

# 微信公众平台开发者文档

tacyeh 收集整理

文档内容截至日期：2013-12-25

# 目 录

零、首页 .....	- 1 -
一、公众平台开发接口介绍 .....	- 1 -
二、典型案例介绍 .....	- 1 -
壹、新手接入 .....	- 4 -
一、接入指南 .....	- 4 -
二、典型案例介绍 .....	- 6 -
三、开发者规范 .....	- 9 -
贰、基础支持 .....	- 10 -
一、获取 access_token .....	- 10 -
二、全局返回码说明 .....	- 11 -
三、接口频率限制说明 .....	- 14 -
四、上传下载多媒体文件 .....	- 15 -
叁、接收消息 .....	- 18 -
一、验证消息真实性 .....	- 18 -
二、接收普通消息 .....	- 19 -
三、接收事件推送 .....	- 23 -
四、接收语音识别结果 .....	- 27 -
肆、发送消息 .....	- 28 -
一、发送被动响应消息 .....	- 28 -
二、发送客服消息 .....	- 31 -
伍、用户管理 .....	- 35 -

一、分组管理接口 .....	- 35 -
二、获取用户基本信息 .....	- 39 -
三、获取关注者列表 .....	- 40 -
四、获取用户地理位置 .....	- 43 -
五、网页授权获取用户基本信息 .....	- 43 -
六、网页获取用户网络状态 (JS 接口) .....	- 50 -
<b>陆、自定义菜单 .....</b>	<b>- 50 -</b>
一、自定义菜单创建接口 .....	- 50 -
二、自定义菜单查询接口 .....	- 53 -
三、自定义菜单删除接口 .....	- 54 -
四、自定义菜单事件推送 .....	- 54 -
<b>柒、推广支持 .....</b>	<b>- 55 -</b>
一、生成带参数的二维码 .....	- 55 -
二、创建二维码 ticket.....	- 55 -
三、通过 ticket 换取二维码.....	- 56 -
<b>捌、Winxin JS 接口 .....</b>	<b>- 57 -</b>
一、隐藏微信中网页右上角按钮 .....	- 57 -
二、隐藏微信中网页底部导航栏 .....	- 58 -
三、网页获取用户网络状态 .....	- 59 -

## 零、首页

### 一、公众平台开发接口介绍

公众平台是为微信用户提供服务的平台，而公众平台开发接口则是提供服务的基础，开发者在公众平台网站中创建公众号、获取接口权限后，可以通过阅读本接口文档来帮助开发。

公众平台开发接口提供与用户进行消息交互、自定义菜单交互的能力。对于成功接入公众平台开发接口的公众账号，当用户发消息给公众号，微信公众平台服务器会使用 `http` 请求对接入的网址进行消息推送，第三方服务器可通过响应包回复特定结构，从而达到回复消息的目的。

### 二、典型案例介绍

值得借鉴的公众帐号主要是服务号，试列举并介绍如下：

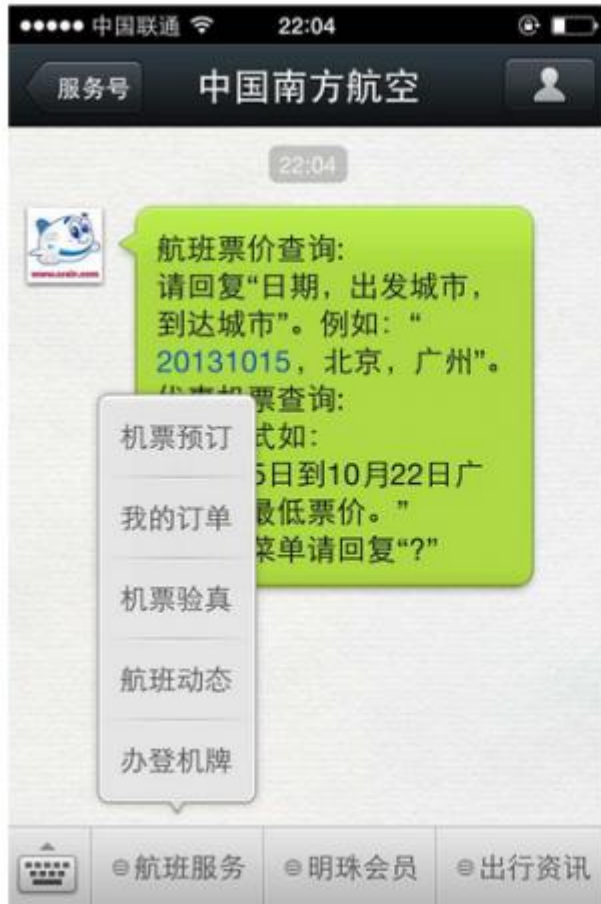
招商银行信用卡中心



如果你是持卡人，可快捷查询信用卡账单、额度及积分；快速还款、申请账单分期；微信转接人工服务；信用卡消费，微信免费笔笔提醒。如果不是持卡人，可以微信办卡！

招商银行公众号通过提示消息引导用户将自己的微信号和信用卡号安全绑定。用户可以通过该公众号查询账单、收取刷卡通知等功能，这是由招行开发人员通过公众号接口实现的功能。

中国南方航空



你可以办理值机手续，挑选座位，查询航班信息，查询目的地城市天气，并为明珠会员提供专业的服务。

南方航空公众号可以让用户将明珠会员服务和微信号绑定起来。用户可以通过该公众号预订机票、查询订单，甚至办理登机牌。

广东联通



你可以在微信里绑定手机号、积分流量，套餐余量、手机上网流量，微信专属流量查询，客服咨询。

广东联通公众号可以绑定手机号，来查询流量、套餐等等功能。  
广东联通更与微信深度合作，购买微信沃卡可以获得微信五大特权。

## 壹、新手接入

### 一、接入指南

#### 第一步：申请消息接口

**接口配置信息**

请填写接口配置信息，此信息需要你拥有自己的服务器资源。填写的URL需要正确响应微信发送的Token验证，请阅读[消息接口使用指南](#)

URL

Token

[什么是Token ?](#)

在公众平台网站的高级功能 - 开发模式页，点击“成为开发者”按钮，填写 URL 和 Token，其中 URL 是开发者用来接收微信服务器数据的接口 URL。Token 可由开发者可以任意填写，用作生成签名（该 Token 会和接口 URL 中包含的 Token 进行比对，从而验证安全性）。

## 第二步：验证 URL 有效性

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

开发者通过检验 signature 对请求进行校验（下面有校验方式）。若确认此次 GET 请求来自微信服务器，请原样返回 echostr 参数内容，则接入生效，成为开发者成功，否则接入失败。

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。

检验 signature 的 PHP 示例代码：

```
1. private function checkSignature()  
2. {  
3.     $signature = $_GET["signature"];  
4.     $timestamp = $_GET["timestamp"];  
5.     $nonce = $_GET["nonce"];  
6.     $token = TOKEN;
```



```
7.  $tmpArr = array($token, $timestamp, $nonce);
8.  sort($tmpArr);
9.  $tmpStr = implode( $tmpArr );
10. $tmpStr = sha1( $tmpStr );
11. if( $tmpStr == $signature ){
12.     return true;
13. }else{
14.     return false;
15. }
16. }
```

PHP 示例代码下载：[http://mp.weixin.qq.com/mpres/htmledition/res/wx\\_sample.zip](http://mp.weixin.qq.com/mpres/htmledition/res/wx_sample.zip)

### 第三步：成为开发者

验证 URL 有效性成功后即接入生效，成为开发者。如果公众号类型为服务号（订阅号只能使用普通消息接口），可以在公众平台网站中申请认证，认证成功的服务号将获得众多接口权限，以满足开发者需求。

此后用户每次向公众号发送消息、或者产生自定义菜单点击事件时，响应 URL 将得到推送。

公众号调用各接口时，一般会获得正确的结果，具体结果可见对应接口的说明。返回错误时，可根据返回码来查询错误原因。全局返回码说明

用户向公众号发送消息时，公众号方收到的消息发送者是一个 OpenID，是使用用户微信号加密后的结果，每个用户对每个公众号有一个唯一的 OpenID。

此外请注意，**微信公众号接口只支持 80 接口。**

## 二、典型案例介绍

值得借鉴的公众帐号主要是服务号，试列举并介绍如下：

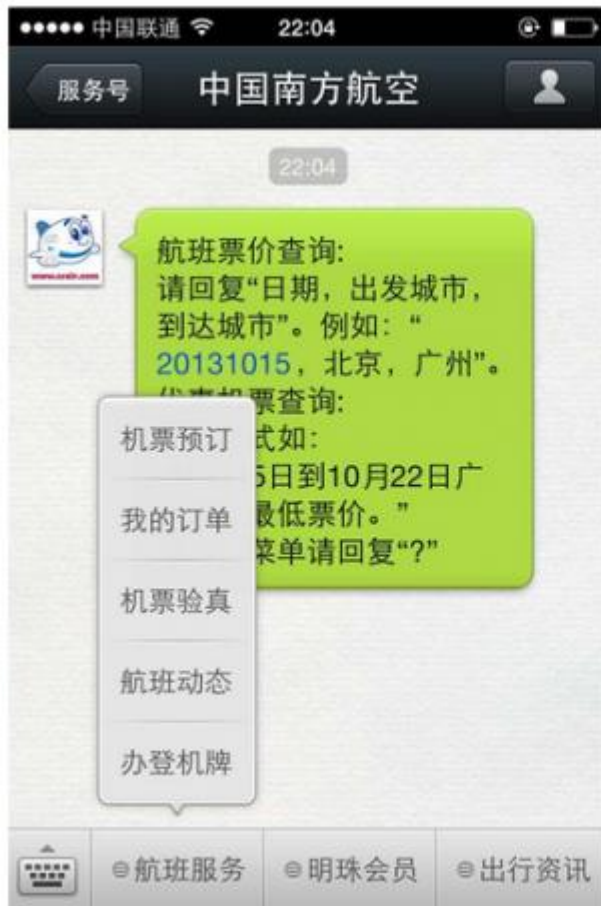
招商银行信用卡中心



如果你是持卡人，可快捷查询信用卡账单、额度及积分；快速还款、申请账单分期；微信转接人工服务；信用卡消费，微信免费笔笔提醒。如果不是持卡人，可以微信办卡！

招商银行公众号通过提示消息引导用户将自己的微信号和信用卡号安全绑定。用户可以通过该公众号查询账单、收取刷卡通知等功能，这是由招行开发人员通过公众号接口实现的功能。

中国南方航空



你可以办理值机手续，挑选座位，查询航班信息，查询目的地城市天气，并为明珠会员提供专业的服务。

南方航空公众号可以让用户将明珠会员服务和微信号绑定起来。用户可以通过该公众号预订机票、查询订单，甚至办理登机牌。

广东联通



你可以在微信里绑定手机号、积分流量，套餐余量、手机上网流量，微信专属流量查询，客服咨询。

广东联通公众号可以绑定手机号，来查询流量、套餐等等功能。  
广东联通更与微信深度合作，购买微信沃卡可以获得微信五大特权。

### 三、开发者规范

开发者进行公众号开发时，除了需要满足每个接口的规范限制、调用频率限制外，还需特别注意模版消息、用户数据等敏感信息的使用规范。

#### 涉及用户数据时：

- 您的服务需要收集用户任何数据的，必须事先获得用户的明确同意，且仅应当收集为运营及功能实现目的而必要的用户数据，同时应当告知用户相关数据收集的目的、范围及使用方式等，保障用户知情权。
- 您收集用户的数据后，必须采取必要的保护措施，防止用户数据被盗、泄漏等。

- 您在特定微信公众号中收集的用户数据仅可以在该特定微信公众号中使用，不得将其使用在该特定微信公众号之外或为其他任何目的进行使用，也不得以任何方式将其提供给他人。
- 如果腾讯认为您收集、使用用户数据的方式，可能损害用户体验，腾讯有权要求您删除相关数据并不得再以该方式收集、使用用户数据。
- 一旦您停止使用本服务，或腾讯基于任何原因终止您使用本服务，您必须立即删除全部因使用本服务而获得的数据（包括各种备份），且不得再以任何方式进行使用。

#### 其他规范：

- 请勿为任何用户自动登录到微信公众平台提供代理身份验证凭据。
- 请勿提供跟踪功能，包括但不限于识别其他用户在个人主页上查看、点击等操作行为。
- 请勿自动将浏览器窗口定向到其他网页。
- 请勿设置或发布任何违反相关法规、公序良俗、社会公德等的玩法、内容等。
- 请勿公开表达或暗示，您与腾讯之间存在合作关系，包括但不限于相互持股、商业往来或合作关系等，或声称腾讯对您的认可。

完整的开发者规范和接口限制，请详见开发者接口文档，以及腾讯微信公众平台开发者协议。

## 贰、基础支持

### 一、获取 `access_token`

`access_token` 是公众号的全局唯一票据，公众号调用各接口时都需使用 `access_token`。正常情况下 `access_token` 有效期为 7200 秒，重复获取将导致上次获取的 `access_token` 失效。

公众号可以使用 AppID 和 AppSecret 调用本接口来获取 access\_token。AppID 和 AppSecret 可在开发模式中获取（需要已经成为开发者，且帐号没有异常状态）。注意调用所有微信接口时均需使用 **https** 协议。

#### 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
```

#### 参数说明

参数	是否必须
grant_type	是
appid	是
secret	是

#### 返回说明

正常情况下，微信会返回下述 JSON 数据包给公众号：

```
{"access_token":"ACCESS_TOKEN","expires_in":7200}
```

参数	说明
access_token	获取到的凭证
expires_in	凭证有效时间，单位：秒

错误时微信会返回错误码等信息，JSON 数据包示例如下（该示例为 AppID 无效错误）：

```
{"errcode":40013,"errmsg":"invalid appid"}
```

## 二、全局返回码说明

公众号每次调用接口时，可能获得正确或错误的返回码，开发者可以根据返

回码信息调试接口，排查错误。

全局返回码说明如下：

返回码	说明
-1	系统繁忙
0	请求成功
40001	获取 access_token 时 AppSecret 错误，或者 access_token 无效
40002	不合法的凭证类型
40003	不合法的 OpenID
40004	不合法的媒体文件类型
40005	不合法的文件类型
40006	不合法的文件大小
40007	不合法的媒体文件 id
40008	不合法的消息类型
40009	不合法的图片文件大小
40010	不合法的语音文件大小
40011	不合法的视频文件大小
40012	不合法的缩略图文件大小
40013	不合法的 APPID
40014	不合法的 access_token
40015	不合法的菜单类型
40016	不合法的按钮个数
40017	不合法的按钮个数
40018	不合法的按钮名字长度
40019	不合法的按钮 KEY 长度
40020	不合法的按钮 URL 长度
40021	不合法的菜单版本号
40022	不合法的子菜单级数
40023	不合法的子菜单按钮个数
40024	不合法的子菜单按钮类型
40025	不合法的子菜单按钮名字长度
40026	不合法的子菜单按钮 KEY 长度
40027	不合法的子菜单按钮 URL 长度
40028	不合法的自定义菜单使用用户
40029	不合法的 oauth_code
40030	不合法的 refresh_token
40031	不合法的 openid 列表
40032	不合法的 openid 列表长度
40033	不合法的请求字符，不能包含 \uxxxx 格式的字符
40035	不合法的参数

40038	不合法的请求格式
40039	不合法的 URL 长度
40050	不合法的分组 id
40051	分组名字不合法
41001	缺少 access_token 参数
41002	缺少 appid 参数
41003	缺少 refresh_token 参数
41004	缺少 secret 参数
41005	缺少多媒体文件数据
41006	缺少 media_id 参数
41007	缺少子菜单数据
41008	缺少 oauth code
41009	缺少 openid
42001	access_token 超时
42002	refresh_token 超时
42003	oauth_code 超时
43001	需要 GET 请求
43002	需要 POST 请求
43003	需要 HTTPS 请求
43004	需要接收者关注
43005	需要好友关系
44001	多媒体文件为空
44002	POST 的数据包为空
44003	图文消息内容为空
44004	文本消息内容为空
45001	多媒体文件大小超过限制
45002	消息内容超过限制
45003	标题字段超过限制
45004	描述字段超过限制
45005	链接字段超过限制
45006	图片链接字段超过限制
45007	语音播放时间超过限制
45008	图文消息超过限制
45009	接口调用超过限制
45010	创建菜单个数超过限制
45015	回复时间超过限制
45016	系统分组，不允许修改
45017	分组名字过长
45018	分组数量超过上限



46001	不存在媒体数据
46002	不存在的菜单版本
46003	不存在的菜单数据
46004	不存在的用户
47001	解析 JSON/XML 内容错误
48001	api 功能未授权
50001	用户未授权该 api

### 三、接口频率限制说明

公众号调用接口并不是无限制的。为了防止公众号的程序错误而引发微信服务器负载异常，默认情况下，每个公众号调用接口都不能超过一定限制，当超过一定限制时，调用对应接口会收到如下错误返回码：

```
{"errcode":45009,"errmsg":"api freq out of limit"}
```

各接口调用频率限制如下：

接口	每日限额
获取 access_token	2000
自定义菜单创建	1000
自定义菜单查询	10000
自定义菜单删除	1000
创建分组	1000
获取分组	1000
修改分组名	1000
移动用户分组	100000
上传多媒体文件	5000
下载多媒体文件	10000
发送客服消息	500000
获取带参数的二维码	100000
获取关注者列表	500
获取用户基本信息	5000000
获取网页授权 access_token	2000000
刷新网页授权 access_token	2000000
网页授权获取用户信息	2000000

请注意，在测试号申请页中申请的测试号，接口调用频率限制如下：

接口	每日限额
获取 access_token	200
自定义菜单创建	100

自定义菜单查询	1000
自定义菜单删除	100
创建分组	100
获取分组	100
修改分组名	100
移动用户分组	1000
上传多媒体文件	500
下载多媒体文件	1000
发送客服消息	50000
获取带参数的二维码	10000
获取关注者列表	100
获取用户基本信息	500000
获取网页授权 access_token	200000
刷新网页授权 access_token	200000
网页授权获取用户信息	200000

#### 四、上传下载多媒体文件

公众号在使用接口时，对多媒体文件、多媒体消息的获取和调用等操作，是通过 `media_id` 来进行的。通过本接口，公众号可以上传或下载多媒体文件。但请注意，每个多媒体文件（`media_id`）会在上传、用户发送到微信服务器 3 天后自动删除，以节省服务器资源。

##### 上传多媒体文件

公众号可调用本接口来上传图片、语音、视频等文件到微信服务器，上传后服务器会返回对应的 `media_id`，公众号此后可根据该 `media_id` 来获取多媒体。

请注意，`media_id` 是可复用的，调用该接口需 **http** 协议。

##### 接口调用请求说明

http 请求方式: POST/FORM

http://file.api.weixin.qq.com/cgi-bin/media/upload?access\_token=ACCESS\_T  
OKEN&type=TYPE

调用示例（使用 curl 命令，用 FORM 表单方式上传一个多媒体文件）：

```
curl -F media=@test.jpg
```

```
"http://file.api.weixin.qq.com/cgi-bin/media/upload?access_token=ACCESS_T  
OKEN&type=TYPE"
```

### 参数说明

参数	是否必须	说明
access_token	是	调用接口凭证
type	是	媒体文件类型，分别有图片（image）、语音（voice）、视频（video）和缩略图（thumb）
media	是	form-data 中媒体文件标识，有 filename、filelength、content-type 等信息

### 返回说明

正确情况下的返回 JSON 数据包结果如下：

```
{"type":"TYPE","media_id":"MEDIA_ID","created_at":123456789}
```

参数	描述
type	媒体文件类型，分别有图片（image）、语音（voice）、视频（video）和缩略图（thumb，主要用于视频与音乐格式的缩略图）
media_id	媒体文件上传后，获取时的唯一标识
created_at	媒体文件上传时间戳

错误情况下的返回 JSON 数据包示例如下（示例为无效媒体类型错误）：

```
{"errcode":40004,"errmsg":"invalid media type"}
```

### 注意事项

上传的多媒体文件有格式和大小限制，如下：

- 图片（image）：128K，支持 JPG 格式
- 语音（voice）：256K，播放长度不超过 60s，支持 AMR\MP3 格式
- 视频（video）：1MB，支持 MP4 格式
- 缩略图（thumb）：64KB，支持 JPG 格式

媒体文件在后台保存时间为 **3** 天，即 **3** 天后 **media\_id** 失效。

## 下载多媒体文件

公众号可调用本接口来获取多媒体文件。请注意，视频文件不支持下载，调用该接口需 **http** 协议。

### 接口调用请求说明

http 请求方式: GET

```
http://file.api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOKE  
N&media_id=MEDIA_ID
```

请求示例（示例为通过 curl 命令获取多媒体文件）

```
curl -I -G
```

```
"http://file.api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOK  
EN&media_id=MEDIA_ID"
```

### 参数说明

参数	是否必须	说明
access_token	是	调用接口凭证
media_id	是	媒体文件 ID

### 返回说明

正确情况下的返回 HTTP 头如下：

```
HTTP/1.1 200 OK  
Connection: close  
Content-Type: image/jpeg  
Content-disposition: attachment; filename="MEDIA_ID.jpg"  
Date: Sun, 06 Jan 2013 10:20:18 GMT  
Cache-Control: no-cache, must-revalidate  
Content-Length: 339721
```

```
curl -G

"http://file.api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOK
EN&media_id=MEDIA_ID"
```

错误情况下的返回 JSON 数据包示例如下（示例为无效媒体 ID 错误）：：

```
{"errcode":40007,"errmsg":"invalid media_id"}
```

## 叁、接收消息

### 一、验证消息真实性

在开发者首次提交验证申请时，微信服务器将发送 GET 请求到填写的 URL 上，并且带上四个参数（signature、timestamp、nonce、echostr），开发者通过对签名（即 signature）的效验，来判断此条消息的真实性。

此后，每次开发者接收用户消息的时候，微信也都会带上前面三个参数（signature、timestamp、nonce）访问开发者设置的 URL，开发者依然通过对签名的效验判断此条消息的真实性。效验方式与首次提交验证申请一致。

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

开发者通过检验 signature 对请求进行校验（下面有校验方式）。若确认此次 GET 请求来自微信服务器，请原样返回 echostr 参数内容，则接入生效，成为开发者成功，否则接入失败。

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

检验 signature 的 PHP 示例代码：

```
17. private function checkSignature()
18. {
19.     $signature = $_GET["signature"];
20.     $timestamp = $_GET["timestamp"];
21.     $nonce = $_GET["nonce"];
22.     $token = TOKEN;
23.     $tmpArr = array($token, $timestamp, $nonce);
24.     sort($tmpArr);
25.     $tmpStr = implode( $tmpArr );
26.     $tmpStr = sha1( $tmpStr );
27.     if( $tmpStr == $signature ){
28.         return true;
29.     }else{
30.         return false;
31.     }
32. }
```

## 二、接收普通消息

当普通微信用户向公众账号发消息时，微信服务器将 POST 消息的 XML 数据包到开发者填写的 URL 上。各消息类型的推送 XML 数据包结构如下。

### 1、文本消息

XML 数据包示例：

```
33. <xml>
34. <ToUserName><![CDATA[toUser]]></ToUserName>
35. <FromUserName><![CDATA[fromUser]]></FromUserName>
36. <CreateTime>1348831860</CreateTime>
37. <MsgType><![CDATA[text]]></MsgType>
38. <Content><![CDATA[this is a test]]></Content>
39. <MsgId>1234567890123456</MsgId>
40. </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	text
Content	文本消息内容
MsgId	消息 id，64 位整型

## 2、图片消息

**XML 数据包示例：**

```

41. <xml>
42. <ToUserName><![CDATA[toUser]]></ToUserName>
43. <FromUserName><![CDATA[fromUser]]></FromUserName>
44. <CreateTime>1348831860</CreateTime>
45. <MsgType><![CDATA[image]]></MsgType>
46. <PicUrl><![CDATA[this is a url]]></PicUrl>
47. <MediaId><![CDATA[media_id]]></MediaId>
48. <MsgId>1234567890123456</MsgId>
49. </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	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

## 3、语音消息

**XML 数据包示例：**

```

50. <xml>
51. <ToUserName><![CDATA[toUser]]></ToUserName>

```

```

52. <FromUserName><![CDATA[fromUser]]></FromUserName>
53. <CreateTime>1357290913</CreateTime>
54. <MsgType><![CDATA[voice]]></MsgType>
55. <MediaId><![CDATA[media_id]]></MediaId>
56. <Format><![CDATA[Format]]></Format>
57. <MsgId>1234567890123456</MsgId>
58. </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	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

## 4、视频消息

XML 数据包示例:

```

59. <xml>
60. <ToUserName><![CDATA[toUser]]></ToUserName>
61. <FromUserName><![CDATA[fromUser]]></FromUserName>
62. <CreateTime>1357290913</CreateTime>
63. <MsgType><![CDATA[video]]></MsgType>
64. <MediaId><![CDATA[media_id]]></MediaId>
65. <ThumbMediaId><![CDATA[thumb_media_id]]></ThumbMediaId>
66. <MsgId>1234567890123456</MsgId>
67. </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	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

## 5、地理位置消息

**XML 数据包示例:**

```

68. <xml>
69. <ToUserName><![CDATA[toUser]]></ToUserName>
70. <FromUserName><![CDATA[fromUser]]></FromUserName>
71. <CreateTime>1351776360</CreateTime>
72. <MsgType><![CDATA[location]]></MsgType>
73. <Location_X>23.134521</Location_X>
74. <Location_Y>113.358803</Location_Y>
75. <Scale>20</Scale>
76. <Label><![CDATA[位置信息]]></Label>
77. <MsgId>1234567890123456</MsgId>
78. </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	location
Location_X	地理位置维度
Location_Y	地理位置精度
Scale	地图缩放大小
Label	地理位置信息
MsgId	消息 id, 64 位整型

## 6、链接消息

**XML 数据包示例:**

```

79. <xml>

```

```

80. <ToUserName><![CDATA[toUser]]></ToUserName>
81. <FromUserName><![CDATA[fromUser]]></FromUserName>
82. <CreateTime>1351776360</CreateTime>
83. <MsgType><![CDATA[link]]></MsgType>
84. <Title><![CDATA[公众平台官网链接]]></Title>
85. <Description><![CDATA[公众平台官网链接]]></Description>
86. <Url><![CDATA[url]]></Url>
87. <MsgId>1234567890123456</MsgId>
88. </xml>

```

参数说明：

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

### 三、接收事件推送

#### 1、关注/取消关注事件

用户在关注与取消关注公众号事，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。

推送 XML 数据包示例：

```

89. <xml>
90. <ToUserName><![CDATA[toUser]]></ToUserName>
91. <FromUserName><![CDATA[FromUser]]></FromUserName>
92. <CreateTime>123456789</CreateTime>
93. <MsgType><![CDATA[event]]></MsgType>
94. <Event><![CDATA[subscribe]]></Event>
95. </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)

## 2、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:

2.1 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。

2.2、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。

### 2.1、 用户未关注时, 进行关注后的事件推送

推送 XML 数据包示例:

```
96. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
97. <FromUserName><![CDATA[FromUser]]></FromUserName>
98. <CreateTime>123456789</CreateTime>
99. <MsgType><![CDATA[event]]></MsgType>
100. <Event><![CDATA[subscribe]]></Event>
101. <EventKey><![CDATA[qrcode_123123]]></EventKey>
102. <Ticket><![CDATA[TICKET]]></Ticket>
103. </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

## 2.2、用户已关注时的事件推送

推送 XML 数据包示例：

```

104. <xml>
105. <ToUserName><![CDATA[toUser]]></ToUserName>
106. <FromUserName><![CDATA[FromUser]]></FromUserName>
107. <CreateTime>123456789</CreateTime>
108. <MsgType><![CDATA[event]]></MsgType>
109. <Event><![CDATA[scan]]></Event>
110. <EventKey><![CDATA[SCENE_VALUE]]></EventKey>
111. <Ticket><![CDATA[TICKET]]></Ticket>
112. </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（整形）
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
Ticket	二维码的 ticket，可用来换取二维码图片

## 3、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：

```

113. <xml>
114. <ToUserName><![CDATA[toUser]]></ToUserName>

```

```

115. <FromUserName><![CDATA[fromUser]]></FromUserName>
116. <CreateTime>123456789</CreateTime>
117. <MsgType><![CDATA[event]]></MsgType>
118. <Event><![CDATA[LOCATION]]></Event>
119. <Latitude>23.137466</Latitude>
120. <Longitude>113.352425</Longitude>
121. <Precision>119.385040</Precision>
122. </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

#### 4、自定义菜单事件

用户点击自定义菜单后，如果菜单按钮设置为 click 类型，则微信会把此次点击事件推送给开发者，注意 view 类型（跳转到 URL）的菜单点击不会上报。

推送 XML 数据包示例：

```

123. <xml>
124. <ToUserName><![CDATA[toUser]]></ToUserName>
125. <FromUserName><![CDATA[FromUser]]></FromUserName>
126. <CreateTime>123456789</CreateTime>
127. <MsgType><![CDATA[event]]></MsgType>
128. <Event><![CDATA[CLICK]]></Event>
129. <EventKey><![CDATA[EVENTKEY]]></EventKey>
130. </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

#### 四、接收语音识别结果

开通语音识别功能，用户每次发送语音给公众号时，微信会在推送的语音消息 XML 数据包中，增加一个 Recognition 字段。

注：由于客户端缓存，开发者开启或者关闭语音识别功能，对新关注者立刻生效，对已关注用户需要 24 小时生效。开发者可以重新关注此帐号进行测试。

开启语音识别后的语音 XML 数据包如下：

```

131. <xml>
132. <ToUserName><![CDATA[toUser]]></ToUserName>
133. <FromUserName><![CDATA[fromUser]]></FromUserName>
134. <CreateTime>1357290913</CreateTime>
135. <MsgType><![CDATA[voice]]></MsgType>
136. <MediaId><![CDATA[media_id]]></MediaId>
137. <Format><![CDATA[Format]]></Format>
138. <Recognition><![CDATA[腾讯微信团队]]></Recognition>
139. <MsgId>1234567890123456</MsgId>
140. </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	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该媒体
Format	语音格式：amr

Recognition	语音识别结果，UTF8 编码
MsgID	消息 id，64 位整型

## 肆、发送消息

### 一、发送被动响应消息

对于每一个 POST 请求，开发者在响应包（Get）中返回特定 XML 结构，对该消息进行响应（现支持回复文本、图片、图文、语音、视频、音乐）。请注意，回复图片等多媒体消息时需要预先上传多媒体文件到微信服务器，只支持认证服务号。

微信服务器在五秒内收不到响应会断掉连接，如果在调试中，发现用户无法收到响应的消息，可以检查是否消息处理超时。各消息类型需要的 XML 数据包结构如下。

#### 1、回复文本消息

```

141. <xml>
142. <ToUserName><![CDATA[toUser]]></ToUserName>
143. <FromUserName><![CDATA[fromUser]]></FromUserName>
144. <CreateTime>12345678</CreateTime>
145. <MsgType><![CDATA[text]]></MsgType>
146. <Content><![CDATA[你好]]></Content>
147. </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）

#### 2、回复图片消息

```

148. <xml>
149. <ToUserName><![CDATA[toUser]]></ToUserName>
150. <FromUserName><![CDATA[fromUser]]></FromUserName>
151. <CreateTime>12345678</CreateTime>

```

```

152. <MsgType><![CDATA[image]]></MsgType>
153. <Image>
154. <MediaId><![CDATA[media_id]]></MediaId>
155. </Image>
156. </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。

### 3、回复语音消息

```

157. <xml>
158. <ToUserName><![CDATA[toUser]]></ToUserName>
159. <FromUserName><![CDATA[fromUser]]></FromUserName>
160. <CreateTime>12345678</CreateTime>
161. <MsgType><![CDATA[voice]]></MsgType>
162. <Voice>
163. <MediaId><![CDATA[media_id]]></MediaId>
164. </Voice>
165. </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

### 4、回复视频消息

```

166. <xml>
167. <ToUserName><![CDATA[toUser]]></ToUserName>
168. <FromUserName><![CDATA[fromUser]]></FromUserName>
169. <CreateTime>12345678</CreateTime>
170. <MsgType><![CDATA[video]]></MsgType>
171. <Video>
172. <MediaId><![CDATA[media_id]]></MediaId>
173. <Title><![CDATA[title]]></Title>
174. <Description><![CDATA[description]]></Description>
175. </Video>
176. </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

## 5、回复音乐消息

```

177. <xml>
178. <ToUserName><![CDATA[toUser]]></ToUserName>
179. <FromUserName><![CDATA[fromUser]]></FromUserName>
180. <CreateTime>12345678</CreateTime>
181. <MsgType><![CDATA[music]]></MsgType>
182. <Music>
183. <Title><![CDATA[TITLE]]></Title>
184. <Description><![CDATA[DESCRIPTION]]></Description>
185. <MusicUrl><![CDATA[MUSIC_Url]]></MusicUrl>
186. <HQMusicUrl><![CDATA[HQ_MUSIC_Url]]></HQMusicUrl>
187. <ThumbMediaId><![CDATA[media_id]]></ThumbMediaId>
188. </Music>
189. </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

## 6、回复图文消息

```

190. <xml>
191. <ToUserName><![CDATA[toUser]]></ToUserName>
192. <FromUserName><![CDATA[fromUser]]></FromUserName>
193. <CreateTime>12345678</CreateTime>
194. <MsgType><![CDATA[news]]></MsgType>
195. <ArticleCount>2</ArticleCount>
196. <Articles>

```

```

197. <item>
198. <Title><![CDATA[title1]]></Title>
199. <Description><![CDATA[description1]]></Description>
200. <PicUrl><![CDATA[picurl]]></PicUrl>
201. <Url><![CDATA[url]]></Url>
202. </item>
203. <item>
204. <Title><![CDATA[title]]></Title>
205. <Description><![CDATA[description]]></Description>
206. <PicUrl><![CDATA[picurl]]></PicUrl>
207. <Url><![CDATA[url]]></Url>
208. </item>
209. </Articles>
210. </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

## 二、发送客服消息

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 `click` 事件、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（目前为 24 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 24 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。

### 接口调用请求说明

http 请求方式: POST

https://api.weixin.qq.com/cgi-bin/message/custom/send?access\_token=ACC  
ESS\_TOKEN

各消息类型所需的 JSON 数据包如下:

## 1、发送文本消息

```
211. {  
212.     "touser":"OPENID",  
213.     "msgtype":"text",  
214.     "text":  
215.     {  
216.         "content":"Hello World"  
217.     }  
218. }
```

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

## 2、发送图片消息

```
219. {  
220.     "touser":"OPENID",  
221.     "msgtype":"image",  
222.     "image":  
223.     {  
224.         "media_id":"MEDIA_ID"  
225.     }  
226. }
```

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	发送的图像的媒体 ID

## 3、发送语音消息

```

227. {
228.     "touser":"OPENID",
229.     "msgtype":"voice",
230.     "voice":
231.     {
232.         "media_id":"MEDIA_ID"
233.     }
234. }

```

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

#### 4、发送视频消息

```

235. {
236.     "touser":"OPENID",
237.     "msgtype":"video",
238.     "video":
239.     {
240.         "media_id":"MEDIA_ID",
241.         "title":"TITLE",
242.         "description":"DESCRIPTION"
243.     }
244. }

```

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
title	否	视频消息的标题
description	否	视频消息的描述

#### 5、发送音乐消息

```

245. {
246.     "touser":"OPENID",
247.     "msgtype":"music",
248.     "music":
249.     {
250.         "title":"MUSIC_TITLE",
251.         "description":"MUSIC_DESCRIPTION",
252.         "musicurl":"MUSIC_URL",

```

```

253.     "hqmusicurl":"HQ_MUSIC_URL",
254.     "thumb_media_id":"THUMB_MEDIA_ID"
255.   }
256. }

```

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

## 6、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。

```

257. {
258.   "touser":"OPENID",
259.   "msgtype":"news",
260.   "news":{
261.     "articles": [
262.       {
263.         "title":"Happy Day",
264.         "description":"Is Really A Happy Day",
265.         "url":"URL",
266.         "picurl":"PIC_URL"
267.       },
268.       {
269.         "title":"Happy Day",
270.         "description":"Is Really A Happy Day",
271.         "url":"URL",
272.         "picurl":"PIC_URL"
273.       }
274.     ]
275.   }
276. }

```

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

## 伍、用户管理

### 一、分组管理接口

开发者可以使用接口, 对公众平台的分组进行查询、创建、修改操作, 也可以使用接口在需要时移动用户到某个分组。

#### 1、创建分组

一个公众账号, 最多支持创建 500 个分组。

#### 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/groups/create?access\_token=ACCESS\_TOKEN

POST 数据格式: json

POST 数据例子: {"group":{"name":"test"}}

#### 参数说明

参数	说明
access_token	调用接口凭证
name	分组名字 (30 个字符以内)

#### 返回说明

正常时的返回 JSON 数据包示例:

```

277. {
278.   "group": {
279.     "id": 107,
280.     "name": "test"
281.   }
282. }
```

#### 参数说明

参数	说明
id	分组 id, 由微信分配
name	分组名字, UTF8 编码

错误时的 JSON 数据包示例 (该示例为 AppID 无效错误) :

```
{"errcode":40013,"errmsg":"invalid appid"}
```

## 2、查询所有分组

### 接口调用请求说明

http 请求方式: GET (请使用 https 协议)

[https://api.weixin.qq.com/cgi-bin/groups/get?access\\_token=ACCESS\\_TOKEN](https://api.weixin.qq.com/cgi-bin/groups/get?access_token=ACCESS_TOKEN)

### 参数说明

参数	说明
access_token	调用接口凭证

### 返回说明

正常时的返回 JSON 数据包示例:

```
283. {
284.   "groups": [
285.     {
286.       "id": 0,
287.       "name": "未分组",
288.       "count": 72596
289.     },
290.     {
291.       "id": 1,
292.       "name": "黑名单",
293.       "count": 36
294.     },
295.     {
296.       "id": 2,
297.       "name": "星标组",
298.       "count": 8
299.     },
300.     {
301.       "id": 104,
302.       "name": "华东媒",
303.       "count": 4
304.     },
```

```

305.     {
306.         "id": 106,
307.         "name": "★不测试组★",
308.         "count": 1
309.     }
310. ]
311. }

```

#### 参数说明

参数	说明
groups	公众平台分组信息列表
id	分组 id, 由微信分配
name	分组名字, UTF8 编码
count	分组内用户数量

错误时的 JSON 数据包示例 (该示例为 AppID 无效错误):

```
{"errcode":40013,"errmsg":"invalid appid"}
```

### 3、查询用户所在分组

通过用户的 OpenID 查询其所在的 GroupID。

#### 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/groups/getid?access\_token=ACCESS\_TOKEN

POST 数据格式: json

POST 数据例子: {"openid":"od8XIjsmk6QdVTETa9jLtGWA6KBc"}

#### 参数说明

参数	说明
access_token	调用接口凭证
openid	用户的 OpenID

#### 返回说明

正常时的返回 JSON 数据包示例:

```

312. {
313.     "groupid": 102
314. }

```

#### 参数说明



参数	说明
groupid	用户所属的 groupid

错误时的 JSON 数据包示例（该示例为 OpenID 无效错误）：

```
{"errcode":40003,"errmsg":"invalid openid"}
```

## 4、修改分组名

### 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/groups/update?access\_token=ACCESS\_TOKEN

POST 数据格式: json

POST 数据例子: {"group":{"id":108,"name":"test2\_modify2"}}

### 参数说明

参数	说明
access_token	调用接口凭证
id	分组 id, 由微信分配
name	分组名字（30 个字符以内）

### 返回说明

正常时的返回 JSON 数据包示例：

```
{"errcode": 0, "errmsg": "ok"}
```

错误时的 JSON 数据包示例（该示例为 AppID 无效错误）：

```
{"errcode":40013,"errmsg":"invalid appid"}
```

## 5、移动用户分组

### 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/groups/members/update?access\_token=ACCESS\_TOKEN

POST 数据格式: json

POST 数据例子: {"openid":"oDF3iYx0ro3\_7jD4HFRDfrjdCM58","to\_groupid":108}

### 参数说明

参数	说明
access_token	调用接口凭证
openid	用户唯一标识符
to_groupid	分组 id

### 返回说明

正常时的返回 JSON 数据包示例：

```
{"errcode": 0, "errmsg": "ok"}
```

错误时的 JSON 数据包示例（该示例为 AppID 无效错误）：

```
{"errcode":40013,"errmsg":"invalid appid"}
```

## 二、获取用户基本信息

在关注者与公众号产生消息交互后，公众号可获得关注者的 OpenID（加密后的微信号，每个用户对每个公众号的 OpenID 是唯一的。对于不同公众号，同一用户的 openid 不同）。公众号可通过本接口来根据 OpenID 获取用户基本信息，包括昵称、头像、性别、所在城市、语言和关注时间。

### 获取用户基本信息

开发者可通过 OpenID 来获取用户基本信息。请使用 https 协议。

### 接口调用请求说明

http 请求方式: GET

```
https://api.weixin.qq.com/cgi-bin/user/info?access_token=ACCESS_TOKEN&openid=OPENID
```

### 参数说明

参数	是否必须	说明
access_token	是	调用接口凭证
openid	是	普通用户的标识，对当前公众号唯一

### 返回说明

正常情况下，微信会返回下述 JSON 数据包给公众号：

```

315. {
316.     "subscribe": 1,
317.     "openid": "o6_bmjrrPTIm6_2sgVt7hMZOPfL2M",
318.     "nickname": "Band",
319.     "sex": 1,
320.     "language": "zh_CN",
321.     "city": "广州",
322.     "province": "广东",
323.     "country": "中国",
324.     "headimgurl":
        "http://wx.qlogo.cn/mmopen/g3MonUZtNHkdmzicIlibx6iaFqAc56vxLSUfpb6n5WKSYVY0ChQKkia
        JSgQ1dZuTOgvLLrhJbERQQ4eMsv84eavHiaiceqxibJxCfHe/0",
325.     "subscribe_time": 1382694957
326. }

```

### 参数说明

参数	说明
subscribe	用户是否订阅该公众号标识，值为 0 时，代表此用户没有关注该公众号，拉取不到其余信息。
openid	用户的标识，对当前公众号唯一
nickname	用户的昵称
sex	用户的性别，值为 1 时是男性，值为 2 时是女性，值为 0 时是未知
city	用户所在城市
country	用户所在国家
province	用户所在省份
language	用户的语言，简体中文为 zh_CN
headimgurl	用户头像，最后一个数值代表正方形头像大小（有 0、46、64、96、132 数值可选，0 代表 640*640 正方形头像），用户没有头像时该项为空
subscribe_time	用户关注时间，为时间戳。如果用户曾多次关注，则取最后关注时间

错误时微信会返回错误码等信息，JSON 数据包示例如下（该示例为 AppID

无效错误）：

```
{"errcode":40013,"errmsg":"invalid appid"}
```

### 三、获取关注者列表

公众号可通过本接口来获取帐号的关注者列表，关注者列表由一串 OpenID（加密后的微信号，每个用户对每个公众号的 OpenID 是唯一的）组成。一次拉取调用最多拉取 10000 个关注者的 OpenID，可以通过多次拉取的方式来满足需

求。

### 接口调用请求说明

http 请求方式: GET (请使用 https 协议)

https://api.weixin.qq.com/cgi-bin/user/get?access\_token=ACCESS\_TOKEN&next\_openid=NEXT\_OPENID

参数	是否必须	说明
access_token	是	调用接口凭证
next_openid	是	第一个拉取的 OPENID, 不填默认从头开始拉取

正确时返回 JSON 数据包:

```
{"total":2,"count":2,"data":{"openid":["","OPENID1","OPENID2"],"next_openid":"NEXT_OPENID"}
```

参数	说明
total	关注该公众账号的总用户数
count	拉取的 OPENID 个数, 最大值为 10000
data	列表数据, OPENID 的列表
next_openid	拉取列表的后一个用户的 OPENID

错误时返回 JSON 数据包 (示例为无效 AppID 错误):

```
{"errcode":40013,"errmsg":"invalid appid"}
```

### 附: 关注者数量超过 10000 时

当公众号关注者数量超过 10000 时, 可通过填写 next\_openid 的值, 从而多次拉取列表的方式来满足需求。

具体而言, 就是在调用接口时, 将上一次调用得到的返回中的 next\_openid 值, 作为下一次调用中的 next\_openid 值。

示例如下:

```
327. 公众账号 A 拥有 23000 个关注的人, 想通过拉取关注接口获取所有关注的人, 那么分别请求 url 如下:  
328. https://api.weixin.qq.com/cgi-bin/user/get?access_token=ACCESS_TOKEN  
329. 返回结果:  
330. {  
331.   "total":23000,
```

```
332. "count":10000,
333. "data":{"
334.     openid":[
335.         "OPENID1",
336.         "OPENID2",
337.         ...,
338.         "OPENID10000"
339.     ]
340. },
341. "next_openid":"NEXT_OPENID1"
342. }
343. https://api.weixin.qq.com/cgi-bin/user/get?access_token=ACCESS_TOKEN&next_openid=NEXT_O
PENID1
344. 返回结果:
345. {
346.     "total":23000,
347.     "count":10000,
348.     "data":{"
349.         "openid":[
350.             "OPENID10001",
351.             "OPENID10002",
352.             ...,
353.             "OPENID20000"
354.         ]
355.     },
356.     "next_openid":"NEXT_OPENID2"
357. }
358. https://api.weixin.qq.com/cgi-bin/user/get?access_token=ACCESS_TOKEN&next_openid=NEXT_O
PENID2
359. 返回结果（关注者列表已返回完时，返回 next_openid 为空）:
360. {
361.     "total":23000,
362.     "count":3000,
363.     "data":{"
364.         "openid":[
365.             "OPENID20001",
366.             "OPENID20002",
367.             ...,
368.             "OPENID23000"
369.         ]
370.     },
371.     "next_openid":""
372. }
```

## 四、获取用户地理位置

开通了上报地理位置接口的公众号，用户在关注后进入公众号会话时，会弹框让用户确认是否允许公众号使用其地理位置。弹框只在关注后出现一次，用户以后可以在公众号详情页面进行操作。

### 获取用户地理位置

获取用户地理位置的方式有两种，一种是仅在进入会话时上报一次，一种是进入会话后每隔 5 秒上报一次。公众号可以在公众平台网站中设置。

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，上报地理位置以推送 XML 数据包到开发者填写的 URL 来实现。

### 推送 XML 数据包示例：

```
373. <xml>
374. <ToUserName><![CDATA[toUser]]></ToUserName>
375. <FromUserName><![CDATA[fromUser]]></FromUserName>
376. <CreateTime>123456789</CreateTime>
377. <MsgType><![CDATA[event]]></MsgType>
378. <Event><![CDATA[LOCATION]]></Event>
379. <Latitude>23.137466</Latitude>
380. <Longitude>113.352425</Longitude>
381. <Precision>119.385040</Precision>
382. </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

## 五、网页授权获取用户基本信息

如果用户在微信中（Web 微信除外）访问公众号的第三方网页，公众号开发者可以通过此接口获取当前用户基本信息（包括昵称、性别、城市、国家）。利用用户信息，可以实现体验优化、用户来源统计、帐号绑定、用户身份鉴权等功能。请注意，“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。）”

本接口是通过 OAuth2.0 来完成网页授权的，是安全可靠的，关于 OAuth2.0 的详细介绍，可以参考 OAuth2.0 协议标准。在微信公众号请求用户网页授权之前，开发者需要先到公众平台网站的我的服务页中配置授权回调域名。请注意，这里填写的域名不要加 **http://**

关于配置授权回调域名的说明：

授权回调域名配置规范为全域名，比如需要网页授权的域名为：**www.qq.com**，配置以后此域名下面的页面 **http://www.qq.com/music.html** 、 **http://www.qq.com/login.html** 都可以进行 OAuth2.0 鉴权。但 **http://pay.qq.com** 、 **http://music.qq.com** 、 **http://qq.com** 无法进行 OAuth2.0 鉴权。

具体而言，网页授权流程分为三步：

- A. 引导用户进入授权页面同意授权，获取 **code**
- B. 通过 **code** 换取网页授权 **access\_token**（与基础支持中的 **access\_token** 不同）
- C. 如果需要，开发者可以刷新网页授权 **access\_token**，避免过期
- D. 通过网页授权 **access\_token** 和 **openid** 获取用户基本信息

**第一步：用户同意授权，获取 **code****

在确保微信公众账号拥有授权作用域（scope 参数）的权限的前提下（服务号获得高级接口后，默认带有 scope 参数中的 snsapi\_base 和 snsapi\_userinfo），引导关注者打开如下页面：

[https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect\\_uri=REDIRECT\\_URI&response\\_type=code&scope=SCOPE&state=STATE#wechat\\_redirect](https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=SCOPE&state=STATE#wechat_redirect)

若提示“该链接无法访问”，请检查参数是否填写错误，是否拥有 scope 参数对应的授权作用域权限。

参考链接(请在微信客户端中打开此链接体验)

**Scope 为 snsapi\_base**

[https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx520c15f417810387&redirect\\_uri=http%3A%2F%2Fchong.qq.com%2Fphp%2Findex.php%3Fd%3D%26c%3DwxAdapter%26m%3DmobileDeal%26showwxpaytitle%3D1%26vb2ctag%3D4\\_2030\\_5\\_1194\\_60&response\\_type=code&scope=snsapi\\_base&state=123#wechat\\_redirect](https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx520c15f417810387&redirect_uri=http%3A%2F%2Fchong.qq.com%2Fphp%2Findex.php%3Fd%3D%26c%3DwxAdapter%26m%3DmobileDeal%26showwxpaytitle%3D1%26vb2ctag%3D4_2030_5_1194_60&response_type=code&scope=snsapi_base&state=123#wechat_redirect)

**Scope 为 snsapi\_userinfo**

[https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx0e81c3bee622d60&redirect\\_uri=http%3A%2F%2Fnba.bluewebgame.com%2Foauth\\_response.php&response\\_type=code&scope=snsapi\\_userinfo&state=STATE#wechat\\_redirect](https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx0e81c3bee622d60&redirect_uri=http%3A%2F%2Fnba.bluewebgame.com%2Foauth_response.php&response_type=code&scope=snsapi_userinfo&state=STATE#wechat_redirect)

参数说明

参数	是否必须	说明
appid	是	公众号的唯一标识
redirect_uri	是	授权后重定向的回调链接地址
response_type	是	返回类型，请填写 code
scope	是	应用授权作用域，snsapi_base（不弹出授权页面，直接跳转，只能获取用户 openid），snsapi_userinfo（弹出授权页面，可通过 openid 拿到昵称、性别、所在地。并且，即使在未关注的情况下，只要用户授权，也能获取其信息）
state	否	重定向后会带上 state 参数，开发者可以填写任意参数值
#wechat_redirect	否	直接在微信打开链接，可以不填此参数。做页面 302 重定向时候，必须带此参数

下图为 scope 等于 snsapi\_userinfo 时的授权页面：





### 用户同意授权后

如果用户同意授权，页面将跳转至 `redirect_uri/?code=CODE&state=STATE`。若用户禁止授权，则重定向后不会带上 `code` 参数，仅会带上 `state` 参数 `redirect_uri?state=STATE`

code 说明：

code 作为换取 `access_token` 的票据，每次用户授权带上的 `code` 将不一样，`code` 只能使用一次，5 分钟未被使用自动过期。

## 第二步：通过 `code` 换取网页授权 `access_token`

首先请注意，这里通过 `code` 换取的网页授权 `access_token`，与基础支持中的 `access_token` 不同。公众号可通过下述接口来获取网页授权 `access_token`。如果网页授权的作用域为 `snsapi_base`，则本步骤中获取到网页授权

access\_token 的同时，也获取到了 openid，snsapi\_base 式的网页授权流程即到此为止。

### 请求方法

获取 code 后，请求以下链接获取 access\_token:

```
https://api.weixin.qq.com/sns/oauth2/access_token?appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code
```

### 参数说明

参数	是否必须	说明
appid	是	公众号的唯一标识
secret	是	公众号的 appsecret
code	是	填写第一步获取的 code 参数
grant_type	是	填写为 authorization_code

### 返回说明

正确时返回的 JSON 数据包如下:

```
383. {  
384.   "access_token":"ACCESS_TOKEN",  
385.   "expires_in":7200,  
386.   "refresh_token":"REFRESH_TOKEN",  
387.   "openid":"OPENID",  
388.   "scope":"SCOPE"  
389. }
```

参数	描述
access_token	网页授权接口调用凭证,注意: 此 access_token 与基础支持的 access_token 不同
expires_in	access_token 接口调用凭证超时时间, 单位 (秒)
refresh_token	用户刷新 access_token
openid	用户唯一标识, 请注意, 在未关注公众号时, 用户访问公众号的网页, 也会产生一个用户和公众号唯一的 OpenID
scope	用户授权的作用域, 使用逗号 (,) 分隔

错误时微信会返回 JSON 数据包如下 (示例为 Code 无效错误):

```
{"errcode":40029,"errmsg":"invalid code"}
```

### 第三步: 刷新 access\_token (如果需要)

由于 access\_token 拥有较短的有效期，当 access\_token 超时时，可以使用 refresh\_token 进行刷新，refresh\_token 拥有较长的有效期（7 天、30 天、60 天、90 天），当 refresh\_token 失效后，需要用户重新授权。

#### 请求方法

获取第二步的 refresh\_token 后，请求以下链接获取 access\_token：

```
https://api.weixin.qq.com/sns/oauth2/refresh_token?appid=APPID&grant_type=refresh_token&refresh_token=REFRESH_TOKEN
```

参数	是否必须	说明
appid	是	公众号的唯一标识
grant_type	是	填写为 refresh_token
refresh_token	是	填写通过 access_token 获取到的 refresh_token 参数

#### 返回说明

正确时返回的 JSON 数据包如下：

```
390. {
391.   "access_token":"ACCESS_TOKEN",
392.   "expires_in":7200,
393.   "refresh_token":"REFRESH_TOKEN",
394.   "openid":"OPENID",
395.   "scope":"SCOPE"
396. }
```

参数	描述
access_token	网页授权接口调用凭证,注意：此 access_token 与基础支持的 access_token 不同
expires_in	access_token 接口调用凭证超时时间，单位（秒）
refresh_token	用户刷新 access_token
openid	用户唯一标识
scope	用户授权的作用域，使用逗号（,）分隔

错误时微信会返回 JSON 数据包如下（示例为 Code 无效错误）：

```
{"errcode":40029,"errmsg":"invalid code"}
```

#### 第四步：拉取用户信息(需 scope 为 snsapi\_userinfo)

如果网页授权作用域为 snsapi\_userinfo，则此时开发者可以通过 access\_token 和 openid 拉取用户信息了。

### 请求方法

http: GET (请使用 https 协议)

https://api.weixin.qq.com/sns/userinfo?access\_token=ACCESS\_TOKEN&openid=OPENID

### 参数说明

参数	描述
access_token	网页授权接口调用凭证,注意:此 access_token 与基础支持的 access_token 不同
openid	用户的唯一标识

### 返回说明

正确时返回的 JSON 数据包如下:

```

397. {
398.   "openid": "OPENID",
399.   "nickname": NICKNAME,
400.   "sex": "1",
401.   "province": "PROVINCE"
402.   "city": "CITY",
403.   "country": "COUNTRY",
404.   "headimgurl":
      "http://wx.qlogo.cn/mmopen/g3MonUZtNHkdmzicIlibx6iaFqAc56vxLSUfpb6n5WKSXYVY0ChQKkia
      JSgQ1dZuTOgvLLrhJbERQQ4eMsv84eavHiaiceqxibJxCfHe/46",
405.   "privilege": [
406.     "PRIVILEGE1"
407.     "PRIVILEGE2"
408.   ]
409. }
```

参数	描述
openid	用户的唯一标识
nickname	用户昵称
sex	用户的性别, 值为 1 时是男性, 值为 2 时是女性, 值为 0 时是未知
province	用户个人资料填写的省份
city	普通用户个人资料填写的城市
country	国家, 如中国为 CN
headimgurl	用户头像, 最后一个数值代表正方形头像大小 (有 0、46、64、96、132 数值可选, 0 代表 640*640 正方形头像), 用户没有头像时该项为空

privilege	用户特权信息，json 数组，如微信沃卡用户为 (chinaunicom)
-----------	---------------------------------------

错误时微信会返回 JSON 数据包如下（示例为 openid 无效）：

```
{"errcode":40003,"errmsg":" invalid openid "}
```

## 六、网页获取用户网络状态（JS 接口）

为了方便开发者根据用户的网络状态来提供不同质量的服务，公众号可以在公众号内部的网页中使用 JavaScript 代码调用来获取网络状态。

### 接口调用代码（JavaScript）

```
410. WeixinJSBridge.invoke('getNetworkType',{  
411.           function(e){  
412.           WeixinJSBridge.log(e.err_msg);  
413.           });
```

### 返回说明

获取用户网络状态的返回值如下：

```
network_type:wifi wifi 网络  
network_type:edge 非 wifi,包含 3G/2G  
network_type:fail 网络断开连接
```

## 陆、自定义菜单

### 一、自定义菜单创建接口

自定义菜单能够帮助公众号丰富界面，让用户更好更快地理解公众号的功能。开启自定义菜单后，公众号界面如图所示：



目前自定义菜单最多包括 3 个一级菜单，每个一级菜单最多包含 5 个二级菜单。一级菜单最多 4 个汉字，二级菜单最多 7 个汉字，多出来的部分将会以“...”代替。请注意，创建自定义菜单后，由于微信客户端缓存，需要 24 小时微信客户端才会展现出来。建议测试时可以尝试取消关注公众账号后再次关注，则可以看到创建后的效果。

目前自定义菜单接口可实现两种类型按钮，如下：

**click:**

用户点击 click 类型按钮后，微信服务器会通过消息接口推送消息类型为 event 的结构给开发者（参考消息接口指南），并且带上按钮中开发者填写的 key 值，开发者可以通过自定义的 key 值与用户进行交互；

**view:**

用户点击 **view** 类型按钮后，微信客户端将会打开开发者在按钮中填写的 url 值（即网页链接），达到打开网页的目的，建议与网页授权获取用户基本信息接口结合，获得用户的登入个人信息。

### 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\\_TOK](https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOK)

EN

### 请求示例

```
414. {
415.     "button":[
416.     {
417.         "type":"click",
418.         "name":"今日歌曲",
419.         "key":"V1001_TODAY_MUSIC"
420.     },
421.     {
422.         "type":"click",
423.         "name":"歌手简介",
424.         "key":"V1001_TODAY_SINGER"
425.     },
426.     {
427.         "name":"菜单",
428.         "sub_button":[
429.         {
430.             "type":"view",
431.             "name":"搜索",
432.             "url":"http://www.soso.com/"
433.         },
434.         {
435.             "type":"view",
436.             "name":"视频",
437.             "url":"http://v.qq.com/"
438.         },
439.         {
440.             "type":"click",
441.             "name":"赞一下我们",
442.             "key":"V1001_GOOD"
443.         }
444.     ]
445. }
```

```
444.     }}
445. }
```

### 参数说明

参数	是否必须	说明
button	是	一级菜单数组，个数应为 1~3 个
sub_button	否	二级菜单数组，个数应为 1~5 个
type	是	菜单的响应动作类型，目前有 click、view 两种类型
name	是	菜单标题，不超过 16 个字节，子菜单不超过 40 个字节
key	click 类型必须	菜单 KEY 值，用于消息接口推送，不超过 128 字节
url	view 类型必须	网页链接，用户点击菜单可打开链接，不超过 256 字节

### 返回结果

正确时的返回 JSON 数据包如下：

```
{"errcode":0,"errmsg":"ok"}
```

错误时的返回 JSON 数据包如下（示例为无效菜单名长度）：

```
{"errcode":40018,"errmsg":"invalid button name size"}
```

## 二、自定义菜单查询接口

使用接口创建自定义菜单后，开发者还可使用接口查询自定义菜单的结构。

### 请求说明

http 请求方式：GET

https://api.weixin.qq.com/cgi-bin/menu/get?access\_token=ACCESS\_TOKEN

### 返回说明

对应创建接口，正确的 Json 返回结果：

```
{"menu":{"button":[{"type":"click","name":"今日歌曲",
,"key":"V1001_TODAY_MUSIC","sub_button":[]},{"type":"click","name":"歌手简介",
,"key":"V1001_TODAY_SINGER","sub_button":[]},{"name":"菜单",
,"sub_button":[{"type":"view","name":"搜索",
,"url":"http://www.soso.com/","sub_button":[]},{"type":"view","name":"视频"}]}
```



```
","url":"http://v.qq.com/","sub_button":[]},{ "type":"click","name":"赞一下我们", "key":"V1001_GOOD","sub_button":[]}]}}}
```

### 三、自定义菜单删除接口

使用接口创建自定义菜单后，开发者还可使用接口删除当前使用的自定义菜单。

#### 请求说明

http 请求方式：GET

https://api.weixin.qq.com/cgi-bin/menu/delete?access\_token=ACCESS\_TOKEN

#### 返回说明

对应创建接口，正确的 Json 返回结果：

```
{"errcode":0,"errmsg":"ok"}
```

### 四、自定义菜单事件推送

用户点击自定义菜单后，如果菜单按钮设置为 click 类型，则微信会把此次点击事件推送给开发者，注意 view 类型（跳转到 URL）的菜单点击不会上报。

#### 推送 XML 数据包示例：

```
446. <xml>  
447. <ToUserName><![CDATA[toUser]]></ToUserName>  
448. <FromUserName><![CDATA[FromUser]]></FromUserName>  
449. <CreateTime>123456789</CreateTime>  
450. <MsgType><![CDATA[event]]></MsgType>  
451. <Event><![CDATA[CLICK]]></Event>  
452. <EventKey><![CDATA[EVENTKEY]]></EventKey>  
453. </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

## 柒、推广支持

### 一、生成带参数的二维码

为了满足用户渠道推广分析的需要，公众平台提供了生成带参数二维码的接口。使用该接口可以获得多个带不同场景值的二维码，用户扫描后，公众号可以接收到事件推送。

目前有 2 种类型的二维码，分别是临时二维码和永久二维码，前者有过期时间，最大为 1800 秒，但能够生成较多数量，后者无过期时间，数量较少（目前参数只支持 1--1000）。两种二维码分别适用于帐号绑定、用户来源统计等场景。

用户扫描带场景值二维码时，可能推送以下两种事件：

- A. 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。
- B. 如果用户已经关注公众号，在用户扫描后会自动进入会话，微信也会将带场景值扫描事件推送给开发者。

获取带参数的二维码的过程包括两步，首先创建二维码 `ticket`，然后凭借 `ticket` 到指定 URL 换取二维码。

### 二、创建二维码 `ticket`

每次创建二维码 `ticket` 需要提供一个开发者自行设定的参数(`scene_id`)，分别介绍临时二维码和永久二维码的创建二维码 `ticket` 过程。

#### 临时二维码请求说明

http 请求方式: POST

URL: [https://api.weixin.qq.com/cgi-bin/qrcode/create?access\\_token=TOKEN](https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token=TOKEN)

POST 数据格式: json

POST 数据例子: {"expire\_seconds": 1800, "action\_name": "QR\_SCENE", "action\_info": {"scene": {"scene\_id": 123}}}

### 永久二维码请求说明

http 请求方式: POST

URL: https://api.weixin.qq.com/cgi-bin/qrcode/create?access\_token=TOKEN

POST 数据格式: json

POST 数据例子: {"action\_name": "QR\_LIMIT\_SCENE", "action\_info": {"scene": {"scene\_id": 123}}}

### 参数说明

参数	说明
expire_seconds	该二维码有效时间, 以秒为单位。最大不超过 1800。
action_name	二维码类型, QR_SCENE 为临时, QR_LIMIT_SCENE 为永久
action_info	二维码详细信息
scene_id	场景值 ID, 临时二维码时为 32 位非 0 整型, 永久二维码时最大值为 1000 (目前参数只支持 1--1000)

### 返回说明

正确的 Json 返回结果:

```
{"ticket": "gQG28DoAAAAAAAAAASxodHRwOi8vd2VpeGluLnFmNvbS9xL0FuWC1DNmZuVEhvMVp4NDNDNRnNRAAIEesLvUQMECAcAAA==", "expire_seconds": 1800}
```

参数	说明
ticket	获取的二维码 ticket, 凭借此 ticket 可以在有效期内换取二维码。
expire_seconds	二维码的有效时间, 以秒为单位。最大不超过 1800。

错误的 Json 返回示例:

```
{"errcode": 40013, "errmsg": "invalid appid"}
```

## 三、通过 ticket 换取二维码

获取二维码 ticket 后, 开发者可用 ticket 换取二维码图片。请注意, 本接口无须登录态即可调用。

### 请求说明

HTTP GET 请求 (请使用 https 协议)

```
https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=TICKET
```

**提醒：TICKET 记得进行 UriEncode**

## 返回说明

ticket 正确情况下，http 返回码是 200，是一张图片，可以直接展示或者下载。

HTTP 头（示例）如下：

Accept-Ranges:bytes

Cache-control:max-age=604800

Connection:keep-alive

Content-Length:28026

Content-Type:image/jpg

Date:Wed, 16 Oct 2013 06:37:10 GMT

Expires:Wed, 23 Oct 2013 14:37:10 +0800

Server:nginx/1.4.1

错误情况下（如 ticket 非法）返回 HTTP 错误码 404。

## 捌、Winxin JS 接口

### 一、隐藏微信中网页右上角按钮

公众号在有需要时（如不需要用户分享某个页面），可在网页中通过 JavaScript 代码隐藏网页右上角按钮。



## 接口调用代码 (JavaScript)

```
454. document.addEventListener('WeixinJSBridgeReady', function onBridgeReady() {  
455. WeixinJSBridge.call('hideOptionsMenu');  
456. });
```

### 返回说明

隐藏底部导航栏没有返回值。（需要显示请把 hideOptionsMenu 换成 showOptionsMenu）

## 二、隐藏微信中网页底部导航栏

公众号在有需要时(如认为用户在该页面不会用到浏览器前进后退功能)，可在网页中通过 JavaScript 代码隐藏网页底部导航栏。



### 接口调用代码（JavaScript）

```
457. document.addEventListener('WeixinJSBridgeReady', function onBridgeReady() {  
458. WeixinJSBridge.call('hideToolbar');  
459. });
```

### 返回说明

隐藏底部导航栏没有返回值。（需要显示顶部导航栏，请把 hideToolbar

换成 showToolbar）

## 三、网页获取用户网络状态

为了方便开发者根据用户的网络状态来提供不同质量的服务，公众号可以在公众号内部的网页中使用 JavaScript 代码调用来获取网络状态。

### 接口调用代码（JavaScript）

```
460. WeixinJSBridge.invoke('getNetworkType',{  
461. function(e){  
462.   WeixinJSBridge.log(e.err_msg);  
463. });
```

## 返回说明

获取用户网络状态的返回值如下：

```
network_type:wifi wifi 网络  
network_type:edge 非 wifi,包含 3G/2G  
network_type:fail 网络断开连接  
network_type:wwan (2g 或者 3g)
```