# 微信公众平台开发

## 微信平台开发可以大致分成三种

### 主动发送请求

说明：主动发送请求是指根据微信所提供的请求地址，直接发送请求到请求地址达到某项功能，一般来说公众用户通过自己的AppID(应用ID)与AppSecret(应用密钥)，发送请求到请求token的请求地址，获取token后再请求其他功能。

#### 请求内容说明例子

首先我们发送一个请求到获取token的地址：

**接口调用请求说明**

http请求方式: GET

<https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET>

将AppID(应用ID)与AppSecret(应用密钥)作为参数送到该请求地址后，就会获得一个json字符串的回应：

{"access\_token":"ACCESS\_TOKEN","expires\_in":7200}

将access\_token保存下来，

**然后对请求菜单地址：**

**接口调用请求说明**

http请求方式：POST（请使用https协议） <https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN>

发送内容为：

{

"button":[

{

"type":"click",

"name":"今日歌曲",

"key":"V1001\_TODAY\_MUSIC"

}]

}

在例程里面的request.html内有例子展示如何定义一个菜单的例子。

所有主动请求的方法都可以通过微信提供的API文档并且通过模仿例程的方法来使用。

通过主动请求而调用的方法大致有：

获取微信服务器IP地址、上传下载多媒体文件、上传图文消息素材【订阅号与服务号认证后均可用】、根据分组进行群发【订阅号与服务号认证后均可用】、根据OpenID列表群发【订阅号不可用，服务号认证后可用】、删除群发【订阅号与服务号认证后均可用】、 预览接口【订阅号与服务号认证后均可用】、查询群发消息发送状态【订阅号与服务号认证后均可用】、事件推送群发结果、创建分组、查询所有分组、 查询用户所在分组、修改分组名、移动用户分组、生成带参数的二维码等功能。具体列表应该参照引用文档中的微信接口说明。

### 被动接收推送

被动推动指用户在关注/取消关注、发送各种信息、点击公众号菜单、自动上传地理位置的时候发送的推送信息。

在打开微信公众号的开发者中心并且认证服务地址后，微信服务器会把上述的信息推送到我们的认证地址上。原来的发送接收方式也会改变成由微信服务器将公众号收到的信息转发给我们制定的服务器。然后由指定服务器的回答会被转发到用户。



那么到底如何去部署这个服务，并且如何去响应呢。

微信api文档里用的是php的例子，但是只要上网找一找也会发现不少.net的例子。但是无论是用php、.net还是使用nodejs搭建服务，其原理都是一样的，就是接受微信服务器发送的请求，并且发送相应格式的应答。

##### 接入服务

我们要自己部署服务，首先要指定服务地址，拿点NET为例，我们可以部署一个ashx程序，在微信公众号内的服务器地址中填上我们的ashx服务地址，请注意服务一定要部署在外网并且要保证80端口可用。

一开始微信会发送一个认证请求

开发者提交信息后，微信服务器将发送GET请求到填写的URL上，GET请求携带四个参数：

|  |  |
| --- | --- |
| **参数** | **描述** |
| signature | 微信加密签名，signature结合了开发者填写的token参数和请求中的timestamp参数、nonce参数。 |
| timestamp | 时间戳 |
| nonce | 随机数 |
| echostr | 随机字符串 |

最简单的做法可以直接原样返回echostr就可以完成验证，完成验证后就可以把服务文件内的东西替换成我们自定义接受的服务。

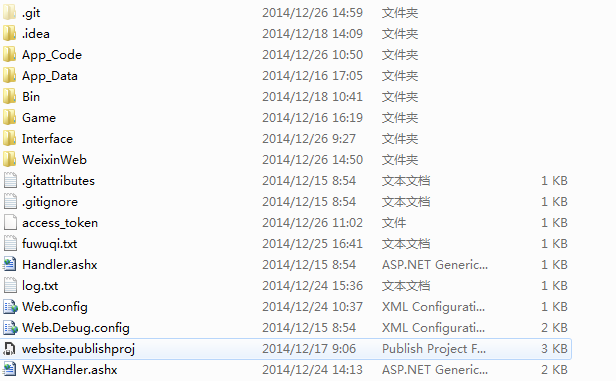
微信中一系列的操作如：关注，发送信息到公众号，点击公众号的菜单等操作，都会发送请求到我们的服务，我们根据参数判断请求类型然后返回相应格式的应答就能做到自定义回答。具体判断与应答类型可以通过引用文档去找到。

### WeixinJSBridge脚本接口

这个是提供给微网站的接口，可以调用如朋友圈分享等提供的功能。具体是只要你部署的网站通过微信打开，就可以使用相关的接口。接口具体的使用请参考引用文档。

## 程序实战

为了更好理解微信公众平台开发，我们特意开发了一套模拟的系统来支撑。是基于framework4.5后台的站点。获取文件后将



部署到一个可以使用的开发网站下，第一次应把access\_token文件删除，然后打开web.config

配置好

<add key="appid" value=""/>

<add key="secret" value=""/>

这两个参数，这两个参数是来自微信公众号开发者接口里面的，每个公众号都有自己的appid与sercet密码。

### 认证开发者

以我们的例程为例子，我们最终发布的处理文件的入口在WXHandler.ashx,所以我们在



中应该把URL填写到WXHandler.ashx的外网路径，并保证能访问。

Token的值随意填写，填写后在WXCheck.CheckSignature方法中必须让里面的token变量值与这里填写的值一致。

目前例程中的的认证代码是注释掉的，因为认证只在设置时候有用，认证过后正式发送接收信息后认证代码就没有作用了。所以我们认证的时候

把WXHandler.ashx内的认证用注释的代码开出来，然后把下面平时用的代码注释掉。



修改完成后部署到我们的服务器，然后就可以开始认证。

成功认证后就会提示成功。

**注意：在认证后原来的配置过的菜单，设定的自定义回复等东西都会消失。**

完成认证后，就把认证用代码注释，并且把平时用的代码开回来。

### 验证签名

我们最好在每次接受到信息的时候都进行一下签名认证，这样可以过滤无效的请求。

具体认证的过程已经写在WXCheck.CheckSignature中，可以在程序中查看，具体原理是：

**加密/校验流程如下：**

1. 将token、timestamp、nonce三个参数进行**字典序**排序

2. 将三个参数字符串拼接成一个字符串进行sha1加密

3. 开发者获得加密后的字符串可与signature对比，标识该请求来源于微信

我们可以看见例程中的WXHandler.ashx中每次接受请求都要先调用WXCheck.CheckSignature做判断。

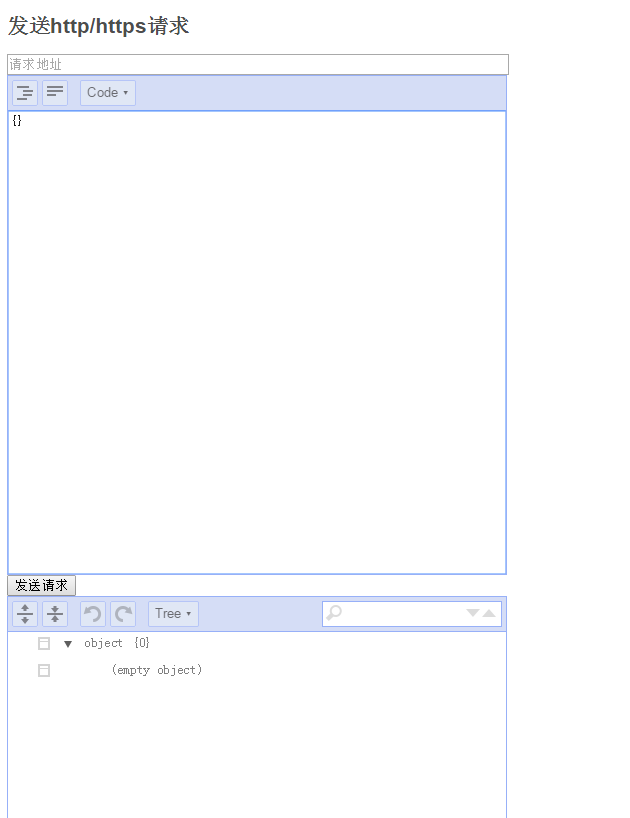
### 获取ACCESS\_TOKEN

对于请求类的接口，如何让请求知道这是哪个公众号发送的请求，access\_token就是解决这个问题的关键。我们通过公众号的appid与secret，通过请求返回access\_token，然后就可以通过它来请求到相应的公众号功能。

由于access\_token有时效性，并且获取access\_token的次数有限，所以我们必须在某个地方存储access\_token，在过期前尽量使用该access\_token，从而避免过多获取而导致无法再次获取access\_token。那么我们程序中的做法是，获取access\_token后在服务器中保存一个access\_token文件，在里面加入过期时间，每次再次获取时，先检查token是否过期，过期就重新获取，没过期就继续用。这么下来，一个月内就不可能超过使用次数了。相关获取的例子在WXCheck.GetAccessTokenString方法中。

### 请求发送

例程中WeixinWeb\request.html中有一个请求发送的快捷位置



输入请求地址如：<https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token=ACCESS_TOKEN>

系统便会发送请求，并且根据web.config中的配置自动替换ACCESS\_TOKEN字符串为公众号的合法access\_token，

然后会post下方json编辑器中的内容到服务地址中。可以完成如群发，自定义菜单等一系列功能。

### 接收消息

#### 接收普通消息

通过认证后，所有粉丝发送的东西都会转发到我们部署的WXHandler.ashx中，所以在代码中我们就可以处理这些接收的消息。我们可以接受

* [1 文本消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E6.96.87.E6.9C.AC.E6.B6.88.E6.81.AF)
* [2 图片消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E5.9B.BE.E7.89.87.E6.B6.88.E6.81.AF)
* [3 语音消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E8.AF.AD.E9.9F.B3.E6.B6.88.E6.81.AF)
* [4 视频消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E8.A7.86.E9.A2.91.E6.B6.88.E6.81.AF)
* [5 地理位置消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E5.9C.B0.E7.90.86.E4.BD.8D.E7.BD.AE.E6.B6.88.E6.81.AF)
* [6 链接消息](http://mp.weixin.qq.com/wiki/10/79502792eef98d6e0c6e1739da387346.html#.E9.93.BE.E6.8E.A5.E6.B6.88.E6.81.AF)

##### 文本消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1348831860</CreateTime>

<MsgType><![CDATA[text]]></MsgType>

<Content><![CDATA[this is a test]]></Content>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | text |
| Content | 文本消息内容 |
| MsgId | 消息id，64位整型 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E6%96%87%E6%9C%AC%E6%B6%88%E6%81%AF)

##### 图片消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1348831860</CreateTime>

<MsgType><![CDATA[image]]></MsgType>

<PicUrl><![CDATA[this is a url]]></PicUrl>

<MediaId><![CDATA[media\_id]]></MediaId>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | image |
| PicUrl | 图片链接 |
| MediaId | 图片消息媒体id，可以调用多媒体文件下载接口拉取数据。 |
| MsgId | 消息id，64位整型 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E5%9B%BE%E7%89%87%E6%B6%88%E6%81%AF)

##### 语音消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1357290913</CreateTime>

<MsgType><![CDATA[voice]]></MsgType>

<MediaId><![CDATA[media\_id]]></MediaId>

<Format><![CDATA[Format]]></Format>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 语音为voice |
| MediaId | 语音消息媒体id，可以调用多媒体文件下载接口拉取数据。 |
| Format | 语音格式，如amr，speex等 |
| MsgID | 消息id，64位整型 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E8%AF%AD%E9%9F%B3%E6%B6%88%E6%81%AF)

##### 视频消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1357290913</CreateTime>

<MsgType><![CDATA[video]]></MsgType>

<MediaId><![CDATA[media\_id]]></MediaId>

<ThumbMediaId><![CDATA[thumb\_media\_id]]></ThumbMediaId>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 视频为video |
| MediaId | 视频消息媒体id，可以调用多媒体文件下载接口拉取数据。 |
| ThumbMediaId | 视频消息缩略图的媒体id，可以调用多媒体文件下载接口拉取数据。 |
| MsgId | 消息id，64位整型 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E8%A7%86%E9%A2%91%E6%B6%88%E6%81%AF)

##### 地理位置消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1351776360</CreateTime>

<MsgType><![CDATA[location]]></MsgType>

<Location\_X>23.134521</Location\_X>

<Location\_Y>113.358803</Location\_Y>

<Scale>20</Scale>

<Label><![CDATA[位置信息]]></Label>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | location |
| Location\_X | 地理位置维度 |
| Location\_Y | 地理位置经度 |
| Scale | 地图缩放大小 |
| Label | 地理位置信息 |
| MsgId | 消息id，64位整型 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E5%9C%B0%E7%90%86%E4%BD%8D%E7%BD%AE%E6%B6%88%E6%81%AF)

##### 链接消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1351776360</CreateTime>

<MsgType><![CDATA[link]]></MsgType>

<Title><![CDATA[公众平台官网链接]]></Title>

<Description><![CDATA[公众平台官网链接]]></Description>

<Url><![CDATA[url]]></Url>

<MsgId>1234567890123456</MsgId>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | 接收方微信号 |
| FromUserName | 发送方微信号，若为普通用户，则是一个OpenID |
| CreateTime | 消息创建时间 |
| MsgType | 消息类型，link |
| Title | 消息标题 |
| Description | 消息描述 |
| Url | 消息链接 |
| MsgId | 消息id，64位整型 |

##### 例程说明

根据上面的发送形式，我们就可以通过解释request来判断我们接受的是何种消息，例程中的MessageHelper.cs文件中有接收消息的例子，其中的TextHandle函数就是处理环节就是展现如何处理接受文本消息的，通过判断参数中的MsgType类型来判断出类型然后做相应的处理。其他的类型也以此类推。

#### 接收事件推送消息

* [1 关注/取消关注事件](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E5.85.B3.E6.B3.A8.2F.E5.8F.96.E6.B6.88.E5.85.B3.E6.B3.A8.E4.BA.8B.E4.BB.B6)
* [2 扫描带参数二维码事件](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E6.89.AB.E6.8F.8F.E5.B8.A6.E5.8F.82.E6.95.B0.E4.BA.8C.E7.BB.B4.E7.A0.81.E4.BA.8B.E4.BB.B6)
* [3 上报地理位置事件](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E4.B8.8A.E6.8A.A5.E5.9C.B0.E7.90.86.E4.BD.8D.E7.BD.AE.E4.BA.8B.E4.BB.B6)
* [4 自定义菜单事件](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E8.87.AA.E5.AE.9A.E4.B9.89.E8.8F.9C.E5.8D.95.E4.BA.8B.E4.BB.B6)
* [5 点击菜单拉取消息时的事件推送](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E7.82.B9.E5.87.BB.E8.8F.9C.E5.8D.95.E6.8B.89.E5.8F.96.E6.B6.88.E6.81.AF.E6.97.B6.E7.9A.84.E4.BA.8B.E4.BB.B6.E6.8E.A8.E9.80.81)
* [6 点击菜单跳转链接时的事件推送](http://mp.weixin.qq.com/wiki/2/5baf56ce4947d35003b86a9805634b1e.html#.E7.82.B9.E5.87.BB.E8.8F.9C.E5.8D.95.E8.B7.B3.E8.BD.AC.E9.93.BE.E6.8E.A5.E6.97.B6.E7.9A.84.E4.BA.8B.E4.BB.B6.E6.8E.A8.E9.80.81)

##### 关注/取消关注事件

用户在关注与取消关注公众号时，微信会把这个事件推送到开发者填写的URL。方便开发者给用户下发欢迎消息或者做帐号的解绑。

**微信服务器在五秒内收不到响应会断掉连接，并且重新发起请求，总共重试三次**

关于重试的消息排重，推荐使用FromUserName + CreateTime 排重。

**假如服务器无法保证在五秒内处理并回复，可以直接回复空串，微信服务器不会对此作任何处理，并且不会发起重试。**

推送XML数据包示例：

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[FromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[subscribe]]></Event>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，subscribe(订阅)、unsubscribe(取消订阅) |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E4%BA%8B%E4%BB%B6%E6%B6%88%E6%81%AF)

##### 扫描带参数二维码事件

用户扫描带场景值二维码时，可能推送以下两种事件：

1. 如果用户还未关注公众号，则用户可以关注公众号，关注后微信会将带场景值关注事件推送给开发者。
2. 如果用户已经关注公众号，则微信会将带场景值扫描事件推送给开发者。

**1. 用户未关注时，进行关注后的事件推送**

推送XML数据包示例：

<xml><ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[FromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[subscribe]]></Event>

<EventKey><![CDATA[qrscene\_123123]]></EventKey>

<Ticket><![CDATA[TICKET]]></Ticket>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，subscribe |
| EventKey | 事件KEY值，qrscene\_为前缀，后面为二维码的参数值 |
| Ticket | 二维码的ticket，可用来换取二维码图片 |

**2. 用户已关注时的事件推送**

推送XML数据包示例：

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[FromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[SCAN]]></Event>

<EventKey><![CDATA[SCENE\_VALUE]]></EventKey>

<Ticket><![CDATA[TICKET]]></Ticket>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，SCAN |
| EventKey | 事件KEY值，是一个32位无符号整数，即创建二维码时的二维码scene\_id |
| Ticket | 二维码的ticket，可用来换取二维码图片 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E4%BA%8B%E4%BB%B6%E6%B6%88%E6%81%AF)

##### 上报地理位置事件

用户同意上报地理位置后，每次进入公众号会话时，都会在进入时上报地理位置，或在进入会话后每5秒上报一次地理位置，公众号可以在公众平台网站中修改以上设置。上报地理位置时，微信会将上报地理位置事件推送到开发者填写的URL。

推送XML数据包示例：

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[LOCATION]]></Event>

<Latitude>23.137466</Latitude>

<Longitude>113.352425</Longitude>

<Precision>119.385040</Precision>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，LOCATION |
| Latitude | 地理位置纬度 |
| Longitude | 地理位置经度 |
| Precision | 地理位置精度 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%E8%B0%83%E8%AF%95&form=%E4%BA%8B%E4%BB%B6%E6%B6%88%E6%81%AF)

##### 自定义菜单事件

用户点击自定义菜单后，微信会把点击事件推送给开发者，请注意，点击菜单弹出子菜单，不会产生上报。

##### 点击菜单拉取消息时的事件推送

推送XML数据包示例：

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[FromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[CLICK]]></Event>

<EventKey><![CDATA[EVENTKEY]]></EventKey>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，CLICK |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |

##### 点击菜单跳转链接时的事件推送

推送XML数据包示例：

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[FromUser]]></FromUserName>

<CreateTime>123456789</CreateTime>

<MsgType><![CDATA[event]]></MsgType>

<Event><![CDATA[VIEW]]></Event>

<EventKey><![CDATA[www.qq.com]]></EventKey>

</xml>

参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息类型，event |
| Event | 事件类型，VIEW |
| EventKey | 事件KEY值，设置的跳转URL |

##### 例程说明

这里与上面普通消息的接收其实差别不大，也是通过MsgType来判断，如果是event就表示是事件推送消息，不过事件要通过进一步判断Event参数来判断到底触发的是何种事件，同样例程里面有一个MessageHelper.cs 内的EventHandle来举例如何处理菜单Click事件和关注及取消关注时触发的事件。

### 发送消息

#### 回复客户发送消息

|  |
| --- |
| * [1 回复文本消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E6.96.87.E6.9C.AC.E6.B6.88.E6.81.AF) * [2 回复图片消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E5.9B.BE.E7.89.87.E6.B6.88.E6.81.AF) * [3 回复语音消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E8.AF.AD.E9.9F.B3.E6.B6.88.E6.81.AF) * [4 回复视频消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E8.A7.86.E9.A2.91.E6.B6.88.E6.81.AF) * [5 回复音乐消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E9.9F.B3.E4.B9.90.E6.B6.88.E6.81.AF) * [6 回复图文消息](http://mp.weixin.qq.com/wiki/9/2c15b20a16019ae613d413e30cac8ea1.html#.E5.9B.9E.E5.A4.8D.E5.9B.BE.E6.96.87.E6.B6.88.E6.81.AF) |

##### 回复文本消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[text]]></MsgType>

<Content><![CDATA[你好]]></Content>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **描述** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间 （整型） |
| MsgType | 是 | text |
| Content | 是 | 回复的消息内容（换行：在content中能够换行，微信客户端就支持换行显示） |

##### 回复图片消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[image]]></MsgType>

<Image>

<MediaId><![CDATA[media\_id]]></MediaId>

</Image>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间 （整型） |
| MsgType | 是 | image |
| MediaId | 是 | 通过上传多媒体文件，得到的id。 |

##### 回复语音消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[voice]]></MsgType>

<Voice>

<MediaId><![CDATA[media\_id]]></MediaId>

</Voice>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间戳 （整型） |
| MsgType | 是 | 语音，voice |
| MediaId | 是 | 通过上传多媒体文件，得到的id |

##### 回复视频消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[video]]></MsgType>

<Video>

<MediaId><![CDATA[media\_id]]></MediaId>

<Title><![CDATA[title]]></Title>

<Description><![CDATA[description]]></Description>

</Video>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间 （整型） |
| MsgType | 是 | video |
| MediaId | 是 | 通过上传多媒体文件，得到的id |
| Title | 否 | 视频消息的标题 |
| Description | 否 | 视频消息的描述 |

##### 回复音乐消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[music]]></MsgType>

<Music>

<Title><![CDATA[TITLE]]></Title>

<Description><![CDATA[DESCRIPTION]]></Description>

<MusicUrl><![CDATA[MUSIC\_Url]]></MusicUrl>

<HQMusicUrl><![CDATA[HQ\_MUSIC\_Url]]></HQMusicUrl>

<ThumbMediaId><![CDATA[media\_id]]></ThumbMediaId>

</Music>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间 （整型） |
| MsgType | 是 | music |
| Title | 否 | 音乐标题 |
| Description | 否 | 音乐描述 |
| MusicURL | 否 | 音乐链接 |
| HQMusicUrl | 否 | 高质量音乐链接，WIFI环境优先使用该链接播放音乐 |
| ThumbMediaId | 是 | 缩略图的媒体id，通过上传多媒体文件，得到的id |

##### 回复图文消息

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>12345678</CreateTime>

<MsgType><![CDATA[news]]></MsgType>

<ArticleCount>2</ArticleCount>

<Articles>

<item>

<Title><![CDATA[title1]]></Title>

<Description><![CDATA[description1]]></Description>

<PicUrl><![CDATA[picurl]]></PicUrl>

<Url><![CDATA[url]]></Url>

</item>

<item>

<Title><![CDATA[title]]></Title>

<Description><![CDATA[description]]></Description>

<PicUrl><![CDATA[picurl]]></PicUrl>

<Url><![CDATA[url]]></Url>

</item>

</Articles>

</xml>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| ToUserName | 是 | 接收方帐号（收到的OpenID） |
| FromUserName | 是 | **开发者**微信号 |
| CreateTime | 是 | 消息创建时间 （整型） |
| MsgType | 是 | news |
| ArticleCount | 是 | 图文消息个数，限制为10条以内 |
| Articles | 是 | 多条图文消息信息，默认第一个item为大图,注意，如果图文数超过10，则将会无响应 |
| Title | 否 | 图文消息标题 |
| Description | 否 | 图文消息描述 |
| PicUrl | 否 | 图片链接，支持JPG、PNG格式，较好的效果为大图360\*200，小图200\*200 |
| Url | 否 | 点击图文消息跳转链接 |

##### 例程说明

回复消息的时候必须要按照格式发送response，在ReplyType类中定义了若干格式模板，利用C# string.foramt函数可以比较轻松的调用这些模板。并在TextHandle内示意了如何返回一个应答的字符串供response使用。

#### 发送客服消息

前面说到的基本是属于被动接收推送类的开发。发送客服消息，就是主动发送请求的类别。主动发送请求的方法甚至不用编程，在浏览器的地址栏中输入请求地址也能完成。但是由于很多请求还需要post值，所以还是需要有一定的HTML基础才能完成。

##### 接口调用请求说明

当用户主动发消息给公众号的时候（包括发送信息、点击自定义菜单、订阅事件、扫描二维码事件、支付成功事件、用户维权），微信将会把消息数据推送给开发者，开发者在一段时间内（目前修改为48小时）可以调用客服消息接口，通过POST一个JSON数据包来发送消息给普通用户，在48小时内不限制发送次数。此接口主要用于客服等有人工消息处理环节的功能，方便开发者为用户提供更加优质的服务。

http请求方式: POST

<https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token=ACCESS_TOKEN>

|  |
| --- |
| * [1 发送文本消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E6.96.87.E6.9C.AC.E6.B6.88.E6.81.AF) * [2 发送图片消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E5.9B.BE.E7.89.87.E6.B6.88.E6.81.AF) * [3 发送语音消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E8.AF.AD.E9.9F.B3.E6.B6.88.E6.81.AF) * [4 发送视频消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E8.A7.86.E9.A2.91.E6.B6.88.E6.81.AF) * [5 发送音乐消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E9.9F.B3.E4.B9.90.E6.B6.88.E6.81.AF) * [6 发送图文消息](http://mp.weixin.qq.com/wiki/7/12a5a320ae96fecdf0e15cb06123de9f.html#.E5.8F.91.E9.80.81.E5.9B.BE.E6.96.87.E6.B6.88.E6.81.AF) |

##### 发送文本消息

{

"touser":"OPENID",

"msgtype":"text",

"text":

{

"content":"Hello World"

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，text |
| content | 是 | 文本消息内容 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E5%90%91%E7%94%A8%E6%88%B7%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF&form=%E5%8F%91%E9%80%81%E5%AE%A2%E6%9C%8D%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%20/message/custom/send)

##### 发送图片消息

{

"touser":"OPENID",

"msgtype":"image",

"image":

{

"media\_id":"MEDIA\_ID"

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，image |
| media\_id | 是 | 发送的图片的媒体ID |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E5%90%91%E7%94%A8%E6%88%B7%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF&form=%E5%8F%91%E9%80%81%E5%AE%A2%E6%9C%8D%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%20/message/custom/send)

##### 发送语音消息

{

"touser":"OPENID",

"msgtype":"voice",

"voice":

{

"media\_id":"MEDIA\_ID"

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，voice |
| media\_id | 是 | 发送的语音的媒体ID |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E5%90%91%E7%94%A8%E6%88%B7%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF&form=%E5%8F%91%E9%80%81%E5%AE%A2%E6%9C%8D%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%20/message/custom/send)

##### 发送视频消息

{

"touser":"OPENID",

"msgtype":"video",

"video":

{

"media\_id":"MEDIA\_ID",

"thumb\_media\_id":"MEDIA\_ID",

"title":"TITLE",

"description":"DESCRIPTION"

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，video |
| media\_id | 是 | 发送的视频的媒体ID |
| thumb\_media\_id | 是 | 缩略图的媒体ID |
| title | 否 | 视频消息的标题 |
| description | 否 | 视频消息的描述 |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E5%90%91%E7%94%A8%E6%88%B7%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF&form=%E5%8F%91%E9%80%81%E5%AE%A2%E6%9C%8D%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%20/message/custom/send)

##### 发送音乐消息

{

"touser":"OPENID",

"msgtype":"music",

"music":

{

"title":"MUSIC\_TITLE",

"description":"MUSIC\_DESCRIPTION",

"musicurl":"MUSIC\_URL",

"hqmusicurl":"HQ\_MUSIC\_URL",

"thumb\_media\_id":"THUMB\_MEDIA\_ID"

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，music |
| title | 否 | 音乐标题 |
| description | 否 | 音乐描述 |
| musicurl | 是 | 音乐链接 |
| hqmusicurl | 是 | 高品质音乐链接，wifi环境优先使用该链接播放音乐 |
| thumb\_media\_id | 是 | 缩略图的媒体ID |

[使用网页调试工具调试该接口](https://mp.weixin.qq.com/debug/cgi-bin/apiinfo?t=index&type=%E5%90%91%E7%94%A8%E6%88%B7%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF&form=%E5%8F%91%E9%80%81%E5%AE%A2%E6%9C%8D%E6%B6%88%E6%81%AF%E6%8E%A5%E5%8F%A3%20/message/custom/send)

##### 发送图文消息

图文消息条数限制在10条以内，注意，如果图文数超过10，则将会无响应。

{

"touser":"OPENID",

"msgtype":"news",

"news":{

"articles": [

{

"title":"Happy Day",

"description":"Is Really A Happy Day",

"url":"URL",

"picurl":"PIC\_URL"

},

{

"title":"Happy Day",

"description":"Is Really A Happy Day",

"url":"URL",

"picurl":"PIC\_URL"

}

]

}

}

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| access\_token | 是 | 调用接口凭证 |
| touser | 是 | 普通用户openid |
| msgtype | 是 | 消息类型，news |
| title | 否 | 标题 |
| description | 否 | 描述 |
| url | 否 | 点击后跳转的链接 |
| picurl | 否 | 图文消息的图片链接，支持JPG、PNG格式，较好的效果为大图640\*320，小图80\*80 |

#### 发送群发消息

与发送客服消息一样只是请求的地址不同，详细请参考引用文档中的发送群发消息。

### 自定义菜单

自定义菜单也是一种请求方式，

{

"button": [

{

"type": "view",

"name": "点餐",

"url": "http://112.124.112.166/Ordering/"

},

{

"name": "菜单",

"sub\_button": [

{

"type": "view",

"name": "WeixinJSBridge",

"url": "http://112.124.112.166/weixintest/WeixinWeb/WeixinJSBridge.html"

},

{

"type": "click",

"name": "抽奖活动",

"key": "HZGift"

},

{

"type": "click",

"name": "我的中奖记录",

"key": "HZGiftList"

}

]

}

]

}

这是一个珠海浩智的菜单请求内容，只需要使用请求发送例子在请求地址中填入

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

并在下方输入上面的json 就能创建出对应的菜单

如图所示



### 网页授权获取用户基本信息

我们在例程中部署的抽奖游戏，属于第三方应用，实际就是在微信内置浏览器中打开了我们的网页而已。

获取用户基本信息接口是在用户和公众号产生消息交互时，才能根据用户OpenID获取用户基本信息，而网页授权的方式获取用户基本信息，则无需消息交互，只是用户进入到公众号的网页，就可弹出请求用户授权的界面，用户授权后，就可获得其基本信息（此过程甚至不需要用户已经关注公众号。）

我们在实现自己的游戏时候，往往需要获取用户以完成一系列认证的工作。

获取的方式其实很简单，我们在跳转到自己页面的时候不要直接跳转，而是用以下的链接

<https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=SCOPE&state=STATE#wechat_redirect>

我们把我们的站点通过urlencode后作为参数添加到redirect\_url参数中，这样跳转的页面就会带上一个code，然后我们就可以通过页面参数中的code去获取登录用户的openid，从而获取用户信息。

* [1 第一步：用户同意授权，获取code](http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html#.E7.AC.AC.E4.B8.80.E6.AD.A5.EF.BC.9A.E7.94.A8.E6.88.B7.E5.90.8C.E6.84.8F.E6.8E.88.E6.9D.83.EF.BC.8C.E8.8E.B7.E5.8F.96code)
* [2 第二步：通过code换取网页授权access\_token](http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html#.E7.AC.AC.E4.BA.8C.E6.AD.A5.EF.BC.9A.E9.80.9A.E8.BF.87code.E6.8D.A2.E5.8F.96.E7.BD.91.E9.A1.B5.E6.8E.88.E6.9D.83access_token)
* [3 第三步：刷新access\_token（如果需要）](http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html#.E7.AC.AC.E4.B8.89.E6.AD.A5.EF.BC.9A.E5.88.B7.E6.96.B0access_token.EF.BC.88.E5.A6.82.E6.9E.9C.E9.9C.80.E8.A6.81.EF.BC.89)
* [4 第四步：拉取用户信息(需scope为 snsapi\_userinfo)](http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html#.E7.AC.AC.E5.9B.9B.E6.AD.A5.EF.BC.9A.E6.8B.89.E5.8F.96.E7.94.A8.E6.88.B7.E4.BF.A1.E6.81.AF.28.E9.9C.80scope.E4.B8.BA_snsapi_userinfo.29)
* [5 附：检验授权凭证（access\_token）是否有效](http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html#.E9.99.84.EF.BC.9A.E6.A3.80.E9.AA.8C.E6.8E.88.E6.9D.83.E5.87.AD.E8.AF.81.EF.BC.88access_token.EF.BC.89.E6.98.AF.E5.90.A6.E6.9C.89.E6.95.88)

共有5步，在我们的小游戏内就做了这个动作。我们先是通过后台方法WXCheck.GetOAuth2URL方法获取我们的验证URL然后通过客服消息和推送消息发送链接给用户。然后在小游戏初始化的时候在脚本中，通过Interface\GetopenidBycode.ashx服务内的内容返回出openid，然后在通过Interface\GetUserInfo.ashx内的服务获取用户信息，过程在例程里请自行查看。而相关脚本的调用则在Game\HZGift\inc\js\index.js中。

### 微网站侦听微信事件

在Game\HZGift\core\WeixinJSBridge.js中有例子如何侦听转发事件并转发指定的url

在Game\HZGift\inc\js\index.js中的HZGift.listenSendAppMessage()方法中则有调用的具体例子。我们可见在利用这个方法后就可以在用户点击右上角转发功能时转发出带有自己身份参数的url，别人点击的时候就能够实现记录推荐人的目地。

WeixinJSBridge中很多方法已经被微信封闭，但是侦听发送朋友，发送朋友圈，分享到微博等功能还能侦听。

## 小游戏开发

### 部署应用

在\Game\HZGift\下有一个已经开发好的小游戏，我们先将小游戏部署到外网上，例程里是跟着WXHandler.ashx一同部署在一个域下。

### 引导用户进入游戏

程序入口是Game\HZGift\index.html，我们只需要在公众号内给出一个入口，无论是用菜单的方式还是用消息的方式。只需要把用户引导到我们的网站下就行。在例程中我们提供了2中方式。

#### 通过消息

无论是转发还是群发还是客服消息，我们只需要发出一个URL就可以达到我们的目地。例程中GameHelper.GetLotteryReply方法中，返回了一个带游戏链接的图文消息。点击菜单的抽奖按钮也会返回相同的消息，用户点击我们的应用后便会进入我们的游戏。

#### 通过菜单

在菜单中直接加上链接URL用户点击直接进入。

### 获取用户信息

只需要通过网页授权获取的方式便可获取，我们发送链接的时候都不是直接发送网站，而是把网站通过参数的形式，发送请求到微信提供的认证网站，另外我们需要配置我们的认证域，[OAuth2.0网页授权](http://mp.weixin.qq.com/wiki/index.php?title=%E7%BD%91%E9%A1%B5%E6%8E%88%E6%9D%83%E8%8E%B7%E5%8F%96%E7%94%A8%E6%88%B7%E5%9F%BA%E6%9C%AC%E4%BF%A1%E6%81%AF)功能中把域名配置成我们游戏所在的域名，注意不能添加HTTP字眼。只需要域名本身，如：[www.baidu.com，112.223.44.223](http://www.baidu.com，112.223.44.223)

然后我们打开的index.html便会带上code=XXXXX的参数

然后我们再通过这个code，调用GetopenidBycode.ashx服务，从而返回openid，再根据openid获取用户信息。游戏中我们只是获取了用户是否关注这一信息，从而决定用户是否有参加抽奖的权利。

### 记录抽奖内容

我们通过openid在系统的数据库中记录用户的抽奖次数和中奖纪录。

### 实现推广

由于我们侦听了发送到朋友的动作，如果我们的用户在游戏页面转发了该游戏，便会自动带上自己的系统ID，然后在其他用户点击URL进入游戏时，便会被记录下推荐人。

### 反馈用户中奖信息

程序中GameHelper.GetGiftListReply显示了如何反馈给用户数据库内的第三方信息。

## 引用文档

http://mp.weixin.qq.com/wiki/18/28fc21e7ed87bec960651f0ce873ef8a.html