**[C#开发微信门户及应用(16)-微信企业号的配置和使用](http://www.cnblogs.com/wuhuacong/p/3991735.html)**

 在本系列随笔的前面，主要就是介绍微信公众号的门户应用开发，最近把整个微信框架进行了扩展补充，增加了最新的企业号的API封装和开发，后续主要介绍如何利用C#进行微信企业号的开发工作，本篇作为微信企业号的开发的起步篇，介绍微信企业号的配置和使用。

**1、微信企业号的注册和登陆**

企业号是继公众号、订阅号的另外一种微信类型，它主要是面对企业的。企业号是微信为企业客户提供的移动应用入口。可以帮助企业建立员工、上下游供应链与企业 IT 系统间的连接。利用 企业号 ，企业或第三方合作伙伴可以帮助企业快速、低成本的实现高质量的移动轻应用，实现生产、管理、协作、运营的 移动化 。

个人觉得企业号最大的亮点是可以不限数量的消息发送，也就是可以在企业员工之间畅通交流。相对于公众号和订阅号，发送消息的谨慎程度，微信企业号可谓给人眼前一亮的感觉。不过微信企业号是需要内部建立好通讯录，关注者需要匹配通讯录的微信号、邮箱、电话号码任一个通过才可以关注，也就是可以防止其他外来人员的自由关注了，另外如果为了安全考虑，还可以设置二次验证，也就是一个审核过程。

企业号的认证和公众号一样，需要提供相关的企业资质文件，并且认证每年都要收取费用，否则可能有人员和功能的一些限制。觉得微信真是想着方法赚钱，目前已有的收费模式有，订阅号、公众号、企业号、开放平台，好像都有认证收费的了，而且微信小店也还需要收2万的押金，一切都是钱呀。

好了，其他不多说，微信的注册地址是：https://qy.weixin.qq.com，一个邮箱不能同时注册微信公众号和微信企业号。

对于企业开通企业号并开始使用需要四步

1) 企业到微信官网（ http://qy.weixin.qq.com ）申请开通；

2) 开通后，企业在企业号管理后台导入成员，发布二维码；

3) 企业调用企业号 api 与企业自有系统对接开发；

4) 员工关注，收到微信信息，在微信中与企业交互

注册好企业号，就可以通过微信扫一扫，扫描企业二维码进行登录了，扫描的时候，需要微信进行确认，才可以继续输入密码进行登录，操作界面如下所示（左边是手机截图，右边是网页截图）。

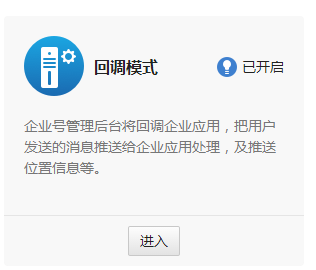
 

登录后我们就可以看到对应的电脑端的管理界面了。



**2、设置开发回调模式**

如果开发过微信公众号，那么我们就知道，如果需要在微信服务器和网站服务器之间建立连接关系，实现消息的转发和处理，那么就应该设置一个回调模式，需要配置好相关的参数。然后在自己 网站服务器里面建立一个处理微信服务器消息的入口。



进入配置后，我们需要修改相关的URL、Token、EncodingAESKey等参数，主要是URL，这个就是和公众号的入口处理一样的，需要我们发布到网站服务器上的处理入口。

Token和AESKey可以根据提示动态生成一个即可，AESKey好像必须是23位的，所以这个一般是让它自己生成的，这个主要用来加密解密使用的。

URL、Token、EncodingAESKey三个参数说明。

1）URL是企业应用接收企业号推送请求的访问协议和地址，支持http或https协议。

2）Token可由企业任意填写，用于生成签名。

3）EncodingAESKey用于消息体的加密，是AES密钥的Base64编码。

验证URL、Token以及加密的详细处理请参考后续 “接收消息时的加解密处理” 的部分。

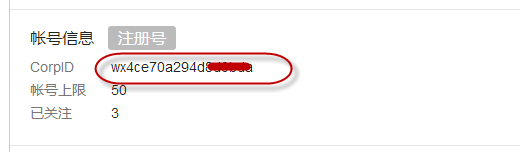


我公司的企业号配置后的界面如下所示。

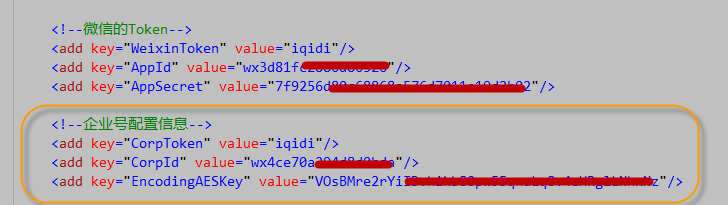


 这个URL里面指向的页面功能，需要对数据进行解析并返回给微信服务器，因此我们需要在服务器上预先部署好这个处理功能入口。

除了上面的几个函数，还有一个CorpID的参数需要使用，我们可以在后台主界面-设置里面查看到。



然后我们为了方便网站后台使用，我们和公众号的配置一样，把它放到了Web.Config里面，如下所示。



**3、实现回调页面的功能开发**

前面介绍了几个配置项，需要在回调页面里面使用的，本小节继续介绍如何实现企业号信息的回发，使之通过回调测试的操作。

由于回调测试的数据是通过Get方式发送的，因此我们的处理逻辑代码如下所示，和公众号的类似处理，只是实现部分不太一样而已。

[复制代码](javascript:void(0);)

/// <summary>

/// 企业号回调信息接口。统一接收并处理信息的入口。

/// </summary>

public class corpapi : IHttpHandler

{

/// <summary>

/// 处理企业号的信息

/// </summary>

/// <param name="context"></param>

public void ProcessRequest(HttpContext context)

{

string postString = string.Empty;

if (HttpContext.Current.Request.HttpMethod.ToUpper() == "POST")

{

using (Stream stream = HttpContext.Current.Request.InputStream)

{

Byte[] postBytes = new Byte[stream.Length];

stream.Read(postBytes, 0, (Int32)stream.Length);

postString = Encoding.UTF8.GetString(postBytes);

}

if (!string.IsNullOrEmpty(postString))

{

Execute(postString);

}

}

else

{

Auth();

}

}

/// <summary>

/// 成为开发者的第一步，验证并相应服务器的数据

/// </summary>

private void Auth()

{

#region 获取关键参数

string token = ConfigurationManager.AppSettings["CorpToken"];//从配置文件获取Token

if (string.IsNullOrEmpty(token))

{

LogTextHelper.Error(string.Format("CorpToken 配置项没有配置！"));

}

string encodingAESKey = ConfigurationManager.AppSettings["EncodingAESKey"];//从配置文件获取EncodingAESKey

if (string.IsNullOrEmpty(encodingAESKey))

{

LogTextHelper.Error(string.Format("EncodingAESKey 配置项没有配置！"));

}

string corpId = ConfigurationManager.AppSettings["CorpId"];//从配置文件获取corpId

if (string.IsNullOrEmpty(corpId))

{

LogTextHelper.Error(string.Format("CorpId 配置项没有配置！"));

}

#endregion

string echoString = HttpContext.Current.Request.QueryString["echoStr"];

string signature = HttpContext.Current.Request.QueryString["msg\_signature"];//企业号的 msg\_signature

string timestamp = HttpContext.Current.Request.QueryString["timestamp"];

string nonce = HttpContext.Current.Request.QueryString["nonce"];

string decryptEchoString = "";

if (new CorpBasicApi().CheckSignature(token, signature, timestamp, nonce, corpId, encodingAESKey, echoString, ref decryptEchoString))

{

if (!string.IsNullOrEmpty(decryptEchoString))

{

HttpContext.Current.Response.Write(decryptEchoString);

HttpContext.Current.Response.End();

}

}

}

[复制代码](javascript:void(0);)

具体的处理代码如下所示，里面的一个加解密处理的类是微信企业号附录里面提供的，我使用了C#版本的SDK而已。

[复制代码](javascript:void(0);)

/// <summary>

/// 企业号基础操作API实现

/// </summary>

public class CorpBasicApi : ICorpBasicApi

{

/// <summary>

/// 验证企业号签名

/// </summary>

/// <param name="token">企业号配置的Token</param>

/// <param name="signature">签名内容</param>

/// <param name="timestamp">时间戳</param>

/// <param name="nonce">nonce参数</param>

/// <param name="corpId">企业号ID标识</param>

/// <param name="encodingAESKey">加密键</param>

/// <param name="echostr">内容字符串</param>

/// <param name="retEchostr">返回的字符串</param>

/// <returns></returns>

public bool CheckSignature(string token, string signature, string timestamp, string nonce, string corpId, string encodingAESKey, string echostr, ref string retEchostr)

{

WXBizMsgCrypt wxcpt = new WXBizMsgCrypt(token, encodingAESKey, corpId);

int result = wxcpt.VerifyURL(signature, timestamp, nonce, echostr, ref retEchostr);

if (result != 0)

{

LogTextHelper.Error("ERR: VerifyURL fail, ret: " + result);

return false;

}

return true;

//ret==0表示验证成功，retEchostr参数表示明文，用户需要将retEchostr作为get请求的返回参数，返回给企业号。

// HttpUtils.SetResponse(retEchostr);

}