**[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)**

 前面一篇随笔企业号的一些基础信息，以及介绍如何配置企业号的回调方式实现和企业号服务器进行沟通的桥梁。本篇主要还是继续介绍企业号的开发工作的开展，介绍微信企业号通讯录管理开发功能，介绍其中组织机构里面如何获取和管理部门的信息等内容。

**1、企业组织的创建和配置**

首先我们可以在企业号的管理后台里面创建一个组织机构，里面创建一些部门和人员列表，方便我们开发和使用。

例如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。

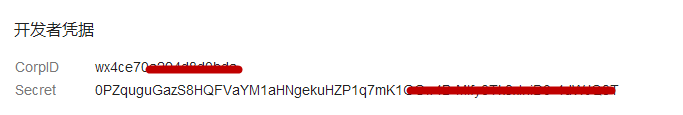


然后给组织结构根节点“广州爱奇迪”增加一个管理员权限，以后再开发接口里面也就可以使用这个管理员所属的权限Secret值进行调用了。



CorpID是企业号的标识，每个企业号拥有一个唯一的CorpID；Secret是管理组凭证密钥。  
系统管理员可通过管理端的权限管理功能创建管理组，分配管理组对应用、通讯录、接口的访问权限。完成后，管理组即可获得唯一的secret。系统管理员可通过权限管理查看所有管理组的secret，其他管理员可通过设置中的开发者凭据查看。

我的企业号的创建者和“广州爱奇迪”组织结构的管理员是不同的，由于Secret是管理组凭证密钥，因此管理者负责不同的组织机构管理的话，自己的管理Secret值可能就不同了。如果我们需要调用接口，就需要用到这个属于自己权限级别的Secret值，如下图所示。



 如果不是企业号的创建者，那么可能不能修改里面的一些权限分配，只能查看。



**2、API访问的全局唯一票据AccessToken的获取**

和公众号一样，我们调用企业号API的第一步也是需要先获取访问的票据AccessToken。这个票据是全局性的，有一定的时效和频率控制，因此需要适当的进行缓存，不能每次调用都去刷新获取。

企业号获取访问票据的主要的逻辑代码如下所示，其主要的就是需要使用管理者的Secret值去获取对应的口令，这样它就能够知道管理的是那个组织结构的了。

[复制代码](javascript:void(0);)

/// <summary>

/// 获取每次操作微信API的Token访问令牌

/// </summary>

/// <param name="corpid">企业Id</param>

/// <param name="corpsecret">管理组的凭证密钥</param>

/// <returns></returns>

public string GetAccessTokenNoCache(string corpid, string corpsecret)

{

var url = string.Format("https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid={0}&corpsecret={1}",

corpid, corpsecret);

HttpHelper helper = new HttpHelper();

string result = helper.GetHtml(url);

string regex = "\"access\_token\":\"(?<token>.\*?)\"";

string token = CRegex.GetText(result, regex, "token");

return token;

}

[复制代码](javascript:void(0);)

微信企业号的说明如下所示：

*当企业应用调用企业号接口时，企业号后台为根据此次访问的AccessToken,校验访问的合法性以及所对应的管理组的管理权限以返回相应的结果。*

***注：****你应该审慎配置管理组的权限，够用即好，权限过大会增加误操作可能性及信息安全隐患。*

AccessToken是企业号的全局唯一票据，调用接口时需携带AccessToken。AccessToken需要用[CorpID](http://qydev.weixin.qq.com/wiki/index.php?title=CorpID" \o "CorpID)和[Secret](http://qydev.weixin.qq.com/wiki/index.php?title=Secret" \o "Secret)来换取，不同的Secret会返回不同的AccessToken。**正常情况下AccessToken有效期为7200秒，有效期内重复获取返回相同结果，并自动续期。由于获取access\_token的api调用次数非常有限，建议企业全局存储与更新access\_token，频繁刷新access\_token会导致api调用受限，影响自身业务**。

**2、通讯录管理之部门信息的维护**

有了第一节里面的访问票据，我们就可以利用API来做很多事情了，包括组织结构的获取、创建、删除等等功能。

创建部门的官方接口定义如下所示。

* 请求说明

Https请求方式: POST

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

请求包结构体为:

{

"name": "邮箱产品组",

"parentid": "1"

}

* 参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| name | 是 | 部门名称。长度限制为1~64个字符 |
| parentid | 是 | 父亲部门id。根部门id为1 |

* 返回结果

{

"errcode": 0,

"errmsg": "created",

"id": 2

}

根据上面的一些类似的接口定义说明，我们先来定义下组织机构部门数据的维护接口，然后在逐步实现和调用。

[复制代码](javascript:void(0);)

#region 部门管理

/// <summary>

/// 创建部门。

/// 管理员须拥有“操作通讯录”的接口权限，以及父部门的管理权限。

/// </summary>

CorpDeptCreateJson CreateDept(string accessToken, string name, string parentId);

/// <summary>

/// 更新部门。

/// 管理员须拥有“操作通讯录”的接口权限，以及该部门的管理权限。

/// </summary>

CommonResult DeleteDept(string accessToken, int id);

/// <summary>

/// 删除部门.

/// 管理员须拥有“操作通讯录”的接口权限，以及该部门的管理权限。

/// </summary>

CorpDeptListJson ListDept(string accessToken);

/// <summary>

/// 获取部门列表.

/// 管理员须拥有’获取部门列表’的接口权限，以及对部门的查看权限。

/// </summary>

CommonResult UpdateDept(string accessToken, int id, string name);

#endregion

[复制代码](javascript:void(0);)

如创建部门的接口实现如下所示，主要就是构建URL和POST的数据包，然后统一调用并获取返回数据，转换为具体的Json对象实体即可。其他接口的实现方式类似，不在赘述。

[复制代码](javascript:void(0);)

/// <summary>

/// 创建部门。

/// 管理员须拥有“操作通讯录”的接口权限，以及父部门的管理权限。

/// </summary>

public CorpDeptCreateJson CreateDept(string accessToken, string name, string parentId)

{

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

var data = new

{

name = name,

parentId = parentId

};

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

var postData = data.ToJson();

CorpDeptCreateJson result = CorpJsonHelper<CorpDeptCreateJson>.ConvertJson(url, postData);

return result;

}

[复制代码](javascript:void(0);)

CorpDeptCreateJson 对象实体类的定义如下所示，我们主要是根据返回结果进行定义的。

[复制代码](javascript:void(0);)

/// <summary>

/// 创建部门的返回结果

/// </summary>

public class CorpDeptCreateJson : BaseJsonResult

{

/// <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>

/// 创建的部门id。

/// </summary>

public int id { get; set; }

}

[复制代码](javascript:void(0);)

**3、部门管理的API调用**

 上面小节介绍了如何封装部门管理的API，那么我们封装好了对应的接口和接口实现，怎么样在实际环境里面进行调用处理的呢，为了方便我创建一个小的Winform程序来测试对应API的功能，如下所示。



下面我们来介绍一下调用的代码和效果展示。

[复制代码](javascript:void(0);)

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

{

ICorpAddressBookApi bll = new CorpAddressBookApi();

string name = "测试部门";

CorpDeptCreateJson json = bll.CreateDept(token, name, "2");

if (json != null)

{

Console.WriteLine("创建了部门：{0}， ID:{1}", name, json.id);

//更新部门信息

name = "测试部门修改名称";

CommonResult result = bll.UpdateDept(token, json.id, name);

if(result != null)

{

Console.WriteLine("修改部门名称:{0} {1}", (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

//删除部门

result = bll.DeleteDept(token, json.id);

if (result != null)

{

Console.WriteLine("删除部门名称:{0} {1}", (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

}

}

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

/// <summary>

/// 获取部门列表

/// </summary>

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

{

ICorpAddressBookApi bll = new CorpAddressBookApi();

CorpDeptListJson list = bll.ListDept(token);

foreach (CorpDeptJson info in list.department)

{

string tips = string.Format("{0}:{1}", info.name, info.id);

Console.WriteLine(tips);

}

}

[复制代码](javascript:void(0);)

