**[C#开发微信门户及应用(19)-微信企业号的消息发送（文本、图片、文件、语音、视频、图文消息等）](http://www.cnblogs.com/wuhuacong/p/3995494.html)**

 我们知道，企业号主要是面向企业需求而生的，因此内部消息的交流显得非常重要，而且发送、回复消息数量应该很可观，对于大企业尤其如此，因此可以结合企业号实现内部消息的交流。企业号具有关注安全、消息无限制等特点，很适合企业内部的环境。本文主要介绍如何利用企业号实现文本、图片、文件、语音、视频、图文消息等消息的发送操作。

**1、企业号特点**

对于企业号，有以下一些特点：

1）关注更安全

–只有企业通讯录的成员才能关注企业号，分级管理员、保密消息等各种特性确保企业内部信息的安全。

企业可以设置自行验证关注者身份，进行二次安全验证，保证企业信息使用和传递安全。

若员工离职，企业管理员可在通讯录中删除该成员，该成员即自动取消关注企业号，同时微信中的企业号历史记录也会被清除。

2）应用可配置

–企业可自行在企业号中可配置多个服务号，可以连接不同的企业应用系统，只有授权的企业成员才能使用相应的服务号。

3）消息无限制

–发送消息无限制，并提供完善的的管理接口及微信原生能力，以适应企业复杂、个性化的应用场景。

企业可以主动发消息给员工，**消息量不受限制**。

4）使用更便捷

–企业号在微信中有统一的消息入口，用户可以更方便地管理企业号消息。微信通讯录也可以直接访问企业号中的应用。

**2、企业号的管理接口内容**

目前企业号的内容可以用下面的分层图来展示，分别包含素材管理、被动响应消息、通讯录管理、自定义菜单等内容，详细可以看下面图示。

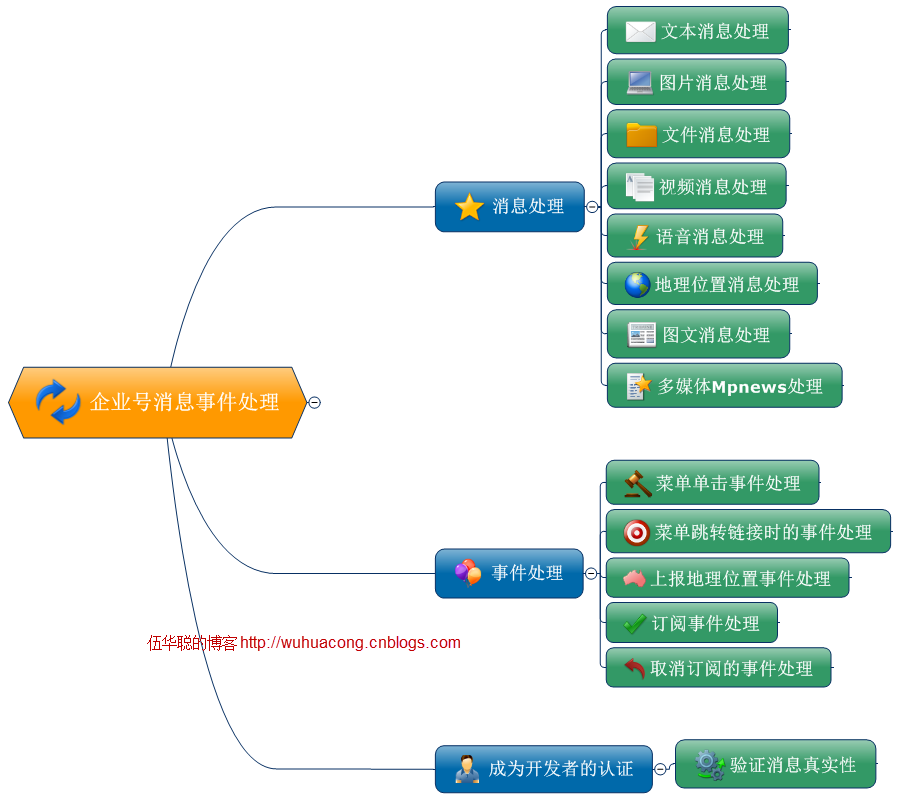


**3、企业号消息和事件的处理**

企业号和公众号一样，可以分为消息处理和事件处理，下面是他们两种类型的处理操作，也就发送的消息有文本消息、图片消息、文件消息、视频消息、语音消息、地理文字消息、图文和多媒体消息等。

事件处理主要就是关注、取消关注事件，以及菜单click类型和view类型两种操作，还有就是地理位置上报事件等。

两种类型的处理图如下所示。



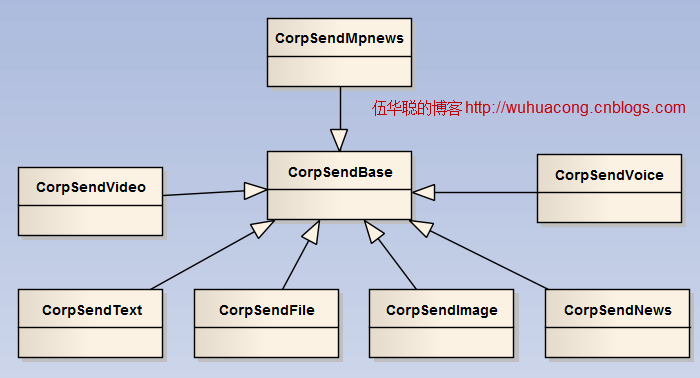
**4、企业号消息管理**

在企业的管理后台，和公众号一样，可以看到对应信息交流记录，包括文字、图片、地理位置等等，如下所示。



由于消息分为几种类型，包括文本(Text)、图片(Image)、文件(File)、语音(Voice)、视频(Video)、图文消息等(News)、MpNews等。

因此我们需要分别对它们进行一定的定义和封装处理，如下是它们的信息对象设计图。



企业号发送消息的官方定义如下:

企业可以主动发消息给员工，**消息量不受限制**。

调用接口时，使用Https协议、JSON数据包格式，数据包不需做加密处理。

目前支持文本、图片、语音、视频、文件、图文等消息类型。除了news类型，其它类型的消息可在发送时加上保密选项，保密消息会被打上水印，并且只有接收者才能阅读。

我们以发送的文本消息为例进行说明，它的定义如下所示。

* text消息

{

"touser": "UserID1|UserID2|UserID3",

"toparty": " PartyID1 | PartyID2 ",

"totag": " TagID1 | TagID2 ",

"msgtype": "text",

"agentid": "1",

"text": {

"content": "Holiday Request For Pony([http://xxxxx](http://xxxxx/))"

},

"safe":"0"

}

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| touser | 否 | UserID列表（消息接收者，多个接收者用‘|’分隔）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | PartyID列表，多个接受者用‘|’分隔。当touser为@all时忽略本参数 |
| totag | 否 | TagID列表，多个接受者用‘|’分隔。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：text |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| content | 是 | 消息内容 |
| safe | 否 | 表示是否是保密消息，0表示否，1表示是，默认0 |

 其中每种消息都会包含以下消息所示，也就是它们共同的属性：

touser": "UserID1|UserID2|UserID3",

"toparty": " PartyID1 | PartyID2 ",

"totag": " TagID1 | TagID2 ",

"msgtype": "text",

"agentid": "1",

因此我们可以定义一个基类用来方便承载这些共同的信息。

[复制代码](javascript:void(0);)

/// <summary>

/// 企业号发送消息的基础消息内容

/// </summary>

public class CorpSendBase

{

/// <summary>

/// UserID列表（消息接收者，多个接收者用‘|’分隔）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送

/// </summary>

public string touser { get; set; }

/// <summary>

/// PartyID列表，多个接受者用‘|’分隔。当touser为@all时忽略本参数

/// </summary>

public string toparty { get; set; }

/// <summary>

/// TagID列表，多个接受者用‘|’分隔。当touser为@all时忽略本参数

/// </summary>

public string totag { get; set; }

/// <summary>

/// 消息类型

/// </summary>

public string msgtype { get; set; }

/// <summary>

/// 企业应用的id，整型。可在应用的设置页面查看

/// </summary>

public string agentid { get; set; }

/// <summary>

/// 表示是否是保密消息，0表示否，1表示是，默认0

/// </summary>

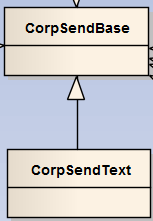
[JsonProperty(NullValueHandling = NullValueHandling.Ignore)]

public string safe { get; set; }

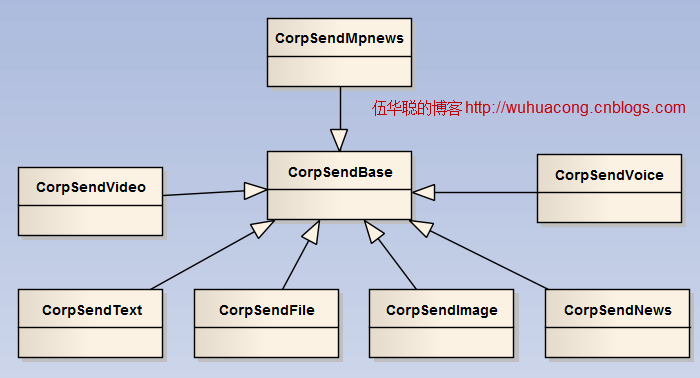
}

[复制代码](javascript:void(0);)

然后其他消息逐一继承这个基类即可，如下所示。



最终会构成下面这个继承关系图。



**5、消息接口的定义和实现**

 定义好相关的发送对象后，我们就可以定义它的统一发送接口了，如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 企业号消息管理接口定义

/// </summary>

public interface ICorpMessageApi

{

/// <summary>

/// 发送消息。

/// 需要管理员对应用有使用权限，对收件人touser、toparty、totag有查看权限，否则本次调用失败。

/// </summary>

/// <param name="accessToken"></param>

/// <returns></returns>

CommonResult SendMessage(string accessToken, CorpSendBase data);

}

[复制代码](javascript:void(0);)

最终，文本等类型的消息会根据接口定义进行实现，实现代码如下所示。注意，发送过程**不需要**调用加密类进行加密。

[复制代码](javascript:void(0);)

/// <summary>

/// 企业号消息管理实现类

/// </summary>

public class CorpMessageApi : ICorpMessageApi

{

/// <summary>

/// 发送消息。

/// 需要管理员对应用有使用权限，对收件人touser、toparty、totag有查看权限，否则本次调用失败。

/// </summary>

/// <param name="accessToken"></param>

/// <returns></returns>

public CommonResult SendMessage(string accessToken, CorpSendBase data)

{

CommonResult result = new CommonResult();

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

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

var postData = data.ToJson();

//数据不用加密发送

CorpSendResult sendResult = CorpJsonHelper<CorpSendResult>.ConvertJson(url, postData);

if (sendResult != null)

{

result.Success = (sendResult.errcode == CorpReturnCode.请求成功);

result.ErrorMessage = string.Format("invaliduser:{0},invalidparty:{1},invalidtag:{2}",

sendResult.invaliduser, sendResult.invalidparty, sendResult.invalidtag);

}

return result;

}

}

[复制代码](javascript:void(0);)

**6、消息的发送操作和实际效果**

 定义好相应的发送对象后，我们就可以进行统一的消息发送操作，包括文本、图片、文件、语音等等类型的消息，注意有些消息是需要上传到服务器上，然后在根据mediaId进行发送出去的。

发送文本和图片的操作代码如下所示。

[复制代码](javascript:void(0);)

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

{

//发送文本内容

ICorpMessageApi bll = new CorpMessageApi();

CorpSendText text = new CorpSendText("API 中文测试(http://www.iqidi.com)");

text.touser = "wuhuacong";

text.toparty = "4";//部门ID

text.totag = "0";

text.safe = "0";

text.agentid = "0";

CommonResult result = bll.SendMessage(token, text);

if (result != null)

{

Console.WriteLine("发送消息:{0} {1} {2}", text.text.content, (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

}

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

{

btnUpload\_Click(sender, e);

if (!string.IsNullOrEmpty(image\_mediaId))

{

//发送图片内容

ICorpMessageApi bll = new CorpMessageApi();

CorpSendImage image = new CorpSendImage(image\_mediaId);

CommonResult result = bll.SendMessage(token, image);

if (result != null)

{

Console.WriteLine("发送图片消息:{0} {1} {2}", image\_mediaId, (result.Success ? "成功" : "失败"), result.ErrorMessage);

}

}

}

[复制代码](javascript:void(0);)

最后在微信企业号上截图效果如下所示，包括了文本测试、文件测试、图文测试、语音测试均正常。