson();

return Helper.GetCorpExecuteResult(url, postData);

}

[复制代码](javascript:void(0);)

**3、成员的删除、成员的获取、部门成员的获取操作**

这些操作和上面的类似，不在赘述，主要就是根据需要定义他们对应的返回数据信息，然后解析Json数据即可转换为对应的实体。

1）删除人员的定义如下：

* 请求说明

Https请求方式: GET

<https://qyapi.weixin.qq.com/cgi-bin/user/delete?access_token=ACCESS_TOKEN&userid=lisi>

* 参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 员工UserID。对应管理端的帐号 |

* 返回结果

{

"errcode": 0,

"errmsg": "deleted"

}

2）成员的获取定义如下：

* 请求说明

Https请求方式: GET

<https://qyapi.weixin.qq.com/cgi-bin/user/get?access_token=ACCESS_TOKEN&userid=lisi>

* 参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 员工UserID |

* 返回结果

{

"errcode": 0,

"errmsg": "ok",

"userid": "zhangsan",

"name": "李四",

"department": [1, 2],

"position": "后台工程师",

"mobile": "15913215421",

"gender": 1,

"tel": "62394",

"email": "zhangsan@gzdev.com",

"weixinid": "lisifordev",

"avatar": ["http://wx.qlogo.cn/mmopen/ajNVdqHZLLA3WJ6DSZUfiakYe37PKnQhBIeOQBO4czqrnZDS79FH5Wm5m4X69TBicnHFlhiafvDwklOpZeXYQQ2icg/0"](http://wx.qlogo.cn/mmopen/ajNVdqHZLLA3WJ6DSZUfiakYe37PKnQhBIeOQBO4czqrnZDS79FH5Wm5m4X69TBicnHFlhiafvDwklOpZeXYQQ2icg/0),

"status": 1

}

3）部门成员的获取定义如下：

* 请求说明

Https请求方式: GET

<https://qyapi.weixin.qq.com/cgi-bin/user/simplelist?access_token=ACCESS_TOKEN&department_id=1&fetch_child=0&status=0>

* 参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| department\_id | 是 | 获取的部门id |
| fetch\_child | 否 | 1/0：是否递归获取子部门下面的成员 |
| status | 否 | 0获取全部员工，1获取已关注成员列表，2获取禁用成员列表，4获取未关注成员列表。status可叠加 |

* 权限说明

管理员须拥有’获取部门成员’的接口权限，以及指定部门的查看权限。

* 返回结果

{

"errcode": 0,

"errmsg": "ok",

"userlist": [

{

"userid": "zhangsan",

"name": "李四"

}

]

}

这个返回值我们定义一个实体对象用来存储数据即可。

[复制代码](javascript:void(0);)

/// <summary>

/// 获取部门成员返回的数据

/// </summary>

public class CorpUserListJson : BaseJsonResult

{

public CorpUserListJson()

{

this.userlist = new List<CorpUserSimpleJson>();

}

/// <summary>

/// 返回的错误消息

/// </summary>

public CorpReturnCode errcode { get; set; }

/// <summary>

/// 对返回码的文本描述内容

/// </summary>

public string errmsg { get; set; }

/// <summary>

/// 成员列表

/// </summary>

public List<CorpUserSimpleJson> userlist { get; set; }

}

[复制代码](javascript:void(0);)

**7、综合例子调用代码**

上面介绍了一些企业号的接口定义和我对API的C#封装接口和部分实现代码，实现了功能后，我们就可以在代码中对它进行测试，确信是否正常使用。

[复制代码](javascript:void(0);)

/// <summary>

/// 人员管理综合性操作（创建、修改、获取信息、删除）

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnCorpUser\_Click(object sender, EventArgs e)

{

CorpUserJson user = new CorpUserJson();

user.userid = "test";

user.name ="测试用户";

user.department = new List<int>(){2};

user.email = "test@163.com";

ICorpAddressBookApi bll = new CorpAddressBookApi();

CommonResult result = bll.CreateUser(token, user);

if (result != null)

{

Console.WriteLine("创建成员:{0} {1} {2}", user.name, (result.Success ? "成功" : "失败"), result.ErrorMessage);

string name = "修改测试";

user.name = name;

CorpUserUpdateJson userUpdate = new CorpUserUpdateJson(user);

result = bll.UpdateUser(token, userUpdate);

if (result != null)

{

Console.WriteLine("修改名称:{0} {1} {2}", name, (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

CorpUserGetJson userGet = bll.GetUser(token, user.userid);

if (userGet != null)

{

Console.WriteLine("成员名称:{0} ({1} {2})", userGet.name, user.userid, user.email);

}

result = bll.DeleteUser(token, user.userid);

if (result != null)

{

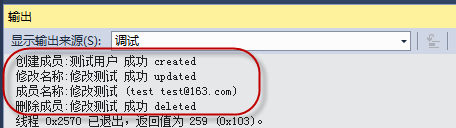
Console.WriteLine("删除成员:{0} {1} {2}", name, (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

}

}

[复制代码](javascript:void(0);)



获取部门人员的操作代码如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 获取部门人员

/// </summary>

private void btnCorpUserList\_Click(object sender, EventArgs e)

{

int deptId = 1;

ICorpAddressBookApi bll = new CorpAddressBookApi();

CorpUserListJson result = bll.GetDeptUser(token, deptId);

if (result != null)

{

foreach(CorpUserSimpleJson item in result.userlist)

{

Console.WriteLine("成员名称:{0} {1}", item.name, item.userid);

}

}

}

[复制代码](javascript:void(0);)

人员的管理，相对来说比较简单，主要是在一定的部门下创建人员，然后也可以给标签增加相应的人员，基本上就是这些了，不过一定需要确保有相应的权限进行操作。