微信公众平台开发概述

微信公众平台是运营者通过公众号为微信用户提供资讯和服务的平台，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。

为了识别用户，每个用户针对每个公众号会产生一个安全的OpenID，如果需要在多公众号、移动应用之间做用户共通，则需前往微信开放平台，将这些公众号和应用绑定到一个开放平台账号下，绑定后，一个用户虽然对多个公众号和应用有多个不同的OpenID，但他对所有这些同一开放平台账号下的公众号和应用，只有一个UnionID，可以在用户管理-获取用户基本信息（UnionID机制）文档了解详情。

请开发者注意：

|  |
| --- |
| 1、微信公众平台开发是指为微信公众号进行业务开发，为移动应用、PC端网站、公众号第三方平台（为各行各业公众号运营者提供服务）的开发，请前往微信开放平台接入。  2、在申请到认证公众号之前，你可以先通过测试号申请系统，快速申请一个接口测试号，立即开始接口测试开发。  3、在开发过程中，可以使用接口调试工具来在线调试某些接口。  4、每个接口都有每日接口调用频次限制，可以在公众平台官网-开发者中心处查看具体频次。  5、在开发出现问题时，可以通过接口调用的返回码，以及报警排查指引（在公众平台官网-开发者中心处可以设置接口报警），来发现和解决问题。  6、公众平台以access\_token为接口调用凭据，来调用接口，所有接口的调用需要先获取access\_token，access\_token在2小时内有效，过期需要重新获取，但1天内获取次数有限，开发者需自行存储，详见获取接口调用凭据（access\_token）文档。  7、公众平台接口调用仅支持80端口。 |

公众号主要通过公众号消息会话和公众号内网页来为用户提供服务的，下面分别介绍这两种情况：

**1、公众号消息会话**

公众号是以微信用户的一个联系人形式存在的，消息会话是公众号与用户交互的基础。目前公众号内主要有这样几类消息服务的类型，分别用于不同的场景。

|  |
| --- |
| 1）群发消息：公众号可以以一定频次（订阅号为每天1次，服务号为每月4次），向用户群发消息，包括文字消息、图文消息、图片、视频、语音等。  2）被动回复消息：在用户给公众号发消息后，微信服务器会将消息发到开发者预先在开发者中心设置的服务器地址（开发者需要进行消息真实性验证），公众号可以在5秒内做出回复，可以回复一个消息，也可以回复命令告诉微信服务器这条消息暂不回复。被动回复消息可以设置加密（在公众平台官网的开发者中心处设置，设置后，按照消息加解密文档来进行处理。其他3种消息的调用因为是API调用而不是对请求的返回，所以不需要加解密）。  3）客服消息：在用户给公众号发消息后的48小时内，公众号可以给用户发送不限数量的消息，主要用于客服场景。用户的行为会触发事件推送，某些事件推送是支持公众号据此发送客服消息的，详见微信推送消息与事件说明文档。  4）模板消息：在需要对用户发送服务通知（如刷卡提醒、服务预约成功通知等）时，公众号可以用特定内容模板，主动向用户发送消息。 |

**2、公众号内网页**

许多复杂的业务场景，需要通过网页形式来提供服务，这时需要用到：

|  |
| --- |
| 1）网页授权获取用户基本信息：通过该接口，可以获取用户的基本信息（获取用户的OpenID是无需用户同意的，获取用户的基本信息则需用户同意）  2）微信JS-SDK：是开发者在网页上通过JavaScript代码使用微信原生功能的工具包，开发者可以使用它在网页上录制和播放微信语音、监听微信分享、上传手机本地图片、拍照等许多能力。 |

**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。 |

更新日志

2017年6月5日

图文消息正文支持插入自己帐号和其他公众号已群发的图文消息超链接。

2017年5月15日

新增图文消息留言管理接口。

2017年5月2日

删除群发接口新增 article\_idx 参数，开发者可以指定删除多图文中的某一篇。并新增返回码 40060，代表指定的 article\_idx 不合法。

群发接口每分钟限制请求60次，超过限制的请求会被拒绝。

2017年4月21日

群发接口新增 clientmsgid 参数，开发者调用群发接口时可以主动设置，避免重复推送。

2017年3月22日

用户标签管理接口更新。标签功能目前支持公众号为用户打上最多20个标签。

2016年12月27日

callback 被动回复用户消息时，返回的多图文消息条数上限调整为8条。

2016年8月17日

新增黑名单管理接口。

2016年4月29日

公众平台推出用户标签接口，涉及到的更新包括：用户标签管理、获取用户基本信息(UnionID机制)、个性化菜单接口、高级群发接口等。

2016年4月19日

微信浏览器全面升级至X5 Blink内核，web 开发者工具同步推出0.5.0版本，支持远程调试 X5 Blink 内核上的网页。

2016年3月28日

为帮助公众号及第三方开发者解决接口调用频率受限的问题，我们向开发者提供了接口调用次数清零的接口，每个自然月可调用10次。

2016年3月2日

为帮助开发者解决群发失误问题，使用群发接口群发后的消息，随时可以删除，而不再是以前的半小时内可删除。

2016年1月22日

为帮助网页开发者实现与微信客户端一致的视觉体验，并降低设计和开发成本，微信团队推出了网页设计样式库：WeUI。

2016年1月11日

公众平台发布微信 web 开发者工具，可以在 PC 或 Mac 上模拟访问微信内网页，帮助开发者更方便地进行开发和调试。

2016年1月6日

在素材管理接口中的获取图文消息素材时，为了防止图文消息素材的封面图片media\_id失效，增加提供了封面图片URL字段的输出。第三方开发者也可以使用这个URL下载图片到自己服务器中，然后显示在自己网站上。

2015年12月16日

为了帮助公众号实现灵活的业务运营，微信公众平台新增了个性化菜单接口，开发者可以通过该接口，让公众号的不同用户群体看到不一样的自定义菜单。该接口开放给已认证订阅号和已认证服务号。

2015年12月2日

为了增加带参数二维码的使用场景，临时带参数二维码最长可设置的有效期增长为30天。

2015年11月18日

客服接口增加支持发送图文消息（跳转到图文消息页）的接口。此前客服接口已经支持发送跳转到外部链接的图文消息。

2015年10月9日

微信公众平台的微信认证涉及开发者接口的赋予，为了帮助公众号开发者获取公众号的认证状态，微信公众平台开始提供公众号认证过程中各阶段的事件推送，会推送到在开发者中心填写的服务器地址中。

2015年8月12日

微信连Wi-Fi新增获取公众号连网URL接口，获取二维码物料接口中更新了二维码物料样式，添加设备接口允许第三方的ssid包含中文。

2015年7月31日

公众平台增加了群发图文消息时使用的图片上传接口。开发者可以通过该接口上传图片，获得图片链接并放置到图文消息中，然后使用群发接口群发给用户。该接口开放给所有认证公众号使用。同时，为了保证安全性，从8月15日开始，所有图文消息页面中的外链图片将被去除（8月15日之前的图文消息不受影响，外链图片仍会显示），请开发者尽快替代为使用上述新接口。

2015年7月28日

群发接口在群发图文消息时，增加了msg\_data\_id的返回，可以用于在群发后使用图文分析接口获取图文消息的数据。 卡券接口部分新增数据接口，会员卡信息查询接口（含积分查询）。

2015年7月2日

为了保证开发者迁移公众号业务到第三方平台时的便利性，使用获取自动回复配置接口和获取菜单配置接口获取公众号配置时，图文消息的配置将增加mediaID信息的输出。

2015年6月29日

为了帮助开发者提高效率，用户管理接口中新增了批量获取用户基本信息的接口。

2015年6月17日

摇一摇周边新增申请开通摇一摇周边接口，新增摇一摇事件通知接口，新增查询设备ID申请审核状态接口。优化修改小部分文案。

2015年6月15日

为了满足第三方平台开发者的需求，发送消息-群发接口中的预览接口，增加了对微信号发送预览的能力（此前只能对openID发送预览）。

2015年5月28日

素材管理-获取永久素材接口，在获取图文消息素材时，接口返回数据字段中增加了URL字段，便于开发者直接使用图文消息文章页。在获取图片素材时，直接返回图片URL，便于开发者使用。但请注意，公众平台的图片仅能在腾讯系内域名使用，在其他域名使用将被禁止。

2015年5月19日

向认证的政府与媒体类订阅号开放网页授权接口。

2015年4月29日

自定义菜单接口和素材管理接口，向第三方平台旗下未认证订阅号开放。

2015年4月22日

帐号管理-带参数二维码接口更新，临时二维码的最长有效时间从半小时提升至7天，并且永久二维码的场景值参数可以是字符串形式。

2015年4月14日

新增获取自动回复和自定义菜单配置接口，对所有公众号开放。对于接入了公众号登录授权的第三方平台，在公众号授权给开发者后，开发者应当通过上述接口获取公众号的配置，为其重新配置，以保证公众号运营者的业务运营体验。

2015年4月11日

门店相关接口已更换，旧门店接口将于5月正式停用，请尽快查看最新平台文档，及时切换新接口。

2015年3月18日

公众平台新增素材管理接口，对所有认证公众号开放。该接口的开放，可以帮助公众号开发者更方便地进行素材管理，也可以帮助第三方平台实现与公众平台官网互通的素材管理功能。

2015年3月13日

微信公众平台接口测试账号是为满足公众号开发者了解平台接口能力以及新功能开发上线前进行测试，其地址为http://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login，本次更新让测试帐号获得了与认证服务号相同的能力，包括JS-SDK相关能力。

2015年2月3日

用户管理-用户分组管理接口中，修改用户分组增加了批量操作的接口能力。

更多更早的开发者和接口方面的更新，请参见[微信公众平台公告](https://mp.weixin.qq.com/cgi-bin/announce?action=getannouncementlist&lang=zh_CN)和[微信开放平台公告](https://open.weixin.qq.com/cgi-bin/announce?action=getannouncementlist&lang=zh_CN&token=)。

## 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。

公众号接口权限说明

不同的公众号类型具备不同的接口权限，具体如下表： 请注意：

1、微博认证视作未认证,因此微博认证的公众号不会拥有微信认证公众号特有的接口。

2、微信认证分为资质认证和名称认证两部分，只需要资质认证通过，就可获得接口。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 未认证订阅号 | 微信认证订阅号 | 未认证服务号 | 微信认证服务号 |
| 基础支持-获取access\_token | 有 | 有 | 有 | 有 |
| 基础支持-获取微信服务器IP地址 | 有 | 有 | 有 | 有 |
| 接收消息-验证消息真实性、接收普通消息、接收事件推送、接收语音识别结果 | 有 | 有 | 有 | 有 |
| 发送消息-被动回复消息 | 有 | 有 | 有 | 有 |
| 发送消息-客服接口 |  | 有 |  | 有 |
| 发送消息-群发接口 |  | 有 |  | 有 |
| 发送消息-模板消息接口（发送业务通知） |  |  |  | 有 |
| 用户管理-用户分组管理 |  | 有 |  | 有 |
| 用户管理-设置用户备注名 |  | 有 |  | 有 |
| 用户管理-获取用户基本信息 |  | 有 |  | 有 |
| 用户管理-获取用户列表 |  | 有 |  | 有 |
| 用户管理-获取用户地理位置 |  |  |  | 有 |
| 用户管理-网页授权获取用户openid/用户基本信息 |  |  |  | 有 |
| 推广支持-生成带参数二维码 |  |  |  | 有 |
| 推广支持-长链接转短链接口 |  |  |  | 有 |
| 界面丰富-自定义菜单 |  | 有 | 有 | 有 |
| 素材管理-素材管理接口 |  | 有 |  | 有 |
| 智能接口-语义理解接口 |  |  |  | 有 |
| 多客服-获取多客服消息记录、客服管理 |  |  |  | 有 |
| 微信支付接口 |  |  |  | 需申请 |
| 微信小店接口 |  |  |  | 需申请 |
| 微信卡券接口 |  | 需申请 |  | 需申请 |
| 微信设备功能接口 |  |  |  | 需申请 |
| 微信JS-SDK-基础接口 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-分享接口 |  | 有 |  | 有 |
| 微信JS-SDK-图像接口 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-音频接口 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-智能接口（网页语音识别） | 有 | 有 | 有 | 有 |
| 微信JS-SDK-设备信息 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-地理位置 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-界面操作 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-微信扫一扫 | 有 | 有 | 有 | 有 |
| 微信JS-SDK-微信小店 |  |  |  | 有 |
| 微信JS-SDK-微信卡券 |  | 有 |  | 有 |
| 微信JS-SDK-微信支付 |  |  |  | 有 |

接口调用频次限制说明

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"}

开发者可以登录微信公众平台，在帐号后台开发者中心接口权限模板查看帐号各接口当前的日调用上限和实时调用量，对于认证帐号可以对实时调用量清零，说明如下：

1、由于指标计算方法或统计时间差异，实时调用量数据可能会出现误差，一般在1%以内。

2、每个帐号每月共10次清零操作机会，清零生效一次即用掉一次机会（10次包括了平台上的清零和调用接口API的清零）。

3、第三方帮助公众号调用时，实际上是在消耗公众号自身的quota。

4、每个有接口调用限额的接口都可以进行清零操作。

**新注册帐号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 |
| 高级群发接口 | 100 |
| 上传图文消息接口 | 10 |
| 删除图文消息接口 | 10 |
| 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 | 无 |
| 刷新网页授权access\_token | 无 |
| 网页授权获取用户信息 | 无 |
| 设置用户备注名 | 10000 |

**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 |
| 素材管理-临时素材上传 | 500 |
| 素材管理-临时素材下载 | 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 | 无 |
| 刷新网页授权access\_token | 无 |
| 网页授权获取用户信息 | 无 |

**公众号调用或第三方平台帮公众号调用对公众号的所有api调用（包括第三方帮其调用）次数进行清零：**

|  |  |
| --- | --- |
| 1  2  3 | HTTP请求：POST  HTTP调用：  https://api.weixin.qq.com/cgi-bin/clear\_quota?access\_token=ACCESS\_TOKEN |

**调用示例：**

|  |  |
| --- | --- |
| 1  2  3 | {  “appid”:“APPID”  } |

**参数说明：**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 调用接口凭据 |
| appid | 是 | 公众号的APPID |

正常情况下，会返回：

|  |  |
| --- | --- |
| 1  2  3  4 | {  "errcode":0,  "errmsg":"ok"  } |

如果调用超过限制次数，则返回：

|  |  |
| --- | --- |
| 1  2  3  4 | {  "errcode":48006,  "errmsg":"forbid to clear quota because of reaching the limit"  } |

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，此时请开发者稍候再试 |
| 0 | 请求成功 |
| 40001 | 获取access\_token时AppSecret错误，或者access\_token无效。请开发者认真比对AppSecret的正确性，或查看是否正在为恰当的公众号调用接口 |
| 40002 | 不合法的凭证类型 |
| 40003 | 不合法的OpenID，请开发者确认OpenID（该用户）是否已关注公众号，或是否是其他公众号的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，请开发者检查AppID的正确性，避免异常字符，注意大小写 |
| 40014 | 不合法的access\_token，请开发者认真比对access\_token的有效性（如是否过期），或查看是否正在为恰当的公众号调用接口 |
| 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 |
| 40060 | 删除单篇图文时，指定的 article\_idx 不合法 |
| 40117 | 分组名字不合法 |
| 40118 | media\_id大小不合法 |
| 40119 | button类型错误 |
| 40120 | button类型错误 |
| 40121 | 不合法的media\_id类型 |
| 40132 | 微信号不合法 |
| 40137 | 不支持的图片格式 |
| 40155 | 请勿添加其他公众号的主页链接 |
| 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，请检查access\_token的有效期，请参考基础支持-获取access\_token中，对access\_token的详细机制说明 |
| 42002 | refresh\_token超时 |
| 42003 | oauth\_code超时 |
| 42007 | 用户修改微信密码，accesstoken和refreshtoken失效，需要重新授权 |
| 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 |
| 43019 | 需要将接收者从黑名单中移除 |
| 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 |
| 45011 | API调用太频繁，请稍候再试 |
| 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 |
| 45047 | 客服接口下行条数超过上限 |
| 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，请确认公众号已获得该接口，可以在公众平台官网-开发者中心页中查看接口权限 |
| 48002 | 粉丝拒收消息（粉丝在公众号选项中，关闭了“接收消息”） |
| 48004 | api接口被封禁，请登录mp.weixin.qq.com查看详情 |
| 48005 | api禁止删除被自动回复和自定义菜单引用的素材 |
| 48006 | api禁止清零调用次数，因为清零次数达到上限 |
| 50001 | 用户未授权该api |
| 50002 | 用户受限，可能是违规后接口被封禁 |
| 61451 | 参数错误(invalid parameter) |
| 61452 | 无效客服账号(invalid kf\_account) |
| 61453 | 客服帐号已存在(kf\_account exsited) |
| 61454 | 客服帐号名长度超过限制(仅允许10个英文字符，不包括@及@后的公众号的微信号)(invalid   kf\_acount length) |
| 61455 | 客服帐号名包含非法字符(仅允许英文+数字)(illegal character in     kf\_account) |
| 61456 | 客服帐号个数超过限制(10个客服账号)(kf\_account count exceeded) |
| 61457 | 无效头像文件类型(invalid   file type) |
| 61450 | 系统错误(system error) |
| 61500 | 日期格式错误 |
| 65301 | 不存在此menuid对应的个性化菜单 |
| 65302 | 没有相应的用户 |
| 65303 | 没有默认菜单，不能创建个性化菜单 |
| 65304 | MatchRule信息为空 |
| 65305 | 个性化菜单数量受限 |
| 65306 | 不支持个性化菜单的帐号 |
| 65307 | 个性化菜单信息为空 |
| 65308 | 包含没有响应类型的button |
| 65309 | 个性化菜单开关处于关闭状态 |
| 65310 | 填写了省份或城市信息，国家信息不能为空 |
| 65311 | 填写了城市信息，省份信息不能为空 |
| 65312 | 不合法的国家信息 |
| 65313 | 不合法的省份信息 |
| 65314 | 不合法的城市信息 |
| 65316 | 该公众号的菜单设置了过多的域名外跳（最多跳转到3个域名的链接） |
| 65317 | 不合法的URL |
| 9001001 | POST数据参数不合法 |
| 9001002 | 远端服务不可用 |
| 9001003 | Ticket不合法 |
| 9001004 | 获取摇周边用户信息失败 |
| 9001005 | 获取商户信息失败 |
| 9001006 | 获取OpenID失败 |
| 9001007 | 上传文件缺失 |
| 9001008 | 上传素材的文件类型不合法 |
| 9001009 | 上传素材的文件尺寸不合法 |
| 9001010 | 上传失败 |
| 9001020 | 帐号不合法 |
| 9001021 | 已有设备激活率低于50%，不能新增设备 |
| 9001022 | 设备申请数不合法，必须为大于0的数字 |
| 9001023 | 已存在审核中的设备ID申请 |
| 9001024 | 一次查询设备ID数量不能超过50 |
| 9001025 | 设备ID不合法 |
| 9001026 | 页面ID不合法 |
| 9001027 | 页面参数不合法 |
| 9001028 | 一次删除页面ID数量不能超过10 |
| 9001029 | 页面已应用在设备中，请先解除应用关系再删除 |
| 9001030 | 一次查询页面ID数量不能超过50 |
| 9001031 | 时间区间不合法 |
| 9001032 | 保存设备与页面的绑定关系参数错误 |
| 9001033 | 门店ID不合法 |
| 9001034 | 设备备注信息过长 |
| 9001035 | 设备申请参数不合法 |
| 9001036 | 查询起始值begin不合法 |

## 入门指引

目录

[1       开启公众号开发者模式](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1)

[1.1        申请服务器](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1.1)

[1.2        搭建服务](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1.2)

[1.3        申请公众号](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1.3)

[1.4        开发者基本配置](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1.4)

[1.5        重要事情提前交代](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#1.5)

[2       实现“你问我答”](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2)

[2.1        接受文本消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.1)

[2.2        被动回复文本消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.2)

[2.3        回复success问题](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.3)

[2.4        流程图](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.4)

[2.5        码代码](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.5)

[2.6        在线测试](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.6)

[2.7        真实体验](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#2.7)

[3       实现“图”尚往来](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#3)

[3.1        接收图片消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#3.1)

[3.2        被动回复图片消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#3,2)

[3.3        流程图](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#3.3)

[3.4        码代码](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#3.4)

[4       AccessToken](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4)

[4.1        查看appid  及appsecret](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4.1)

[4.2        获取accessToken](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4.2)

[4.2.1         临时方法获取](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4.2.1)

[4.2.2         接口获取](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4.2.2)

[4.3        码代码](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#4.3)

[5       临时素材](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5)

[5.1        新建临时素材](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5.1)

[5.2        获取临时素材MediaID](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5.2)

[5.3        下载临时素材](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5.3)

[5.3.1         手工体验](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5.3.1)

[5.3.2         接口实现](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#5.3.2)

[6       永久素材](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6)

[6.1        新建永久素材的方式](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.1)

[6.1.1         手工体验](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.1.1)

[6.1.2         接口实现](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.1.2)

[6.2        获取永久素材MediaID](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.2)

[6.3        获取素材列表](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.3)

[6.4        删除永久素材](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#6.4)

[7       自定义菜单](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7)

[7.1        创建菜单界面](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7.1)

[7.2        完善菜单功能](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7.2)

[7.2.1         流程图](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7.2.1)

[7.2.2         码代码](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7.2.2)

[7.3        体验](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#7.3)

[8       关于反馈问题](https://mp.weixin.qq.com/wiki?action=doc&id=mp1472017492_58YV5&t=0.9127495329969959#8)

# 1开启公众号开发者模式

公众平台的技术文档目的为了简明扼要的交代接口的使用，语句难免苦涩难懂，甚至对于不同的读者，有语意歧义。万事皆是入门难，对于刚入门的开发者讲，更是难上加难，往往看了半天wiki，就是不懂说的什么鬼。

为了降低门槛，弥补不足，特地编写了《开发者指引》，讲解了一下微信开放平台的基础常见功能，目的重在帮助大家入门微信开放平台的开发者模式。

对于已熟知或者有一定公众平台开发经验的开发小哥，请直接跳过本文。这篇文章不会给你带来厉害的编码技巧亦或接口的深层次讲解。对于现有接口存在的疑问，请直接呼叫客服或者微信投诉，不要在本文浪费时间。

为了防止小编语意上歧义，在给大家带来困扰，直接用最暴力的方法带你入门公众号开发——附录简明代码

## 1.1   申请服务器

以腾讯云服务器为示例：[腾讯云服务器购买入口](https://buy.qcloud.com/cvm?cpu=1&mem=1)，购买指导请参考[快速入门linux云服务器](http://www.qcloud.com/doc/product/213/%E6%AD%A5%E9%AA%A4%E4%B8%80%EF%BC%9A%E6%B3%A8%E5%86%8C%E8%85%BE%E8%AE%AF%E4%BA%91%E5%B8%90%E5%8F%B7)。

学生党注意：腾讯公司为在读高校生提供了[云+校园计划](http://bbs.qcloud.com/forum.php?mod=viewthread&tid=21113&highlight=%E6%A0%A1%E5%9B%AD)，1元/月即可使用腾讯云。

## 1.2   搭建服务

以web.py网络框，python，腾讯云服务器为例介绍。

1）安装/更新需要用到的软件

    安装python2.7版本以上

    安装web.py

    安装libxml2, libxslt, lxml python

2）编辑代码，如果不懂python 语法，请到[python官方文档](https://docs.python.org/2/)查询说明。vim main.py

# -\*- coding: utf-8 -\*-

# filename: main.py

import web

urls = (

    '/wx', 'Handle',

)

class Handle(object):

    def GET(self):

        return "hello, this is a test"

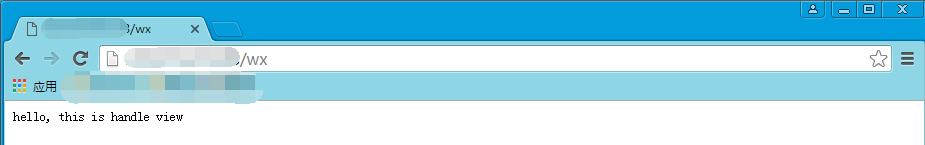
if \_\_name\_\_ == '\_\_main\_\_':

    app = web.application(urls, globals())

    app.run()

3）如果出现“socket.error: No socket could be created“错误信息，可能为80端口号被占用，可能是没有权限，请自行查询解决办法。如果遇见其他错误信息，请到 [web.py官方文档](http://www.webpy.org/) 学习webpy 框架3）执行命令：sudo python main.py 80 。

4）浏览器输入http://外网IP:80/wx （外网IP请到腾讯云购买成功处查询）。如下图，一个简单的web应用已搭建。



## 1.3   申请公众号

[申请公众号网页](https://mp.weixin.qq.com/cgi-bin/loginpage?t=wxm2-login&lang=zh_CN)



邮箱激活后，选择公众号类型。不同的公众号拥有不同的能力，详情请见wiki：[公众号接口权限说明](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1433401084&token=&lang=zh_CN) 。当然，服务号、企业号需要一定的证件和相关资料填写，如果证件一时不能准备好，没关系，公众号其实已注册，下次可以根据此邮箱&密码登录再选择。



## 1.4   开发者基本配置

1） 公众平台官网登录之后，找到“基本配置”菜单栏



2） 填写配置

url填写：http://外网IP：端口号/wx 。外网IP请到腾讯云购买成功处查询， http的端口号固定使用80，不可填写其他。

Token：自主设置，这个token与公众平台wiki中常提的access\_token不是一回事。这个token只用于验证开发者服务器。



3） 现在选择提交肯定是验证token失败，因为还需要完成代码逻辑。改动原先main.py文件，新增handle.py

#### a）vim main.py

# -\*- coding: utf-8 -\*-

# filename: main.py

import web

from handle import Handle

urls = (

    '/wx', 'Handle',

)

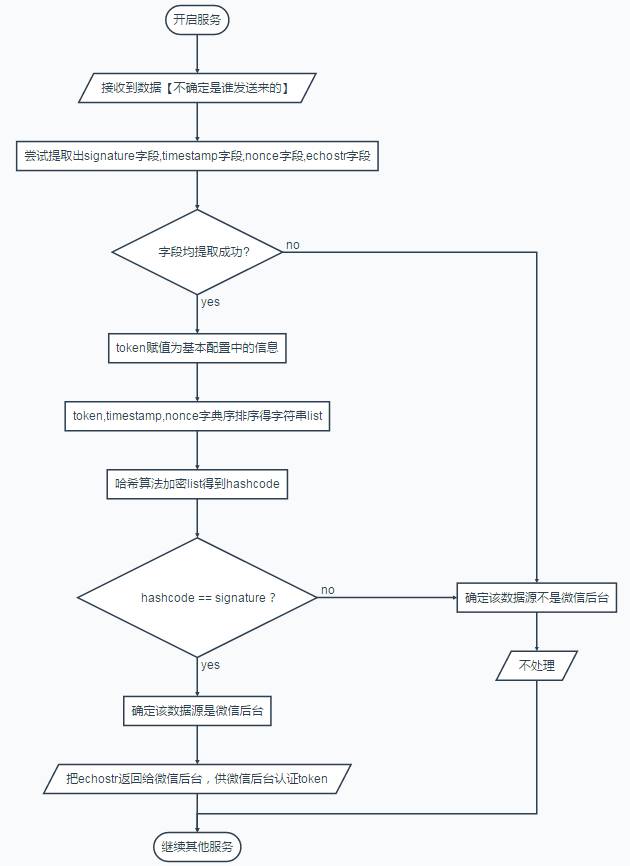
if \_\_name\_\_ == '\_\_main\_\_':

    app = web.application(urls, globals())

    app.run()

#### b）vim handle.py

       先附加逻辑流程图



# -\*- coding: utf-8 -\*-

# filename: handle.py

import hashlib

import web

class Handle(object):

    def GET(self):

        try:

            data = web.input()

            if len(data) == 0:

                return "hello, this is handle view"

            signature = data.signature

            timestamp = data.timestamp

            nonce = data.nonce

            echostr = data.echostr

            token = "xxxx" #请按照公众平台官网\基本配置中信息填写

            list = [token, timestamp, nonce]

            list.sort()

            sha1 = hashlib.sha1()

            map(sha1.update, list)

            hashcode = sha1.hexdigest()

            print "handle/GET func: hashcode, signature: ", hashcode, signature

            if hashcode == signature:

                return echostr

            else:

                return ""

        except Exception, Argument:

            return Argument

4） 重新启动成功后（python main.py 80），点击提交按钮。若提示”token验证失败”, 请认真检查代码或网络链接等。若token验证成功，会自动返回基本配置的主页面，点击启动按钮

## 1.5   重要事情提前交代

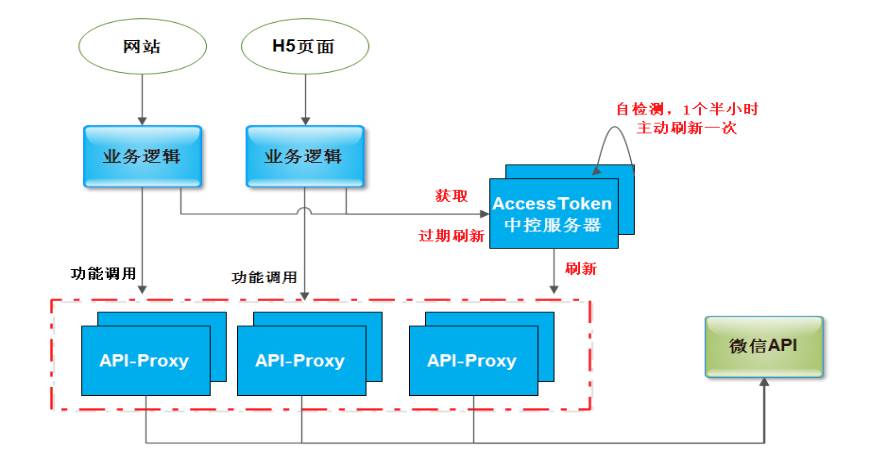
接下来，文章准备从两个简单的示例入手。

示例一：实现“你说我学”

示例一：实现“图尚往来”

两个简单的示例后，是一些基础功能的介绍：素材管理、自定义菜单、群发。所有的示例代码是为了简明的说明问题，避免代码复杂化。

在实际中搭建一个安全稳定高效的公众号，建议参考框架如下图：



主要有三个部分：负责业务逻辑部分的服务器，负责对接微信API的API-Proxy服务器，以及唯一的AccessToken中控服务器

1）AccessToken中控服务器：

负责： 提供主动刷新和被动刷新机制来刷新accessToken并存储（为了防止并发刷新，注意加并发锁），提供给业务逻辑有效的accessToken。

优点： 避免业务逻辑方并发获取access\_token，避免AccessToken互相覆盖，提高业务功能的稳定性。

       2）API-Proxy服务器：

负责：专一与微信API对接，不同的服务器可以负责对接不同的业务逻辑，更可进行调用频率、权限限制。

优点：某台API-proxy异常，还有其余服务器支持继续提供服务，提高稳定性，

避免直接暴漏内部接口，有效防止恶意攻击，提高安全性。

# 2       实现“你问我答”

    目的：

1）理解被动消息的含义

2）理解收\发消息机制

    预实现功能：

粉丝给公众号一条文本消息，公众号立马回复一条文本消息给粉丝，不需要通过公众平台网页操作。

## 2.1   接受文本消息

即粉丝给公众号发送的文本消息。官方wiki链接：[消息管理/接收消息-接受普通消息](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140453&token=&lang=zh_CN)

粉丝给公众号发送文本消息：“欢迎开启公众号开发者模式”，在开发者后台，收到公众平台发送的xml 如下：（下文均隐藏了ToUserName 及 FromUserName 信息）

<xml>

<ToUserName><![CDATA[公众号]]></ToUserName>  
<FromUserName><![CDATA[粉丝号]]></FromUserName>  
<CreateTime>1460537339</CreateTime>  
<MsgType><![CDATA[text]]></MsgType>  
<Content><![CDATA[欢迎开启公众号开发者模式]]></Content>  
<MsgId>6272960105994287618</MsgId>  
</xml>

解释：

createTime 是微信公众平台记录粉丝发送该消息的具体时间

text: 用于标记该xml 是文本消息，一般用于区别判断

欢迎开启公众号开发者模式: 说明该粉丝发给公众号的具体内容是欢迎开启公众号开发者模式

MsgId: 是公众平台为记录识别该消息的一个标记数值, 微信后台系统自动产生

## 2.2   被动回复文本消息

即公众号给粉丝发送的文本消息，官方wiki链接: [消息管理/接收消息-被动回复消息](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140543&token=&lang=zh_CN)

特别强调：

1） 被动回复消息，即发送被动响应消息，不同于客服消息接口

2） 它其实并不是一种接口，而是对微信服务器发过来消息的一次回复

3） 收到粉丝消息后不想或者不能5秒内回复时，需回复“success”字符串（下文详细介绍）

4） 客服接口在满足一定条件下随时调用

公众号想回复给粉丝一条文本消息，内容为“test”, 那么开发者发送给公众平台后台的xml 内容如下：

 <xml>  
<ToUserName><![CDATA[粉丝号]]></ToUserName>  
<FromUserName><![CDATA[公众号]]></FromUserName>  
<CreateTime>1460541339</CreateTime>  
<MsgType><![CDATA[text]]></MsgType>  
<Content><![CDATA[test]]></Content>  
</xml>

特别备注：

1）ToUserName（接受者）、FromUserName(发送者) 字段请实际填写。

2）createtime 只用于标记开发者回复消息的时间，微信后台发送此消息都是不受这个字段约束。

3）text : 用于标记 此次行为是发送文本消息 （当然可以是image/voice等类型）。

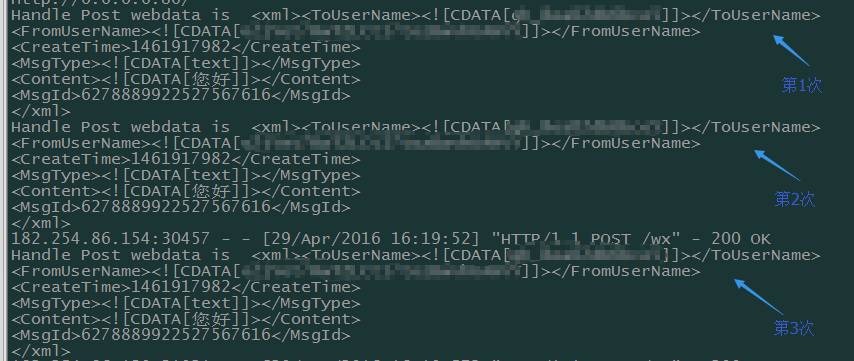
4）文本换行 ‘\n’。

## 2.3   回复success问题

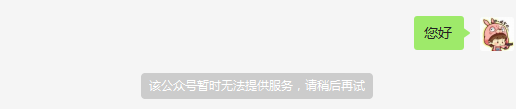
查询官方wiki 开头强调： 假如服务器无法保证在五秒内处理回复，则必须回复“success”或者“”（空串），否则微信后台会发起三次重试。

解释一下为何有这么奇怪的规定。发起重试是微信后台为了尽可以保证粉丝发送的内容开发者均可以收到。如果开发者不进行回复，微信后台没办法确认开发者已收到消息，只好重试。

真的是这样子吗？尝试一下收到消息后，不做任何回复。在日志中查看到微信后台发起了三次重试操作，日志截图如下：

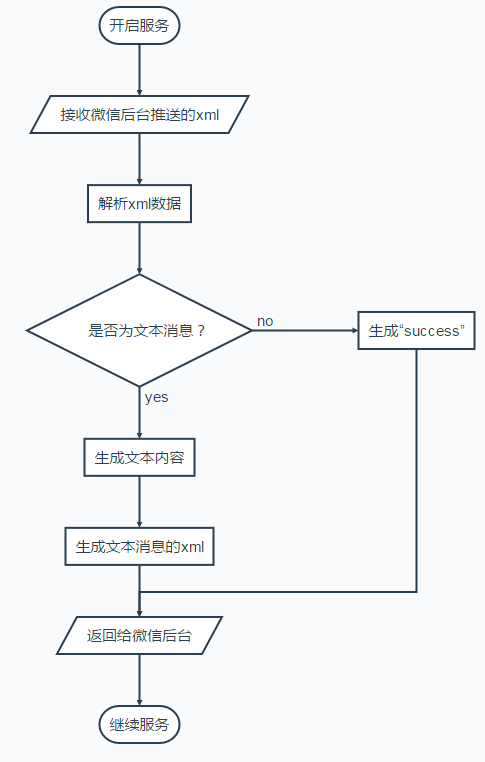


三次重试后，依旧没有及时回复任何内容，系统自动在粉丝会话界面出现错误提示“该公众号暂时无法提供服务，请稍后再试”。



如果回复success，微信后台可以确定开发者收到了粉丝消息，没有任何异常提示。因此请大家注意回复success的问题。

## 2.4   流程图



## 2.5   码代码

  main.py文件不改变，handle.py 需要增加一下代码，增加新的文件receive.py, reply.py

#### 1）vim handle.py

# -\*- coding: utf-8 -\*-  
# filename: handle.py  
import hashlib  
import reply  
import receive  
import web  
  
class Handle(object):  
    def POST(self):  
        try:  
            webData = web.data()  
            print "Handle Post webdata is ", webData   #后台打日志  
            recMsg = receive.parse\_xml(webData)  
            if isinstance(recMsg, receive.Msg) and recMsg.MsgType == 'text':  
                toUser = recMsg.FromUserName  
                fromUser = recMsg.ToUserName  
                content = "test"  
                replyMsg = reply.TextMsg(toUser, fromUser, content)  
                return replyMsg.send()  
            else:  
                print "暂且不处理"  
                return "success"  
        except Exception, Argment:  
            return Argment

#### 2）vim receive.py

# -\*- coding: utf-8 -\*-  
# filename: receive.py  
import xml.etree.ElementTree as ET  
  
def parse\_xml(web\_data):  
    if len(web\_data) == 0:  
        return None  
    xmlData = ET.fromstring(web\_data)  
    msg\_type = xmlData.find('MsgType').text  
    if msg\_type == 'text':  
        return TextMsg(xmlData)  
    elif msg\_type == 'image':  
        return ImageMsg(xmlData)  
  
class Msg(object):  
    def \_\_init\_\_(self, xmlData):  
        self.ToUserName = xmlData.find('ToUserName').text  
        self.FromUserName = xmlData.find('FromUserName').text  
        self.CreateTime = xmlData.find('CreateTime').text  
        self.MsgType = xmlData.find('MsgType').text  
        self.MsgId = xmlData.find('MsgId').text  
  
class TextMsg(Msg):  
    def \_\_init\_\_(self, xmlData):  
        Msg.\_\_init\_\_(self, xmlData)  
        self.Content = xmlData.find('Content').text.encode("utf-8")  
  
class ImageMsg(Msg):  
    def \_\_init\_\_(self, xmlData):  
        Msg.\_\_init\_\_(self, xmlData)  
        self.PicUrl = xmlData.find('PicUrl').text  
        self.MediaId = xmlData.find('MediaId').text

#### 3）vim reply.py

# -\*- coding: utf-8 -\*-  
# filename: reply.py  
import time  
  
class Msg(object):  
    def \_\_init\_\_(self):  
        pass  
    def send(self):  
        return "success"  
  
class TextMsg(Msg):  
    def \_\_init\_\_(self, toUserName, fromUserName, content):  
        self.\_\_dict = dict()  
        self.\_\_dict['ToUserName'] = toUserName  
        self.\_\_dict['FromUserName'] = fromUserName  
        self.\_\_dict['CreateTime'] = int(time.time())  
        self.\_\_dict['Content'] = content  
  
    def send(self):  
        XmlForm = """  
        <xml>  
        <ToUserName><![CDATA[{ToUserName}]]></ToUserName>  
        <FromUserName><![CDATA[{FromUserName}]]></FromUserName>  
        <CreateTime>{CreateTime}</CreateTime>  
        <MsgType><![CDATA[text]]></MsgType>  
        <Content><![CDATA[{Content}]]></Content>  
        </xml>  
        """  
        return XmlForm.format(\*\*self.\_\_dict)  
      
class ImageMsg(Msg):  
    def \_\_init\_\_(self, toUserName, fromUserName, mediaId):  
        self.\_\_dict = dict()  
        self.\_\_dict['ToUserName'] = toUserName  
        self.\_\_dict['FromUserName'] = fromUserName  
        self.\_\_dict['CreateTime'] = int(time.time())  
        self.\_\_dict['MediaId'] = mediaId  
    def send(self):  
        XmlForm = """  
        <xml>  
        <ToUserName><![CDATA[{ToUserName}]]></ToUserName>  
        <FromUserName><![CDATA[{FromUserName}]]></FromUserName>  
        <CreateTime>{CreateTime}</CreateTime>  
        <MsgType><![CDATA[image]]></MsgType>  
        <Image>  
        <MediaId><![CDATA[{MediaId}]]></MediaId>  
        </Image>  
        </xml>  
        """  
        return XmlForm.format(\*\*self.\_\_dict)

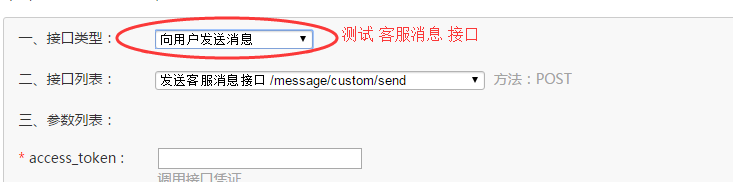
  码好代码之后，重新启动程序，sudo python main.py 80。

## 2.6   在线测试

微信公众平台有提供一个[在线测试的平台](http://mp.weixin.qq.com/debug/)，方便开发者模拟场景测试代码逻辑。正如 2.2被动回复文本消息 交代此被动回复接口不同于客服接口，测试时也要注意区别。

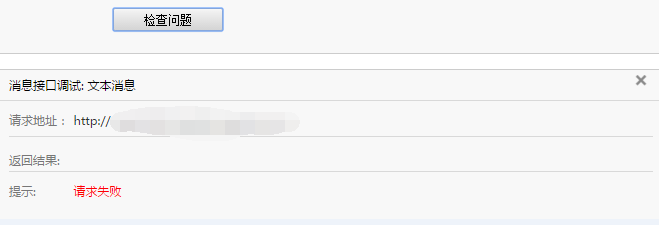
在线测试目的在于测试开发者代码逻辑是否有误、是否符合预期。即便测试成功也不会发送内容给粉丝。所以可以随意测试。





测试结果：

1）”请求失败”，说明代码有问题，请检查代码逻辑。



2）“请求成功”，然后根据返回结果查看是否符合预期。

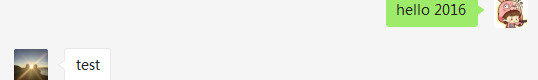


## 2.7   真实体验

拿出手机，微信扫描公众号二维码，成为自己公众号的第一个粉丝。公众号二维码位置如下图：



测试如下图：



# 3       实现“图”尚往来

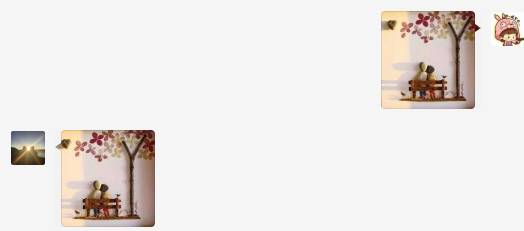
目的：

1）引入素材管理

2）以文本消息，图片消息为基础，可自行理解剩余的语音消息、视频消息、地理消息等

预实现功能：

接受粉丝发送的图片消息，并立马回复相同的图片给粉丝。



## 3.1   接收图片消息

即粉丝给公众号发送的图片消息。官方wiki链接：[消息管理/接收消息-接受普通消息/ 图片消息](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140453&token=&lang=zh_CN)。从实例讲解，粉丝给公众号发送一张图片消息，在公众号开发者后台接收到的xml如下:

 <xml>

<ToUserName><![CDATA[公众号]]></ToUserName>  
<FromUserName><![CDATA[粉丝号]]></FromUserName>  
<CreateTime>1460536575</CreateTime>  
<MsgType><![CDATA[image]]></MsgType>  
<PicUrl><![CDATA[http://mmbiz.qpic.cn/xxxxxx /0]]></PicUrl>  
<MsgId>6272956824639273066</MsgId>  
<MediaId><![CDATA[gyci5a-xxxxx-OL]]></MediaId>  
</xml>

特别说明：

PicUrl: 这个参数是微信系统把“粉丝“发送的图片消息自动转化成url。 这个url可用浏览器打开查看到图片。

MediaId: 是微信系统产生的id 用于标记该图片，详情可参考wiki [素材管理/获取临时素材](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738727&token=&lang=zh_CN)

## 3.2   被动回复图片消息

即公众号给粉丝发送的图片消息。官方wiki链接：[消息管理/发送消息-被动回复用户消息/ 图片消息](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140543&token=&lang=zh_CN)。

特别说明：

1） 被动回复消息，即发送被动响应消息，不同于客服消息接口

2） 它其实并不是一种接口，而是对微信服务器发过来消息的一次回复

3） 收到粉丝消息后不想或者不能5秒内回复时，需回复“success”字符串（下文详细介绍）

4） 客服接口在满足一定条件下随时调用

开发者发送给微信后台的xml 如下：

 <xml>  
<ToUserName><![CDATA[粉丝号]]></ToUserName>  
<FromUserName><![CDATA[公众号]]></FromUserName>  
<CreateTime>1460536576</CreateTime>  
<MsgType><![CDATA[image]]></MsgType>  
<Image>  
<MediaId><![CDATA[gyci5oxxxxxxv3cOL]]></MediaId>  
</Image>  
</xml>

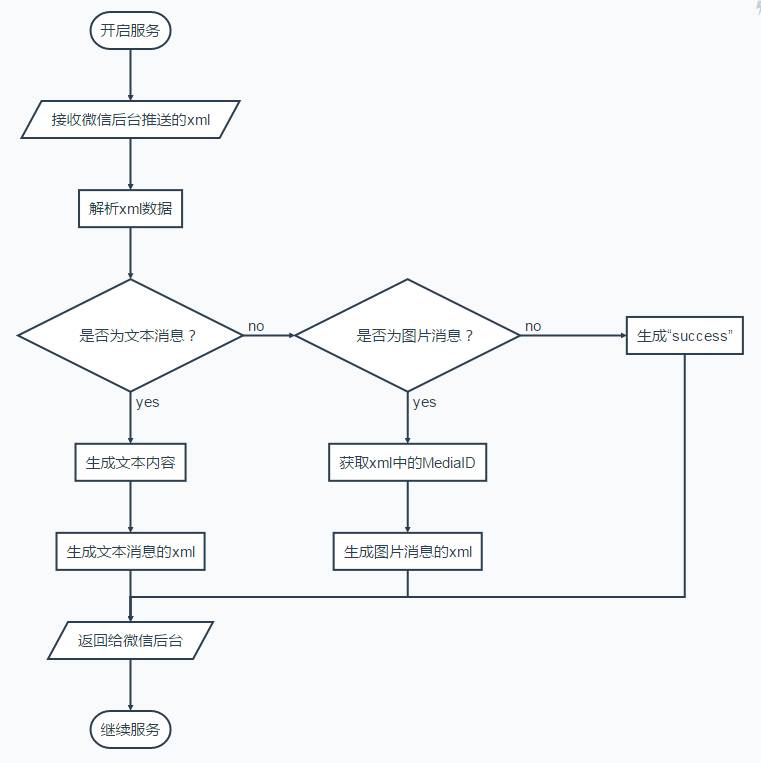
这里填写的MediaId的内容，其实就是粉丝的发送图片的原MediaId，所以粉丝收到了一张一模一样的原图。

如果想回复粉丝其它图片怎么呢？

1） 新增素材，请参考 新增临时素材 或者 新增永久素材

2） 获取其MediaId，请参考 获取临时素材MediaID 或者 获取永久素材MediaID

## 3.3   流程图



## 3.4   码代码

只显示更改的代码部分，其余部分参考上小节，在线测试，真是体验，回复空串，请参考 实现"你问我答"。vim handle.py

# -\*- coding: utf-8 -\*-

# filename: handle.py

import hashlib

import reply

import receive

import web

class Handle(object):

    def POST(self):

        try:

            webData = web.data()

            print "Handle Post webdata is ", webData   #后台打日志

            recMsg = receive.parse\_xml(webData)

            if isinstance(recMsg, receive.Msg):

                toUser = recMsg.FromUserName

                fromUser = recMsg.ToUserName

                if recMsg.MsgType == 'text':

                    content = "test"

                    replyMsg = reply.TextMsg(toUser, fromUser, content)

                    return replyMsg.send()

                if recMsg.MsgType == 'image':

                    mediaId = recMsg.MediaId

                    replyMsg = reply.ImageMsg(toUser, fromUser, mediaId)

                    return replyMsg.send()

                else:

                    return reply.Msg().send()

            else:

                print "暂且不处理"

                return reply.Msg().send()

        except Exception, Argment:

            return Argment

# 4 AccessToken

AccessToken 的意义请参考[公众平台wiki](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140183&token=&lang=zh_CN)介绍。

## 4.1 查看appid及appsecret

公众平台官网查看， 其中AppSecret 不点击重置时候，则一直保持不变。



## 4.2   获取accessToken

#### 4.2.1临时方法获取

    为了方便先体验其他接口，可以临时通过 [在线测试](http://mp.weixin.qq.com/debug/)或者 浏览器获取accessToken。

http://mmbiz.qpic.cn/mmbiz_png/PiajxSqBRaEIQxibpLbyuSK9XkjDgZoL0xBHeQU4pOZnne3ZE7xA9sYvUkxWJOOxibxQudayWeWQ604jgKhI1vK1g/0?wx_fmt=png

#### 4.2.2 接口获取

详情请见[公众平台wiki](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140183&token=&lang=zh_CN)

特别强调：

1） 第三方需要一个access\_token获取和刷新的中控服务器。

2） 并发获取access\_token会导致AccessToken互相覆盖，影响具体的业务功能

## 4.3   码代码

再次重复说明，下面代码只是为了简单说明接口获取方式。实际中并不推荐，尤其是业务繁重的公众号，更需要中控服务器，统一的获取accessToken。vim basic.py

# -\*- coding: utf-8 -\*-  
# filename: basic.py  
import urllib  
import time  
import json  
  
class Basic:  
    def \_\_init\_\_(self):  
        self.\_\_accessToken = ''  
        self.\_\_leftTime = 0  
    def \_\_real\_get\_access\_token(self):  
        appId = "xxxxx"  
        appSecret = "xxxxx"  
  
        postUrl = ("https://api.weixin.qq.com/cgi-bin/token?grant\_type="  
               "client\_credential&appid=%s&secret=%s" % (appId, appSecret))  
        urlResp = urllib.urlopen(postUrl)  
        urlResp = json.loads(urlResp.read())  
          
        self.\_\_accessToken = urlResp['access\_token']  
        self.\_\_leftTime = urlResp['expires\_in']  
  
    def get\_access\_token(self):  
        if self.\_\_leftTime < 10:  
            self.\_\_real\_get\_access\_token()  
        return self.\_\_accessToken  
  
    def run(self):  
        while(True):  
            if self.\_\_leftTime > 10:  
                time.sleep(2)  
                self.\_\_leftTime -= 2  
            else:  
                self.\_\_real\_get\_access\_token()

# 5       临时素材

公众号经常有需要用到一些临时性的多媒体素材的场景，例如在使用接口特别是发送消息时，对多媒体文件、多媒体消息的获取和调用等操作，是通过MediaID来进行的。譬如实现“图”尚往来中，粉丝给公众号发送图片消息，便产生一临时素材。

       因为永久素材有数量的限制，但是公众号又需要临时性使用一些素材，因而产生了临时素材。这类素材不在微信公众平台后台长期存储，所以在公众平台官网的素材管理中查询不到，但是可以通过接口对其操作。

       其他详情请以公众平台官网wiki介绍为依据。

## 5.1   新建临时素材

接口详情请依据wiki介绍。提供参考代码如何上传素材作为临时素材，供其它接口使用。

vim media.py  编写完成之后，直接运行media.py 即可上传临时素材。

# -\*- coding: utf-8 -\*-

# filename: media.py

from basic import Basic

import urllib2

import poster.encode

from poster.streaminghttp import register\_openers

class Media(object):

    def \_\_init\_\_(self):

        register\_openers()

    #上传图片

    def uplaod(self, accessToken, filePath, mediaType):

        openFile = open(filePath, "rb")

        param = {'media': openFile}

        postData, postHeaders = poster.encode.multipart\_encode(param)

        postUrl = "https://api.weixin.qq.com/cgi-bin/media/upload?access\_token=%s&type=%s" % (accessToken, mediaType)

        request = urllib2.Request(postUrl, postData, postHeaders)

        urlResp = urllib2.urlopen(request)

        print urlResp.read()

if \_\_name\_\_ == '\_\_main\_\_':

    myMedia = Media()

    accessToken = Basic().get\_access\_token()

    filePath = "D:/code/mpGuide/media/test.jpg"   #请安实际填写

    mediaType = "image"

    myMedia.uplaod(accessToken, filePath, mediaType)

## 5.2   获取临时素材MediaID

临时素材的MediaID 没有提供特定的接口进行统一查询，因此有俩种方式

1） 通过接口上次的临时素材，在调用成功的情况下，从返回JSON数据中提取MediaID，可临时使用

2） 粉丝互动中的临时素材，可从xml 数据提取MediaID，可临时使用

## 5.3   下载临时素材

#### 5.3.1 手工体验

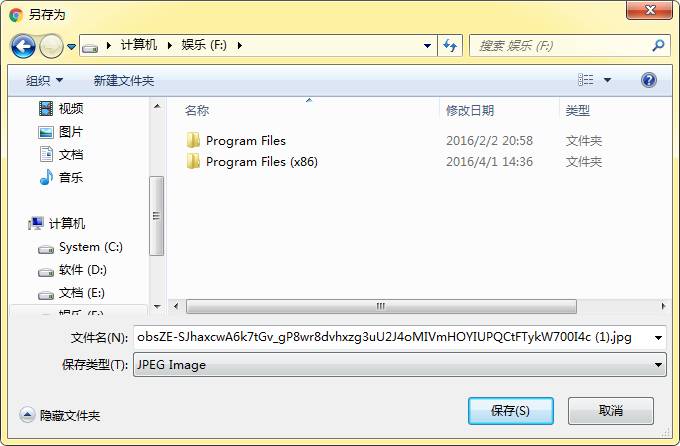
开发者如何保存粉丝发送的图片呢？接口文档：[获取临时素材](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738727&token=&lang=zh_CN)接口，为方便理解，从最简单浏览器获取素材的方法入手，根据实际情况，浏览器输入网址：

<https://api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOKEN&media_id=MEDIA_ID> （自行替换数据）

ACCESS\_TOKEN 如 "AccessToken"章节讲解

MEDIA\_ID 如 图尚往来/接受图片消息xml中的MediaId 讲解

只要数据正确，则会下载图片到本地，如下图：





#### 5.3.2接口获取

现在已经理解这个接口的功能了，只剩码代码了。vim media.py

# -\*- coding: utf-8 -\*-

# filename: media.py

import urllib2

import json

from basic import Basic

class Media(object):

    def get(self, accessToken, mediaId):

        postUrl = "https://api.weixin.qq.com/cgi-bin/media/get?access\_token=%s&media\_id=%s" % (accessToken, mediaId)

        urlResp = urllib2.urlopen(postUrl)

        headers = urlResp.info().\_\_dict\_\_['headers']

        if ('Content-Type: application/json\r\n' in headers) or ('Content-Type: text/plain\r\n' in headers):

            jsonDict = json.loads(urlResp.read())

            print jsonDict

        else:

            buffer = urlResp.read()   #素材的二进制

            mediaFile = file("test\_media.jpg", "wb")

            mediaFile.write(buffer)

            print "get successful"

if \_\_name\_\_ == '\_\_main\_\_':

    myMedia = Media()

    accessToken = Basic().get\_access\_token()

    mediaId = "2ZsPnDj9XIQlGfws31MUfR5Iuz-rcn7F6LkX3NRCsw7nDpg2268e-dbGB67WWM-N"

    myMedia.get(accessToken, mediaId)

直接运行 media.py 即可把想要的素材下载下来，其中图文消息类型的，会直接在屏幕输出json数据段。

# 6       永久素材

## 6.1   新建永久素材的方式

#### 6.1.1 手工体验

       公众号官网的素材管理新增素材。补充一点，公众平台只以MediaID区分素材，MediaID不等于素材的文件名。MediaID只能通过接口查询，公众平台官网看到的是素材的文件名，如下图：



#### 6.1.2接口删除

[新增永久素材接口（详情见wiki）](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738729&token=&lang=zh_CN)跟新增临时素材的操作差不多，使用url不一样而已，这里避免重复，以新增永久图文素材接口为例，新增其他类型的素材请参考新增临时素材代码。vim material.py

# -\*- coding: utf-8 -\*-

# filename: material.py

import urllib2

import json

from basic import Basic

class Material(object):

    #上传图文

    def add\_news(self, accessToken, news):

        postUrl = "https://api.weixin.qq.com/cgi-bin/material/add\_news?access\_token=%s" % accessToken

        urlResp = urllib2.urlopen(postUrl, news)

        print urlResp.read()

if \_\_name\_\_ == '\_\_main\_\_':

    myMaterial = Material()

    accessToken = Basic().get\_access\_token()

    news =(

    {

        "articles":

        [

            {

            "title": "test",

            "thumb\_media\_id": "X2UMe5WdDJSS2AS6BQkhTw9raS0pBdpv8wMZ9NnEzns",

            "author": "vickey",

            "digest": "",

            "show\_cover\_pic": 1,

            "content": "<p><img data-s=\"300,640\" data-type=\"jpeg\" data-src=\"http://mmbiz.qpic.cn/mmbiz/iaK7BytM0QFPLhxfSMhOHlZd2Q5cw3YibKVf4dgNpLHXdUkvl65NBSMU71rFfOEKF3ucmXuwAQbNdiaaS3441d5rg/0?wx\_fmt=jpeg\" data-ratio=\"0.748653500897666\" data-w=\"\"  /><br  /><img data-s=\"300,640\" data-type=\"jpeg\" data-src=\"http://mmbiz.qpic.cn/mmbiz/iaK7BytM0QFPLhxfSMhOHlZd2Q5cw3YibKiaibdNgh0ibgOXAuz9phrGjYFBUKlyTBcrv5WE5zic08FUcz5ODXCHEykQ/0?wx\_fmt=jpeg\" data-ratio=\"0.748653500897666\" data-w=\"\"  /><br  /></p>",

            "content\_source\_url": "",

            }

        ]

    })

    #news 是个dict类型，可通过下面方式修改内容

    #news['articles'][0]['title'] = u"测试".encode('utf-8')

    #print news['articles'][0]['title']

    news = json.dumps(news, ensure\_ascii=False)

    myMaterial.add\_news(accessToken, news)

## 6.2   获取永久素材MediaID

1） 通过[新增永久素材接口（详情见wiki）](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738729&token=&lang=zh_CN)新增素材时，保存MediaID

2） 通过获取永久素材列表(下文介绍) 的方式获取素材信息，从而得到MediaID

## 6.3   获取素材列表

官方wiki链接：[获取素材列表](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738734&token=&lang=zh_CN)。特别说明：此接口只是批量拉取素材信息，不是一次性拉去所有素材的信息，所以可以理解offset字段的含义了吧。vim  material.py

# -\*- coding: utf-8 -\*-

# filename: material.py

import urllib2

import json

import poster.encode

from poster.streaminghttp import register\_openers

from basic import Basic

class Material(object):

    def \_\_init\_\_(self):

        register\_openers()

    #上传

    def uplaod(self, accessToken, filePath, mediaType):

        openFile = open(filePath, "rb")

        fileName = "hello"

        param = {'media': openFile, 'filename': fileName}

        #param = {'media': openFile}

        postData, postHeaders = poster.encode.multipart\_encode(param)

        postUrl = "https://api.weixin.qq.com/cgi-bin/material/add\_material?access\_token=%s&type=%s" % (accessToken, mediaType)

        request = urllib2.Request(postUrl, postData, postHeaders)

        urlResp = urllib2.urlopen(request)

        print urlResp.read()

    #下载

    def get(self, accessToken, mediaId):

        postUrl = "https://api.weixin.qq.com/cgi-bin/material/get\_material?access\_token=%s" % accessToken

        postData = "{ \"media\_id\": \"%s\" }" % mediaId

        urlResp = urllib2.urlopen(postUrl, postData)

        headers = urlResp.info().\_\_dict\_\_['headers']

        if ('Content-Type: application/json\r\n' in headers) or ('Content-Type: text/plain\r\n' in headers):

            jsonDict = json.loads(urlResp.read())

            print jsonDict

        else:

            buffer = urlResp.read()  # 素材的二进制

            mediaFile = file("test\_media.jpg", "wb")

            mediaFile.write(buffer)

            print "get successful"

    #删除

    def delete(self, accessToken, mediaId):

        postUrl = "https://api.weixin.qq.com/cgi-bin/material/del\_material?access\_token=%s" % accessToken

        postData = "{ \"media\_id\": \"%s\" }" % mediaId

        urlResp = urllib2.urlopen(postUrl, postData)

        print urlResp.read()

    #获取素材列表

    def batch\_get(self, accessToken, mediaType, offset=0, count=20):

        postUrl = ("https://api.weixin.qq.com/cgi-bin/material"

               "/batchget\_material?access\_token=%s" % accessToken)

        postData = ("{ \"type\": \"%s\", \"offset\": %d, \"count\": %d }"

                    % (mediaType, offset, count))

        urlResp = urllib2.urlopen(postUrl, postData)

        print urlResp.read()

if \_\_name\_\_ == '\_\_main\_\_':

    myMaterial = Material()

    accessToken = Basic().get\_access\_token()

    mediaType = "news"

    myMaterial.batch\_get(accessToken, mediaType)

## 6.4   删除永久素材

如果我想删除掉 20160102.jpg 这张图片，除了官网直接操作，也可以使用接口： [删除永久素材](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1444738731&token=&lang=zh_CN)接口文档。

首先需要知道该图片的mediaID，方法上小节已讲述。代码可参考上小节：Material().delete() 接口

调用接口成功后，在公众平台官网素材管理的图片中，查询不到已删除的图片。



# 7       自定义菜单

自定义菜单意义作用请参考 [官方wiki](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421141013&token=&lang=zh_CN)介绍。

目标：三个菜单栏，体验click、view、media\_id 三种类型的菜单按钮，其他类型在本小节学习之后，自行请查询公众平台wiki说明领悟。

## 7.1   创建菜单界面

1）根据公众平台wiki 给的json 数据编写代码，其中涉及media\_id部分请阅读"永久素材"章节。vim menu.py

# -\*- coding: utf-8 -\*-

# filename: menu.py

import urllib

from basic import Basic

class Menu(object):

    def \_\_init\_\_(self):

        pass

    def create(self, postData, accessToken):

        postUrl = "https://api.weixin.qq.com/cgi-bin/menu/create?access\_token=%s" % accessToken

        if isinstance(postData, unicode):

            postData = postData.encode('utf-8')

        urlResp = urllib.urlopen(url=postUrl, data=postData)

        print urlResp.read()

    def query(self, accessToken):

        postUrl = "https://api.weixin.qq.com/cgi-bin/menu/get?access\_token=%s" % accessToken

        urlResp = urllib.urlopen(url=postUrl)

        print urlResp.read()

    def delete(self, accessToken):

        postUrl = "https://api.weixin.qq.com/cgi-bin/menu/delete?access\_token=%s" % accessToken

        urlResp = urllib.urlopen(url=postUrl)

        print urlResp.read()

    #获取自定义菜单配置接口

    def get\_current\_selfmenu\_info(self, accessToken):

        postUrl = "https://api.weixin.qq.com/cgi-bin/get\_current\_selfmenu\_info?access\_token=%s" % accessToken

        urlResp = urllib.urlopen(url=postUrl)

        print urlResp.read()

if \_\_name\_\_ == '\_\_main\_\_':

    myMenu = Menu()

    postJson = """

    {

        "button":

        [

            {

                "type": "click",

                "name": "开发指引",

                "key":  "mpGuide"

            },

            {

                "name": "公众平台",

                "sub\_button":

                [

                    {

                        "type": "view",

                        "name": "更新公告",

                        "url": "http://mp.weixin.qq.com/wiki?t=resource/res\_main&id=mp1418702138&token=&lang=zh\_CN"

                    },

                    {

                        "type": "view",

                        "name": "接口权限说明",

                        "url": "http://mp.weixin.qq.com/wiki?t=resource/res\_main&id=mp1418702138&token=&lang=zh\_CN"

                    },

                    {

                        "type": "view",

                        "name": "返回码说明",

                        "url": "http://mp.weixin.qq.com/wiki?t=resource/res\_main&id=mp1433747234&token=&lang=zh\_CN"

                    }

                ]

            },

            {

                "type": "media\_id",

                "name": "旅行",

                "media\_id": "z2zOokJvlzCXXNhSjF46gdx6rSghwX2xOD5GUV9nbX4"

            }

          ]

    }

    """

    accessToken = Basic().get\_access\_token()

    #myMenu.delete(accessToken)

    myMenu.create(postJson, accessToken)

2）在腾讯云服务器上执行命令：python menu.py。

3）查看：

重新关注公众号后即可看到新创建菜单界面，题外话，如果不重新关注，公众号界面也会自动更改，但又时间延迟。

如下图所示，点击子菜单“更新公告“（view类型），弹出网页（pc版本）



             点击旅行（media\_id类型），公众号显示了一篇图文消息，如下图所示：



              点击开发指引（click类型），发现公众号系统提示：“该公众号暂时无法提供服务“。

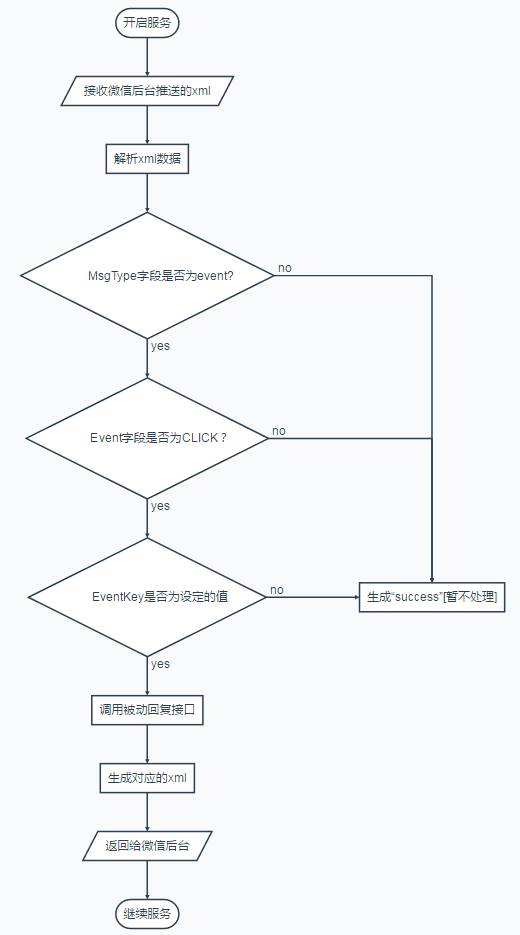


## 7.2   完善菜单功能

查看[公众平台自定义菜单wiki](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421141013&token=&lang=zh_CN) 与[公众平台wiki/自定义菜单事件推送](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421141016&token=&lang=zh_CN)后，可知：点击click类型button，微信后台会推送一个event类型的xml 给开发者。

显然，click类型的还需要开发者进一步完善后台代码逻辑，增加对自定义菜单事件推送的相应。

#### 7.2.1 流程图



#### 7.2.2码代码

 1) vim handle.py （修改）

# -\*- coding: utf-8 -\*-

# filename: handle.py

import reply

import receive

import web

class Handle(object):

    def POST(self):

        try:

            webData = web.data()

            print "Handle Post webdata is ", webData   #后台打日志

            recMsg = receive.parse\_xml(webData)

            if isinstance(recMsg, receive.Msg):

                toUser = recMsg.FromUserName

                fromUser = recMsg.ToUserName

                if recMsg.MsgType == 'text':

                    content = "test"

                    replyMsg = reply.TextMsg(toUser, fromUser, content)

                    return replyMsg.send()

                if recMsg.MsgType == 'image':

                    mediaId = recMsg.MediaId

                    replyMsg = reply.ImageMsg(toUser, fromUser, mediaId)

                    return replyMsg.send()

            if isinstance(recMsg, receive.EventMsg):

                if recMsg.Event == 'CLICK':

                    if recMsg.Eventkey == 'mpGuide':

                        content = u"编写中，尚未完成".encode('utf-8')

                        replyMsg = reply.TextMsg(toUser, fromUser, content)

                        return replyMsg.send()

            print "暂且不处理"

            return reply.Msg().send()

        except Exception, Argment:

            return Argment

  2）vim receive.py (修改)

# -\*- coding: utf-8 -\*-

# filename: receive.py

import xml.etree.ElementTree as ET

def parse\_xml(web\_data):

    if len(web\_data) == 0:

        return None

    xmlData = ET.fromstring(web\_data)

    msg\_type = xmlData.find('MsgType').text

    if msg\_type == 'event':

        event\_type = xmlData.find('Event').text

        if event\_type == 'CLICK':

            return Click(xmlData)

        #elif event\_type in ('subscribe', 'unsubscribe'):

            #return Subscribe(xmlData)

        #elif event\_type == 'VIEW':

            #return View(xmlData)

        #elif event\_type == 'LOCATION':

            #return LocationEvent(xmlData)

        #elif event\_type == 'SCAN':

            #return Scan(xmlData)

    elif msg\_type == 'text':

        return TextMsg(xmlData)

    elif msg\_type == 'image':

        return ImageMsg(xmlData)

class EventMsg(object):

    def \_\_init\_\_(self, xmlData):

        self.ToUserName = xmlData.find('ToUserName').text

        self.FromUserName = xmlData.find('FromUserName').text

        self.CreateTime = xmlData.find('CreateTime').text

        self.MsgType = xmlData.find('MsgType').text

        self.Event = xmlData.find('Event').text

class Click(EventMsg):

    def \_\_init\_\_(self, xmlData):

        EventMsg.\_\_init\_\_(self, xmlData)

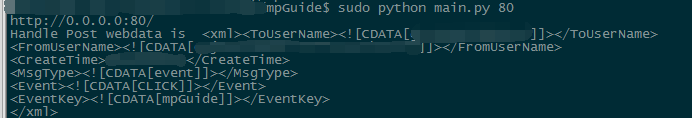
        self.Eventkey = xmlData.find('EventKey').text

## 7.3   体验

  编译好代码后，重新启动服务，（sudo python main.py 80）,view类型、medie\_id类型的本身就很容易实现，现在重点看一下click类型的菜单按钮。

  微信扫码成为公众号的粉丝，点击菜单按钮“开发指引”。

  查看后台日志，发现接收到一条xml，如截图：



  公众号的后台代码设置对该事件的处理是回复一条内容为“编写之中”的文本消息，因此公众号发送了一条文本消息给我，如图：



好啦，到此，目标已实现。对于自定义菜单其他类型，均同理可操作。

# 8       关于反馈问题

是程序肯定有bug，所以在使用开放平台过程中，肯定会遇见各种各样的问题，可能是自己的坑，也可能是微信团队的锅。当自己查自己代码千千遍，依旧没有发现问题时候，可以通过腾讯客服等等渠道，请求微信团队的帮助。如何高效快速的得到帮助呢？下面强调三个要点：

1）简明扼要的描述清楚场景以及遇见问题，描述过程中尽可能使用wiki上的名称，譬如自定义菜单，素材管理等专有名词，不然开发根本不知道你在说什么。

2）提供账号信息：AppID（登陆公众平台官网->基本配置），若牵扯粉丝提供粉丝的OpenID。

3）提供bug的发生时间，至少要以小时为单位（年-月-日-小时），当然越具体越容易查明问题。

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：



请注意：

1、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。

2、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。

3、创建自定义菜单后，菜单的刷新策略是，在用户进入公众号会话页或公众号profile页时，如果发现上一次拉取菜单的请求在5分钟以前，就会拉取一下菜单，如果菜单有更新，就会刷新客户端的菜单。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。

自定义菜单接口可实现多种类型按钮，如下：

1、click：点击推事件用户点击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；

2、view：跳转URL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网页URL，可与网页授权获取用户基本信息接口结合，获得用户基本信息。

3、scancode\_push：扫码推事件用户点击按钮后，微信客户端将调起扫一扫工具，完成扫码操作后显示扫描结果（如果是URL，将进入URL），且会将扫码的结果传给开发者，开发者可以下发消息。

4、scancode\_waitmsg：扫码推事件且弹出“消息接收中”提示框用户点击按钮后，微信客户端将调起扫一扫工具，完成扫码操作后，将扫码的结果传给开发者，同时收起扫一扫工具，然后弹出“消息接收中”提示框，随后可能会收到开发者下发的消息。

5、pic\_sysphoto：弹出系统拍照发图用户点击按钮后，微信客户端将调起系统相机，完成拍照操作后，会将拍摄的相片发送给开发者，并推送事件给开发者，同时收起系统相机，随后可能会收到开发者下发的消息。

6、pic\_photo\_or\_album：弹出拍照或者相册发图用户点击按钮后，微信客户端将弹出选择器供用户选择“拍照”或者“从手机相册选择”。用户选择后即走其他两种流程。

7、pic\_weixin：弹出微信相册发图器用户点击按钮后，微信客户端将调起微信相册，完成选择操作后，将选择的相片发送给开发者的服务器，并推送事件给开发者，同时收起相册，随后可能会收到开发者下发的消息。

8、location\_select：弹出地理位置选择器用户点击按钮后，微信客户端将调起地理位置选择工具，完成选择操作后，将选择的地理位置发送给开发者的服务器，同时收起位置选择工具，随后可能会收到开发者下发的消息。

9、media\_id：下发消息（除文本消息）用户点击media\_id类型按钮后，微信服务器会将开发者填写的永久素材id对应的素材下发给用户，永久素材类型可以是图片、音频、视频、图文消息。请注意：永久素材id必须是在“素材管理/新增永久素材”接口上传后获得的合法id。

10、view\_limited：跳转图文消息URL用户点击view\_limited类型按钮后，微信客户端将打开开发者在按钮中填写的永久素材id对应的图文消息URL，永久素材类型只支持图文消息。请注意：永久素材id必须是在“素材管理/新增永久素材”接口上传后获得的合法id。

**请注意，3到8的所有事件，仅支持微信iPhone5.4.1以上版本，和Android5.4以上版本的微信用户，旧版本微信用户点击后将没有回应，开发者也不能正常接收到事件推送。9和10，是专门给第三方平台旗下未微信认证（具体而言，是资质认证未通过）的订阅号准备的事件类型，它们是没有事件推送的，能力相对受限，其他类型的公众号不必使用。**

**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>

**click和view的请求示例**

 {

     "button":[

     {

          "type":"click",

          "name":"今日歌曲",

          "key":"V1001\_TODAY\_MUSIC"

      },

      {

           "name":"菜单",

           "sub\_button":[

           {

               "type":"view",

               "name":"搜索",

               "url":"http://www.soso.com/"

            },

            {

                 "type":"miniprogram",

                 "name":"wxa",

                 "url":"http://mp.weixin.qq.com",

                 "appid":"wx286b93c14bbf93aa",

                 "pagepath":"pages/lunar/index"

             },

            {

               "type":"click",

               "name":"赞一下我们",

               "key":"V1001\_GOOD"

            }]

       }]

 }

**其他新增按钮类型的请求示例**

{

    "button": [

        {

            "name": "扫码",

            "sub\_button": [

                {

                    "type": "scancode\_waitmsg",

                    "name": "扫码带提示",

                    "key": "rselfmenu\_0\_0",

                    "sub\_button": [ ]

                },

                {

                    "type": "scancode\_push",

                    "name": "扫码推事件",

                    "key": "rselfmenu\_0\_1",

                    "sub\_button": [ ]

                }

            ]

        },

        {

            "name": "发图",

            "sub\_button": [

                {

                    "type": "pic\_sysphoto",

                    "name": "系统拍照发图",

                    "key": "rselfmenu\_1\_0",

                   "sub\_button": [ ]

                 },

                {

                    "type": "pic\_photo\_or\_album",

                    "name": "拍照或者相册发图",

                    "key": "rselfmenu\_1\_1",

                    "sub\_button": [ ]

                },

                {

                    "type": "pic\_weixin",

                    "name": "微信相册发图",

                    "key": "rselfmenu\_1\_2",

                    "sub\_button": [ ]

                }

            ]

        },

        {

            "name": "发送位置",

            "type": "location\_select",

            "key": "rselfmenu\_2\_0"

        },

        {

           "type": "media\_id",

           "name": "图片",

           "media\_id": "MEDIA\_ID1"

        },

        {

           "type": "view\_limited",

           "name": "图文消息",

           "media\_id": "MEDIA\_ID2"

        }

    ]

}

**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，view表示网页类型，click表示点击类型，miniprogram表示小程序类型 |
| name | 是 | 菜单标题，不超过16个字节，子菜单不超过60个字节 |
| key | click等点击类型必须 | 菜单KEY值，用于消息接口推送，不超过128字节 |
| url | view、miniprogram类型必须 | 网页链接，用户点击菜单可打开链接，不超过1024字节。type为miniprogram时，不支持小程序的老版本客户端将打开本url。 |
| media\_id | media\_id类型和view\_limited类型必须 | 调用新增永久素材接口返回的合法media\_id |
| appid | miniprogram类型必须 | 小程序的appid（仅认证公众号可配置） |
| pagepath | miniprogram类型必须 | 小程序的页面路径 |

**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"}

[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=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95&form=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95%E5%88%9B%E5%BB%BA%E6%8E%A5%E5%8F%A3%20/menu/create)

## 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。另外请注意，在设置了个性化菜单后，使用本自定义菜单查询接口可以获取默认菜单和全部个性化菜单信息。

**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（无个性化菜单时）**

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": [ ]

                    }

                ]

            }

        ]

    }

}

**返回说明（有个性化菜单时）**

{

    "menu": {

        "button": [

            {

                "type": "click",

                "name": "今日歌曲",

                "key": "V1001\_TODAY\_MUSIC",

                "sub\_button": [ ]

            }

        ],

        "menuid": 208396938

    },

    "conditionalmenu": [

        {

            "button": [

                {

                    "type": "click",

                    "name": "今日歌曲",

                    "key": "V1001\_TODAY\_MUSIC",

                    "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": [ ]

                        }

                    ]

                }

            ],

            "matchrule": {

                "group\_id": 2,

                "sex": 1,

                "country": "中国",

                "province": "广东",

                "city": "广州",

                "client\_platform\_type": 2

            },

            "menuid": 208396993

        }

    ]

}

注：menu为默认菜单，conditionalmenu为个性化菜单列表。字段说明请见个性化菜单接口页的说明。

[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=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95&form=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95%E6%9F%A5%E8%AF%A2%E6%8E%A5%E5%8F%A3%20/menu/get)

## 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。另请注意，在个性化菜单时，调用此接口会删除默认菜单及全部个性化菜单。

**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"}

[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=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95&form=%E8%87%AA%E5%AE%9A%E4%B9%89%E8%8F%9C%E5%8D%95%E5%88%A0%E9%99%A4%E6%8E%A5%E5%8F%A3%20/menu/delete)

自定义菜单事件推送

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。**请注意，第3个到第8个的所有事件，仅支持微信iPhone5.4.1以上版本，和Android5.4以上版本的微信用户，旧版本微信用户点击后将没有回应，开发者也不能正常接收到事件推送。**

目录

[1 点击菜单拉取消息时的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#1)

[2 点击菜单跳转链接时的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#2)

[3 scancode\_push：扫码推事件的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#3)

[4 scancode\_waitmsg：扫码推事件且弹出“消息接收中”提示框的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#4)

[5 pic\_sysphoto：弹出系统拍照发图的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#5)

[6 pic\_photo\_or\_album：弹出拍照或者相册发图的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#6)

[7 pic\_weixin：弹出微信相册发图器的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#7)

[8 location\_select：弹出地理位置选择器的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141016&t=0.6628182784578314#8)

推送XML数据包示例：点击菜单拉取消息时的事件推送

<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>

<MenuId>MENUID</MenuId>

</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 |
| MenuID | 指菜单ID，如果是个性化菜单，则可以通过这个字段，知道是哪个规则的菜单被点击了。 |

scancode\_push：扫码推事件的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408090502</CreateTime>

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

<Event><![CDATA[scancode\_push]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<ScanCodeInfo><ScanType><![CDATA[qrcode]]></ScanType>

<ScanResult><![CDATA[1]]></ScanResult>

</ScanCodeInfo>

</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，scancode\_push |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| ScanCodeInfo | 扫描信息 |
| ScanType | 扫描类型，一般是qrcode |
| ScanResult | 扫描结果，即二维码对应的字符串信息 |

scancode\_waitmsg：扫码推事件且弹出“消息接收中”提示框的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408090606</CreateTime>

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

<Event><![CDATA[scancode\_waitmsg]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<ScanCodeInfo><ScanType><![CDATA[qrcode]]></ScanType>

<ScanResult><![CDATA[2]]></ScanResult>

</ScanCodeInfo>

</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，scancode\_waitmsg |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| ScanCodeInfo | 扫描信息 |
| ScanType | 扫描类型，一般是qrcode |
| ScanResult | 扫描结果，即二维码对应的字符串信息 |

pic\_sysphoto：弹出系统拍照发图的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408090651</CreateTime>

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

<Event><![CDATA[pic\_sysphoto]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<SendPicsInfo><Count>1</Count>

<PicList><item><PicMd5Sum><![CDATA[1b5f7c23b5bf75682a53e7b6d163e185]]></PicMd5Sum>

</item>

</PicList>

</SendPicsInfo>

</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，pic\_sysphoto |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| SendPicsInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |

pic\_photo\_or\_album：弹出拍照或者相册发图的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408090816</CreateTime>

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

<Event><![CDATA[pic\_photo\_or\_album]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<SendPicsInfo><Count>1</Count>

<PicList><item><PicMd5Sum><![CDATA[5a75aaca956d97be686719218f275c6b]]></PicMd5Sum>

</item>

</PicList>

</SendPicsInfo>

</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，pic\_photo\_or\_album |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| SendPicsInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |

pic\_weixin：弹出微信相册发图器的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408090816</CreateTime>

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

<Event><![CDATA[pic\_weixin]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<SendPicsInfo><Count>1</Count>

<PicList><item><PicMd5Sum><![CDATA[5a75aaca956d97be686719218f275c6b]]></PicMd5Sum>

</item>

</PicList>

</SendPicsInfo>

</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，pic\_weixin |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| SendPicsInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |

location\_select：弹出地理位置选择器的事件推送

推送XML数据包示例：

<xml><ToUserName><![CDATA[gh\_e136c6e50636]]></ToUserName>

<FromUserName><![CDATA[oMgHVjngRipVsoxg6TuX3vz6glDg]]></FromUserName>

<CreateTime>1408091189</CreateTime>

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

<Event><![CDATA[location\_select]]></Event>

<EventKey><![CDATA[6]]></EventKey>

<SendLocationInfo><Location\_X><![CDATA[23]]></Location\_X>

<Location\_Y><![CDATA[113]]></Location\_Y>

<Scale><![CDATA[15]]></Scale>

<Label><![CDATA[ 广州市海珠区客村艺苑路 106号]]></Label>

<Poiname><![CDATA[]]></Poiname>

</SendLocationInfo>

</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\_select |
| EventKey | 事件KEY值，由开发者在创建菜单时设定 |
| SendLocationInfo | 发送的位置信息 |
| Location\_X | X坐标信息 |
| Location\_Y | Y坐标信息 |
| Scale | 精度，可理解为精度或者比例尺、越精细的话 scale越高 |
| Label | 地理位置的字符串信息 |
| Poiname | 朋友圈POI的名字，可能为空 |

## 个性化菜单接口

为了帮助公众号实现灵活的业务运营，微信公众平台新增了个性化菜单接口，开发者可以通过该接口，让公众号的不同用户群体看到不一样的自定义菜单。该接口开放给已认证订阅号和已认证服务号。

开发者可以通过以下条件来设置用户看到的菜单：

1、用户标签（开发者的业务需求可以借助用户标签来完成）

2、性别

3、手机操作系统

4、地区（用户在微信客户端设置的地区）

5、语言（用户在微信客户端设置的语言）

个性化菜单接口说明：

1、个性化菜单要求用户的微信客户端版本在iPhone6.2.2，Android 6.2.4以上，暂时不支持其他版本微信

2、菜单的刷新策略是，在用户进入公众号会话页或公众号profile页时，如果发现上一次拉取菜单的请求在5分钟以前，就会拉取一下菜单，如果菜单有更新，就会刷新客户端的菜单。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

3、普通公众号的个性化菜单的新增接口每日限制次数为2000次，删除接口也是2000次，测试个性化菜单匹配结果接口为20000次

4、出于安全考虑，一个公众号的所有个性化菜单，最多只能设置为跳转到3个域名下的链接

5、创建个性化菜单之前必须先创建默认菜单（默认菜单是指使用普通自定义菜单创建接口创建的菜单）。如果删除默认菜单，个性化菜单也会全部删除

6、个性化菜单接口支持用户标签，请开发者注意，当用户身上的标签超过1个时，以最后打上的标签为匹配

个性化菜单匹配规则说明：

个性化菜单的更新是会被覆盖的。

例如公众号先后发布了默认菜单，个性化菜单1，个性化菜单2，个性化菜单3。那么当用户进入公众号页面时，将从个性化菜单3开始匹配，如果个性化菜单3匹配成功，则直接返回个性化菜单3，否则继续尝试匹配个性化菜单2，直到成功匹配到一个菜单。

根据上述匹配规则，为了避免菜单生效时间的混淆，决定不予提供个性化菜单编辑API，开发者需要更新菜单时，需将完整配置重新发布一轮。

目录

[1 创建个性化菜单](https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.9562335496597527#1)

[2 删除个性化菜单](https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.9562335496597527#2)

[3 测试个性化菜单匹配结果](https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.9562335496597527#3)

[4 查询个性化菜单](https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.9562335496597527#4)

[5 删除所有菜单](https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.9562335496597527#5)

### ****创建个性化菜单****

http请求方式：POST（请使用https协议）

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

**请求示例**

{

  "button":[

  {

     "type":"click",

     "name":"今日歌曲",

      "key":"V1001\_TODAY\_MUSIC"

},

{

"name":"菜单",

"sub\_button":[

{

"type":"view",

"name":"搜索",

"url":"http://www.soso.com/"

},

                 {

                         "type":"miniprogram",

                         "name":"wxa",

                         "url":"http://mp.weixin.qq.com",

                         "appid":"wx286b93c14bbf93aa",

                         "pagepath":"pages/lunar/index"

                   },

          {

"type":"click",

"name":"赞一下我们",

"key":"V1001\_GOOD"

        }]

 }],

"matchrule":{

  "tag\_id":"2",

  "sex":"1",

  "country":"中国",

  "province":"广东",

  "city":"广州",

  "client\_platform\_type":"2",

  "language":"zh\_CN"

  }

}

**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，view表示网页类型，click表示点击类型，miniprogram表示小程序类型 |
| 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等点击类型必须 | 菜单KEY值，用于消息接口推送，不超过128字节 |
| url | view、miniprogram类型必须 | 网页链接，用户点击菜单可打开链接，不超过1024字节。当type为miniprogram时，不支持小程序的老版本客户端将打开本url |
| media\_id | media\_id类型和view\_limited类型必须 | 调用新增永久素材接口返回的合法media\_id |
| appid | miniprogram类型必须 | 小程序的appid |
| pagepath | miniprogram类型必须 | 小程序的页面路径 |
| matchrule | 是 | 菜单匹配规则 |
| tag\_id | 否 | 用户标签的id，可通过用户标签管理接口获取 |
| sex | 否 | 性别：男（1）女（2），不填则不做匹配 |
| client\_platform\_type | 否 | 客户端版本，当前只具体到系统型号：IOS(1), Android(2),Others(3)，不填则不做匹配 |
| country | 否 | 国家信息，是用户在微信中设置的地区，具体请参考地区信息表 |
| province | 否 | 省份信息，是用户在微信中设置的地区，具体请参考地区信息表 |
| city | 否 | 城市信息，是用户在微信中设置的地区，具体请参考地区信息表 |
| language | 否 | 语言信息，是用户在微信中设置的语言，具体请参考语言表：  1、简体中文 "zh\_CN" 2、繁体中文TW "zh\_TW" 3、繁体中文HK "zh\_HK" 4、英文 "en" 5、印尼 "id" 6、马来 "ms" 7、西班牙 "es" 8、韩国 "ko" 9、意大利 "it" 10、日本 "ja" 11、波兰 "pl" 12、葡萄牙 "pt" 13、俄国 "ru" 14、泰文 "th" 15、越南 "vi" 16、阿拉伯语 "ar" 17、北印度 "hi" 18、希伯来 "he" 19、土耳其 "tr" 20、德语 "de" 21、法语 "fr" |

matchrule共六个字段，均可为空，但不能全部为空，至少要有一个匹配信息是不为空的。 country、province、city组成地区信息，将按照country、province、city的顺序进行验证，要符合地区信息表的内容。地区信息从大到小验证，小的可以不填，即若填写了省份信息，则国家信息也必填并且匹配，城市信息可以不填。 例如 “中国 广东省 广州市”、“中国 广东省”都是合法的地域信息，而“中国 广州市”则不合法，因为填写了城市信息但没有填写省份信息。 地区信息表请[点击下载](http://mp.weixin.qq.com/wiki/static/assets/870a3c2a14e97b3e74fde5e88fa47717.zip)。

**返回结果**

正确时的返回JSON数据包如下，错误时的返回码请见接口返回码说明。

{

"menuid":"208379533"

}

### ****删除个性化菜单****

http请求方式：POST（请使用https协议）

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

**请求示例**

{

"menuid":"208379533"

}

menuid为菜单id，可以通过自定义菜单查询接口获取。

正确时的返回JSON数据包如下，错误时的返回码请见接口返回码说明。：

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

### ****测试个性化菜单匹配结果****

http请求方式：POST（请使用https协议）

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

**请求示例**

{

"user\_id":"weixin"

}

user\_id可以是粉丝的OpenID，也可以是粉丝的微信号。

**返回结果** 该接口将返回菜单配置，示例如下：

{

    "button": [

        {

            "type": "view",

            "name": "tx",

            "url": "http://www.qq.com/",

            "sub\_button": [ ]

        },

        {

            "type": "view",

            "name": "tx",

            "url": "http://www.qq.com/",

            "sub\_button": [ ]

        },

        {

            "type": "view",

            "name": "tx",

            "url": "http://www.qq.com/",

            "sub\_button": [ ]

        }

    ]

}

错误时的返回码请见接口返回码说明。

### ****查询个性化菜单****

使用普通自定义菜单查询接口可以获取默认菜单和全部个性化菜单信息，请见自定义菜单查询接口的说明。

### ****删除所有菜单****

使用普通自定义菜单删除接口可以删除所有自定义菜单（包括默认菜单和全部个性化菜单），请见自定义菜单删除接口的说明。

获取自定义菜单配置接口

本接口将会提供公众号当前使用的自定义菜单的配置，如果公众号是通过API调用设置的菜单，则返回菜单的开发配置，而如果公众号是在公众平台官网通过网站功能发布菜单，则本接口返回运营者设置的菜单配置。

请注意：

1、第三方平台开发者可以通过本接口，在旗下公众号将业务授权给你后，立即通过本接口检测公众号的自定义菜单配置，并通过接口再次给公众号设置好自动回复规则，以提升公众号运营者的业务体验。

2、本接口与自定义菜单查询接口的不同之处在于，本接口无论公众号的接口是如何设置的，都能查询到接口，而自定义菜单查询接口则仅能查询到使用API设置的菜单配置。

3、认证/未认证的服务号/订阅号，以及接口测试号，均拥有该接口权限。

4、从第三方平台的公众号登录授权机制上来说，该接口从属于消息与菜单权限集。

5、本接口中返回的图片/语音/视频为临时素材（临时素材每次获取都不同，3天内有效，通过素材管理-获取临时素材接口来获取这些素材），本接口返回的图文消息为永久素材素材（通过素材管理-获取永久素材接口来获取这些素材）。

**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/get\_current\_selfmenu\_info?access\_token=ACCESS\_TOKEN

**返回结果说明**

如果公众号是在公众平台官网通过网站功能发布菜单，则本接口返回的自定义菜单配置样例如下：

{

   "is\_menu\_open": 1,

   "selfmenu\_info": {

       "button": [

           {

               "name": "button",

               "sub\_button": {

                   "list": [

                       {

                           "type": "view",

                           "name": "view\_url",

                           "url": "http://www.qq.com"

                       },

                       {

                           "type": "news",

                           "name": "news",

                           "value":"KQb\_w\_Tiz-nSdVLoTV35Psmty8hGBulGhEdbb9SKs-o",

                           "news\_info": {

                               "list": [

                                   {

                                       "title": "MULTI\_NEWS",

                                       "author": "JIMZHENG",

                                       "digest": "text",

                                       "show\_cover": 0,

                                       "cover\_url": "http://mmbiz.qpic.cn/mmbiz/GE7et87vE9vicuCibqXsX9GPPLuEtBfXfK0HKuBIa1A1cypS0uY1wickv70iaY1gf3I1DTszuJoS3lAVLvhTcm9sDA/0",

                                       "content\_url": "http://mp.weixin.qq.com/s?\_\_biz=MjM5ODUwNTM3Ng==&mid=204013432&idx=1&sn=80ce6d9abcb832237bf86c87e50fda15#rd",

                                       "source\_url": ""

                                   },

                                   {

                                       "title": "MULTI\_NEWS1",

                                       "author": "JIMZHENG",

                                       "digest": "MULTI\_NEWS1",

                                       "show\_cover": 1,

                                       "cover\_url": "http://mmbiz.qpic.cn/mmbiz/GE7et87vE9vicuCibqXsX9GPPLuEtBfXfKnmnpXYgWmQD5gXUrEApIYBCgvh2yHsu3ic3anDUGtUCHwjiaEC5bicd7A/0",

                                       "content\_url": "http://mp.weixin.qq.com/s?\_\_biz=MjM5ODUwNTM3Ng==&mid=204013432&idx=2&sn=8226843afb14ecdecb08d9ce46bc1d37#rd",

                                       "source\_url": ""

                                   }

                               ]

                           }

                       },

                       {

                           "type": "video",

                           "name": "video",

                           "value": "http://61.182.130.30/vweixinp.tc.qq.com/1007\_114bcede9a2244eeb5ab7f76d951df5f.f10.mp4?vkey=77A42D0C2015FBB0A3653D29C571B5F4BBF1D243FBEF17F09C24FF1F2F22E30881BD350E360BC53F&sha=0&save=1"

                       },

                       {

                           "type": "voice",

                           "name": "voice",

                           "value": "nTXe3aghlQ4XYHa0AQPWiQQbFW9RVtaYTLPC1PCQx11qc9UB6CiUPFjdkeEtJicn"

                       }

                   ]

               }

           },

           {

               "type": "text",

               "name": "text",

               "value": "This is text!"

           },

           {

               "type": "img",

               "name": "photo",

               "value": "ax5Whs5dsoomJLEppAvftBUuH7CgXCZGFbFJifmbUjnQk\_ierMHY99Y5d2Cv14RD"

           }

       ]

   }

}

如果公众号是通过API调用设置的菜单，自定义菜单配置样例如下：

{

   "is\_menu\_open": 1,

   "selfmenu\_info": {

       "button": [

           {

               "type": "click",

               "name": "今日歌曲",

               "key": "V1001\_TODAY\_MUSIC"

           },

           {

               "name": "菜单",

               "sub\_button": {

                   "list": [

                       {

                           "type": "view",

                           "name": "搜索",

                           "url": "http://www.soso.com/"

                       },

                       {

                           "type": "view",

                           "name": "视频",

                           "url": "http://v.qq.com/"

                       },

                       {

                           "type": "click",

                           "name": "赞一下我们",

                           "key": "V1001\_GOOD"

                       }

                   ]

               }

           }

       ]

   }

}

**参数说明**

|  |  |
| --- | --- |
| 参数 | 说明 |
| is\_menu\_open | 菜单是否开启，0代表未开启，1代表开启 |
| selfmenu\_info | 菜单信息 |
| button | 菜单按钮 |
| type | 菜单的类型，公众平台官网上能够设置的菜单类型有view（跳转网页）、text（返回文本，下同）、img、photo、video、voice。使用API设置的则有8种，详见《自定义菜单创建接口》 |
| name | 菜单名称 |
| value、url、key等字段 | 对于不同的菜单类型，value的值意义不同。官网上设置的自定义菜单：  Text:保存文字到value； Img、voice：保存mediaID到value； Video：保存视频下载链接到value； News：保存图文消息到news\_info，同时保存mediaID到value； View：保存链接到url。  使用API设置的自定义菜单： click、scancode\_push、scancode\_waitmsg、pic\_sysphoto、pic\_photo\_or\_album、 pic\_weixin、location\_select：保存值到key；view：保存链接到url |
| news\_info | 图文消息的信息 |
| title | 图文消息的标题 |
| digest | 摘要 |
| author | 作者 |
| show\_cover | 是否显示封面，0为不显示，1为显示 |
| cover\_url | 封面图片的URL |
| content\_url | 正文的URL |
| source\_url | 原文的URL，若置空则无查看原文入口 |

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。

请注意：

1、关于重试的消息排重，推荐使用msgid排重。

2、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。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。详情请见“发送消息-被动回复消息”。

3、如果开发者需要对用户消息在5秒内立即做出回应，即使用“发送消息-被动回复消息”接口向用户被动回复消息时，可以在

公众平台官网的开发者中心处设置消息加密。开启加密后，用户发来的消息和开发者回复的消息都会被加密（但开发者通过客服

接口等API调用形式向用户发送消息，则不受影响）。关于消息加解密的详细说明，请见“发送消息-被动回复消息加解密说明”。

各消息类型的推送XML数据包结构如下：

**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（由系统生成） |
| 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)

请注意，开通语音识别后，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Recongnition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：

|  |
| --- |
| <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>  <Recognition><![CDATA[腾讯微信团队]]></Recognition>  <MsgId>1234567890123456</MsgId>  </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 |

**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)

**小视频消息**

|  |
| --- |
| <xml>  <ToUserName><![CDATA[toUser]]></ToUserName>  <FromUserName><![CDATA[fromUser]]></FromUserName>  <CreateTime>1357290913</CreateTime>  <MsgType><![CDATA[shortvideo]]></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 | 小视频为shortvideo |
| 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 |

[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=%E9%93%BE%E6%8E%A5%E6%B6%88%E6%81%AF)

接收事件推送

在微信用户和公众号产生交互的过程中，用户的某些操作会使得微信服务器通过事件推送的形式通知到开发者在开发者中心处设置的服务器地址，从而开发者可以获取到该信息。其中，某些事件推送在发生后，是允许开发者回复用户的，某些则不允许，详细内容如下：

目录

[1 关注/取消关注事件](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#1)

[2 扫描带参数二维码事件](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#2)

[3 上报地理位置事件](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#3)

[4 自定义菜单事件](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#4)

[5 点击菜单拉取消息时的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#5)

[6 点击菜单跳转链接时的事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140454&t=0.7736900574947048#6)

**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 |

[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)

被动回复用户消息

当用户发送消息给公众号时（或某些特定的用户操作引发的事件推送时），会产生一个POST请求，开发者可以在响应包（Get）中返回特定XML结构，来对该消息进行响应（现支持回复文本、图片、图文、语音、视频、音乐）。严格来说，发送被动响应消息其实并不是一种接口，而是对微信服务器发过来消息的一次回复。

微信服务器在将用户的消息发给公众号的开发者服务器地址（开发者中心处配置）后，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，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。关于重试的消息排重，有msgid的消息推荐使用msgid排重。事件类型消息推荐使用FromUserName + CreateTime 排重。

如果开发者希望增强安全性，可以在开发者中心处开启消息加密，这样，用户发给公众号的消息以及公众号被动回复用户消息都会继续加密（但），详见被动回复[消息加解密说明](https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419318479&token=&lang=zh_CN)。

假如服务器无法保证在五秒内处理并回复，必须做出下述回复，这样微信服务器才不会对此作任何处理，并且不会发起重试（这种情况下，可以使用客服消息接口进行异步回复），否则，将出现严重的错误提示。详见下面说明：

|  |
| --- |
| 1、直接回复success（推荐方式）  2、直接回复空串（指字节长度为0的空字符串，而不是XML结构体中content字段的内容为空） |

**一旦遇到以下情况，微信都会在公众号会话中，向用户下发系统提示“该公众号暂时无法提供服务，请稍后再试”：**

|  |
| --- |
| 1、开发者在5秒内未回复任何内容  2、开发者回复了异常数据，比如JSON数据等 |

另外，请注意，回复图片（不支持gif动图）等多媒体消息时需要预先通过素材管理接口上传临时素材到微信服务器，可以使用素材管理中的临时素材，也可以使用永久素材。

各消息类型需要的XML数据包结构如下：

目录

[1 回复文本消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#1)

[2 回复图片消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#2)

[3 回复语音消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#3)

[4 回复视频消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#4)

[5 回复音乐消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#5)

[6 回复图文消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140543&t=0.8145025925308185#6)

**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 | 通过素材管理中的接口上传多媒体文件，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 | 通过素材管理中的接口上传多媒体文件，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 | 通过素材管理中的接口上传多媒体文件，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，通过素材管理中的接口上传多媒体文件，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8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8，则将会无响应 |
| Title | 是 | 图文消息标题 |
| Description | 是 | 图文消息描述 |
| PicUrl | 是 | 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 | 是 | 点击图文消息跳转链接 |

## 消息加解密说明

公众号消息加解密是公众平台为了进一步加强公众号安全保障，提供的新机制。开发者需注意，公众账号主动调用API的情况将不受影响。只有被动回复用户的消息时，才需要进行消息加解密。消息加解密的具体修改包括：

1.新增消息体签名验证，用于公众平台和公众账号验证消息体的正确性

2.针对推送给微信公众账号的普通消息和事件消息，以及推送给设备公众账号的设备消息进行加密

3.公众账号对密文消息的回复也要求加密

请开发者查看接入指引和开发者FAQ来接入消息体签名及加解密功能：接入指引，开发者FAQ，若关注技术实现，可查看技术方案：技术方案

启用加解密功能（即选择兼容模式或安全模式）后，公众平台服务器在向公众账号服务器配置地址（可在“开发者中心”修改）推送消息时，URL将新增加两个参数（加密类型和消息体签名），并以此来体现新功能。加密算法采用AES，具体的加解密流程和方案请看接入指引、技术方案和示例代码。

为了配合消息加密功能的上线，并帮助开发者适配新特性，公众平台提供了3种加解密的模式供开发者选择，即明文模式、兼容模式、安全模式(可在“开发者中心”选择相应模式),选择兼容模式和安全模式前，需在开发者中心填写消息加解密密钥EncodingAESKey。

明文模式：维持现有模式，没有适配加解密新特性，消息体明文收发，默认设置为明文模式

兼容模式：公众平台发送消息内容将同时包括明文和密文，消息包长度增加到原来的3倍左右；公众号回复明文或密文均可，不影响现有消息收发；开发者可在此模式下进行调试

安全模式（推荐）：公众平台发送消息体的内容只含有密文，公众账号回复的消息体也为密文，建议开发者在调试成功后使用此模式收发消息

什么是EncodingAESKey？

微信公众平台采用AES对称加密算法对推送给公众帐号的消息体对行加密，EncodingAESKey则是加密所用的秘钥。公众帐号用此秘钥对收到的密文消息体进行解密，回复消息体也用此秘钥加密。

此外，微信公众平台为开发者提供了5种语言的示例代码（包括C++、php、Java、Python和C#版本，[点击下载](https://wximg.gtimg.com/shake_tv/mpwiki/cryptoDemo.zip)）。

## 客服消息

当用户和公众号产生特定动作的交互时（具体动作列表请见下方说明），微信将会把消息数据推送给开发者，开发者可以在一段时间内（目前修改为48小时）调用客服接口，通过POST一个JSON数据包来发送消息给普通用户。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。

目前允许的动作列表如下（公众平台会根据运营情况更新该列表，不同动作触发后，允许的客服接口下发消息条数不同，下发条数达到上限后，会遇到错误返回码，具体请见返回码说明页）：

1、用户发送信息

2、点击自定义菜单（仅有点击推事件、扫码推事件、扫码推事件且弹出“消息接收中”提示框这3种菜单类型是会触发客服接口的）

3、关注公众号

4、扫描二维码

5、支付成功

6、用户维权

为了帮助公众号使用不同的客服身份服务不同的用户群体，客服接口进行了升级，开发者可以管理客服账号，并设置客服账号的头像和昵称。该能力针对所有拥有客服接口权限的公众号开放。

另外，请开发者注意，本接口中所有使用到media\_id的地方，现在都可以使用素材管理中的永久素材media\_id了。

目录

[1 客服帐号管理](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1)

[1.1 添加客服帐号](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.1)

[1.2 修改客服帐号](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.2)

[1.3 删除客服帐号](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.3)

[1.4 设置客服帐号的头像](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.4)

[1.5 获取所有客服账号](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.5)

[1.6 接口的统一参数说明](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#1.6)

[2 客服接口-发消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140547&t=0.7991761384078979#2)

## ****客服帐号管理****

开发者在根据开发文档的要求完成开发后，使用6.0.2版及以上版本的微信用户在与公众号进行客服沟通，公众号使用不同的客服账号进行回复后，用户可以看到对应的客服头像和昵称。

**请注意，必须先在公众平台官网为公众号设置微信号后才能使用该能力。**

### ****添加客服帐号****

开发者可以通过本接口为公众号添加客服账号，每个公众号最多添加10个客服账号。该接口调用请求如下：

http请求方式: POST

https://api.weixin.qq.com/customservice/kfaccount/add?access\_token=ACCESS\_TOKEN

POST数据示例如下：

{

     "kf\_account" : "test1@test",

     "nickname" : "客服1",

     "password" : "pswmd5",

}

返回说明（正确时的JSON返回结果）：

{

     "errcode" : 0,

     "errmsg" : "ok",

}

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### ****修改客服帐号****

开发者可以通过本接口为公众号修改客服账号。该接口调用请求如下：

http请求方式: POST

https://api.weixin.qq.com/customservice/kfaccount/update?access\_token=ACCESS\_TOKEN

POST数据示例如下：

{

     "kf\_account" : "test1@test",

     "nickname" : "客服1",

     "password" : "pswmd5",

}

返回说明（正确时的JSON返回结果）：

{

     "errcode" : 0,

     "errmsg" : "ok",

}

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### ****删除客服帐号****

开发者可以通过该接口为公众号删除客服帐号。该接口调用请求如下：

http请求方式: GET

https://api.weixin.qq.com/customservice/kfaccount/del?access\_token=ACCESS\_TOKEN

POST数据示例如下：

{

     "kf\_account" : "test1@test",

     "nickname" : "客服1",

     "password" : "pswmd5",

}

返回说明（正确时的JSON返回结果）：

{

     "errcode" : 0,

     "errmsg" : "ok",

}

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### ****设置客服帐号的头像****

开发者可调用本接口来上传图片作为客服人员的头像，头像图片文件必须是jpg格式，推荐使用640\*640大小的图片以达到最佳效果。该接口调用请求如下：

http请求方式: POST/FORM

http://api.weixin.qq.com/customservice/kfaccount/uploadheadimg?access\_token=ACCESS\_TOKEN&kf\_account=KFACCOUNT

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命令的具体用法请自行了解

返回说明（正确时的JSON返回结果）：

{

     "errcode" : 0,

     "errmsg" : "ok",

}

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### ****获取所有客服账号****

开发者通过本接口，获取公众号中所设置的客服基本信息，包括客服工号、客服昵称、客服登录账号。

http请求方式: GET

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

返回说明（正确时的JSON返回结果）：

{

    "kf\_list": [

        {

            "kf\_account": "test1@test",

            "kf\_nick": "ntest1",

            "kf\_id": "1001"

            "kf\_headimgurl": " http://mmbiz.qpic.cn/mmbiz/4whpV1VZl2iccsvYbHvnphkyGtnvjfUS8Ym0GSaLic0FD3vN0V8PILcibEGb2fPfEOmw/0"

        },

        {

            "kf\_account": "test2@test",

            "kf\_nick": "ntest2",

            "kf\_id": "1002"

            "kf\_headimgurl": " http://mmbiz.qpic.cn/mmbiz/4whpV1VZl2iccsvYbHvnphkyGtnvjfUS8Ym0GSaLic0FD3vN0V8PILcibEGb2fPfEOmw /0"

        },

        {

            "kf\_account": "test3@test",

            "kf\_nick": "ntest3",

            "kf\_id": "1003"

            "kf\_headimgurl": " http://mmbiz.qpic.cn/mmbiz/4whpV1VZl2iccsvYbHvnphkyGtnvjfUS8Ym0GSaLic0FD3vN0V8PILcibEGb2fPfEOmw /0"

        }

    ]

}

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### ****接口的统一参数说明****

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 调用接口凭证 |
| kf\_account | 是 | 完整客服账号，格式为：账号前缀@公众号微信号 |
| kf\_nick | 是 | 客服昵称 |
| kf\_id | 是 | 客服工号 |
| nickname | 是 | 客服昵称，最长6个汉字或12个英文字符 |
| password | 否 | 客服账号登录密码，格式为密码明文的32位加密MD5值。该密码仅用于在公众平台官网的多客服功能中使用，若不使用多客服功能，则不必设置密码 |
| media | 是 | 该参数仅在设置客服头像时出现，是form-data中媒体文件标识，有filename、filelength、content-type等信息 |

## ****客服接口-发消息****

**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=ACCESS\_TOKEN

各消息类型所需的JSON数据包如下：

发送文本消息

{

    "touser":"OPENID",

    "msgtype":"text",

    "text":

    {

         "content":"Hello World"

    }

}

发送图片消息

{

    "touser":"OPENID",

    "msgtype":"image",

    "image":

    {

      "media\_id":"MEDIA\_ID"

    }

}

发送语音消息

{

    "touser":"OPENID",

    "msgtype":"voice",

    "voice":

    {

      "media\_id":"MEDIA\_ID"

    }

}

发送视频消息

{

    "touser":"OPENID",

    "msgtype":"video",

    "video":

    {

      "media\_id":"MEDIA\_ID",

      "thumb\_media\_id":"MEDIA\_ID",

      "title":"TITLE",

      "description":"DESCRIPTION"

    }

}

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"

    }

}

发送图文消息（点击跳转到外链） 图文消息条数限制在8条以内，注意，如果图文数超过8，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"

         }

         ]

    }

}

发送图文消息（点击跳转到图文消息页面） 图文消息条数限制在8条以内，注意，如果图文数超过8，则将会无响应。

{

    "touser":"OPENID",

    "msgtype":"mpnews",

    "mpnews":

    {

         "media\_id":"MEDIA\_ID"

    }

}

发送卡券

{

  "touser":"OPENID",

  "msgtype":"wxcard",

  "wxcard":{

           "card\_id":"123dsdajkasd231jhksad"

            },

}

特别注意客服消息接口投放卡券仅支持非自定义Code码和导入code模式的卡券的卡券，详情请见：[是否自定义code码](http://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1451025056&token=&lang=zh_CN&anchor=2.2.2)。

请注意，如果需要以某个客服帐号来发消息（在微信6.0.2及以上版本中显示自定义头像），则需在JSON数据包的后半部分加入customservice参数，例如发送文本消息则改为：

{

    "touser":"OPENID",

    "msgtype":"text",

    "text":

    {

         "content":"Hello World"

    },

    "customservice":

    {

         "kf\_account": "test1@kftest"

    }

}

|  |  |  |
| --- | --- | --- |
| 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，文本为text，图片为image，语音为voice，视频消息为video，音乐消息为music，图文消息（点击跳转到外链）为news，图文消息（点击跳转到图文消息页面）为mpnews，卡券为wxcard |
| content | 是 | 文本消息内容 |
| media\_id | 是 | 发送的图片/语音/视频/图文消息（点击跳转到图文消息页）的媒体ID |
| thumb\_media\_id | 是 | 缩略图的媒体ID |
| title | 否 | 图文消息/视频消息/音乐消息的标题 |
| description | 否 | 图文消息/视频消息/音乐消息的描述 |
| 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 |
| url | 否 | 图文消息被点击后跳转的链接 |
| 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 |

**接口返回说明**

返回数据示例（正确时的JSON返回结果）：

## 发送消息-群发接口和原创校验

在公众平台网站上，为订阅号提供了每天一条的群发权限，为服务号提供每月（自然月）4条的群发权限。而对于某些具备开发能力的公众号运营者，可以通过高级群发接口，实现更灵活的群发能力。

请注意：

1、对于认证订阅号，群发接口每天可成功调用1次，此次群发可选择发送给全部用户或某个标签；

2、对于认证服务号虽然开发者使用高级群发接口的每日调用限制为100次，但是用户每月只能接收4条，无论在公众平台网站上，还是使用接口群发，用户每月只能接收4条群发消息，多于4条的群发将对该用户发送失败；

3、开发者可以使用预览接口校对消息样式和排版，通过预览接口可发送编辑好的消息给指定用户校验效果；

4、群发过程中，微信后台会自动进行图文消息原创校验，请提前设置好相关参数(send\_ignore等)；

5、开发者可以主动设置 clientmsgid 来避免重复推送。

6、群发接口每分钟限制请求60次，超过限制的请求会被拒绝。

7、图文消息正文中插入自己帐号和其他公众号已群发文章链接的能力。

群发图文消息的过程如下：

1、首先，预先将图文消息中需要用到的图片，使用上传图文消息内图片接口，上传成功并获得图片 URL；

2、上传图文消息素材，需要用到图片时，请使用上一步获取的图片 URL；

3、使用对用户标签的群发，或对 OpenID 列表的群发，将图文消息群发出去，群发时微信会进行原创校验，并返回群发操作结果；

4、在上述过程中，如果需要，还可以预览图文消息、查询群发状态，或删除已群发的消息等。

群发图片、文本等其他消息类型的过程如下：

1、如果是群发文本消息，则直接根据下面的接口说明进行群发即可；

2、如果是群发图片、视频等消息，则需要预先通过素材管理接口准备好 mediaID。

关于群发时使用is\_to\_all为true使其进入公众号在微信客户端的历史消息列表：

1、使用is\_to\_all为true且成功群发，会使得此次群发进入历史消息列表。

2、为防止异常，认证订阅号在一天内，只能使用is\_to\_all为true进行群发一次，或者在公众平台官网群发（不管本次群发是对全体还是对某个分组）一次。以避免一天内有2条群发进入历史消息列表。

3、类似地，服务号在一个月内，使用is\_to\_all为true群发的次数，加上公众平台官网群发（不管本次群发是对全体还是对某个分组）的次数，最多只能是4次。

4、设置is\_to\_all为false时是可以多次群发的，但每个用户只会收到最多4条，且这些群发不会进入历史消息列表。

另外，请开发者注意，本接口中所有使用到media\_id的地方，现在都可以使用素材管理中的永久素材media\_id了。**请但注意，使用同一个素材群发出去的链接是一样的，这意味着，删除某一次群发，会导致整个链接失效。**

目录

[1 上传图文消息内的图片获取URL【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#1)

[2 上传图文消息素材【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#2)

[3 根据标签进行群发【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#3)

[4 根据OpenID列表群发【订阅号不可用，服务号认证后可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#4)

[5 删除群发【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#5)

[6 预览接口【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#6)

[7 查询群发消息发送状态【订阅号与服务号认证后均可用】](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#7)

[8 事件推送群发结果](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#8)

[9 使用 clientmsgid 参数，避免重复推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.16121174638452018#9)

### 上传图文消息内的图片获取URL【订阅号与服务号认证后均可用】

请注意，本接口所上传的图片不占用公众号的素材库中图片数量的5000个的限制。图片仅支持jpg/png格式，大小必须在1MB以下。

**接口调用请求说明**

http请求方式: POST

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

调用示例（使用curl命令，用FORM表单方式上传一个图片）：

curl -F media=@test.jpg "https://api.weixin.qq.com/cgi-bin/media/uploadimg?access\_token=ACCESS\_TOKEN"

**参数说明**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 调用接口凭证 |
| media | 是 | form-data中媒体文件标识，有filename、filelength、content-type等信息 |

**返回说明** 正常情况下的返回结果为：

{

    "url":  "http://mmbiz.qpic.cn/mmbiz/gLO17UPS6FS2xsypf378iaNhWacZ1G1UplZYWEYfwvuU6Ont96b1roYs CNFwaRrSaKTPCUdBK9DgEHicsKwWCBRQ/0"

}

其中url就是上传图片的URL，可用于后续群发中，放置到图文消息中。

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### 上传图文消息素材【订阅号与服务号认证后均可用】

**接口调用请求说明**

http请求方式: POST

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

**POST数据说明**

POST数据示例如下：

{

   "articles": [

 {

                        "thumb\_media\_id":"qI6\_Ze\_6PtV7svjolgs-rN6stStuHIjs9\_DidOHaj0Q-mwvBelOXCFZiq2OsIU-p",

                        "author":"xxx",

 "title":"Happy Day",

 "content\_source\_url":"www.qq.com",

 "content":"content",

 "digest":"digest",

                        "show\_cover\_pic":1

 },

 {

                        "thumb\_media\_id":"qI6\_Ze\_6PtV7svjolgs-rN6stStuHIjs9\_DidOHaj0Q-mwvBelOXCFZiq2OsIU-p",

                        "author":"xxx",

 "title":"Happy Day",

 "content\_source\_url":"www.qq.com",

 "content":"content",

 "digest":"digest",

                        "show\_cover\_pic":0

 }

   ]

}

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| Articles | 是 | 图文消息，一个图文消息支持1到8条图文 |
| thumb\_media\_id | 是 | 图文消息缩略图的media\_id，可以在基础支持-上传多媒体文件接口中获得 |
| author | 否 | 图文消息的作者 |
| title | 是 | 图文消息的标题 |
| content\_source\_url | 否 | 在图文消息页面点击“阅读原文”后的页面，受安全限制，如需跳转Appstore，可以使用itun.es或appsto.re的短链服务，并在短链后增加 #wechat\_redirect 后缀。 |
| content | 是 | 图文消息页面的内容，支持HTML标签。具备微信支付权限的公众号，可以使用a标签，其他公众号不能使用，如需插入小程序卡片，可参考下文。 |
| digest | 否 | 图文消息的描述，如本字段为空，则默认抓取正文前64个字 |
| show\_cover\_pic | 否 | 是否显示封面，1为显示，0为不显示 |

**如果需要在群发图文中插入小程序卡片，则在调用上传图文消息素材接口时，需在content字段中添加小程序卡片页面**

代码示例：

<mp-miniprogram data-miniprogram-appid="wx123123123" data-miniprogram-path="pages/index/index" data-miniprogram-title="小程序示例" data-progarm-imageurl="http://mmbizqbic.cn/demo.jpg"></mp-miniprogram>

**参数说明**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| data-miniprogram-appid | 是 | 小程序的AppID |
| data-miniprogram-path | 是 | 点击小程序卡片时要打开的路径 |
| data-miniprogram-title | 是 | 小程序卡片的标题，不超过35个字 |
| data-miniprogram-imageurl | 是 | 小程序卡片的封面图链接，图片必须为1080\*864像素 |

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

   "type":"news",

   "media\_id":"CsEf3ldqkAYJAU6EJeIkStVDSvffUJ54vqbThMgplD-VJXXof6ctX5fI6-aYyUiQ",

   "created\_at":1391857799

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| type | 媒体文件类型，分别有图片（image）、语音（voice）、视频（video）和缩略图（thumb），图文消息（news） |
| media\_id | 媒体文件/图文消息上传后获取的唯一标识 |
| created\_at | 媒体文件上传时间 |

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### 图文消息群发前将进行原创校验

**一、群发接口新增原创校验流程**

开发者调用群发接口进行图文消息的群发时，微信会将开发者准备群发的文章，与公众平台原创库中的文章进行比较，校验结果分为以下几种：

1. 当前准备群发的文章，未命中原创库中的文章，则可以群发。

2. 当前准备群发的文章，已命中原创库中的文章，则：

    2.1 若原创作者允许转载该文章，则可以进行群发。群发时，会自动替换成原文的样式，且会自动将文章注明为转载并显示来源。

          若希望修改原文内容或样式，或群发时不显示转载来源，可自行与原创公众号作者联系并获得授权之后再进行群发。

    2.2 若原创作者禁止转载该文章，则不能进行群发。

          若希望转载该篇文章，可自行与原创公众号作者联系并获得授权之后再进行群发。

**二、群发接口新增 send\_ignore\_reprint 参数**

群发接口新增 send\_ignore\_reprint 参数，开发者可以对群发接口的 send\_ignore\_reprint 参数进行设置，指定待群发的文章被判定为转载时，是否继续群发。

当 send\_ignore\_reprint 参数设置为1时，文章被判定为转载时，且原创文允许转载时，将继续进行群发操作。

当 send\_ignore\_reprint 参数设置为0时，文章被判定为转载时，将停止群发操作。

send\_ignore\_reprint 默认为0。

群发操作的相关返回码，可以参考全局返回码说明文档。

### 根据标签进行群发【订阅号与服务号认证后均可用】

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

图文消息（注意图文消息的media\_id需要通过上述方法来得到）：

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":2

   },

   "mpnews":{

      "media\_id":"123dsdajkasd231jhksad"

   },

    "msgtype":"mpnews",

    "send\_ignore\_reprint":0

}

文本：

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":2

   },

   "text":{

      "content":"CONTENT"

   },

    "msgtype":"text"

}

语音/音频（注意此处media\_id需通过基础支持中的上传下载多媒体文件来得到）：

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":2

   },

   "voice":{

      "media\_id":"123dsdajkasd231jhksad"

   },

    "msgtype":"voice"

}

图片（注意此处media\_id需通过基础支持中的上传下载多媒体文件来得到）：

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":2

   },

   "image":{

      "media\_id":"123dsdajkasd231jhksad"

   },

    "msgtype":"image"

}

视频

请注意，此处视频的media\_id需通过POST请求到下述接口特别地得到：[https://api.weixin.qq.com/cgi-bin/media/uploadvideo?access\_token=ACCESS\_TOKEN](https://file.api.weixin.qq.com/cgi-bin/media/uploadvideo?access_token=ACCESS_TOKEN) POST数据如下（此处media\_id需通过基础支持中的上传下载多媒体文件来得到）：

{

  "media\_id": "rF4UdIMfYK3efUfyoddYRMU50zMiRmmt\_l0kszupYh\_SzrcW5Gaheq05p\_lHuOTQ",

  "title": "TITLE",

  "description": "Description"

}

返回将为

{

  "type":"video",

  "media\_id":"IhdaAQXuvJtGzwwc0abfXnzeezfO0NgPK6AQYShD8RQYMTtfzbLdBIQkQziv2XJc",

  "created\_at":1398848981

}

然后，POST下述数据（将media\_id改为上一步中得到的media\_id），即可进行发送

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":2

   },

   "mpvideo":{

      "media\_id":"IhdaAQXuvJtGzwwc0abfXnzeezfO0NgPK6AQYShD8RQYMTtfzbLdBIQkQziv2XJc"

   },

    "msgtype":"mpvideo"

}

卡券消息（注意图文消息的media\_id需要通过上述方法来得到）：

{

   "filter":{

      "is\_to\_all":false,

      "tag\_id":"2"

   },

  "wxcard":{

           "card\_id":"123dsdajkasd231jhksad"

            },

   "msgtype":"wxcard"

}

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| filter | 是 | 用于设定图文消息的接收者 |
| is\_to\_all | 否 | 用于设定是否向全部用户发送，值为true或false，选择true该消息群发给所有用户，选择false可根据tag\_id发送给指定群组的用户 |
| tag\_id | 否 | 群发到的标签的tag\_id，参加用户管理中用户分组接口，若is\_to\_all值为true，可不填写tag\_id |
| mpnews | 是 | 用于设定即将发送的图文消息 |
| media\_id | 是 | 用于群发的消息的media\_id |
| msgtype | 是 | 群发的消息类型，图文消息为mpnews，文本消息为text，语音为voice，音乐为music，图片为image，视频为video，卡券为wxcard |
| title | 否 | 消息的标题 |
| description | 否 | 消息的描述 |
| thumb\_media\_id | 是 | 视频缩略图的媒体ID |
| send\_ignore\_reprint | 是 | 图文消息被判定为转载时，是否继续群发。  1为继续群发（转载），0为停止群发。  该参数默认为0。 |

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

   "errcode":0,

   "errmsg":"send job submission success",

   "msg\_id":34182,

   "msg\_data\_id": 206227730

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| type | 媒体文件类型，分别有图片（image）、语音（voice）、视频（video）和缩略图（thumb），图文消息为news |
| errcode | 错误码 |
| errmsg | 错误信息 |
| msg\_id | 消息发送任务的ID |
| msg\_data\_id | 消息的数据ID，该字段只有在群发图文消息时，才会出现。可以用于在图文分析数据接口中，获取到对应的图文消息的数据，是图文分析数据接口中的msgid字段中的前半部分，详见图文分析数据接口中的msgid字段的介绍。 |

请注意：**在返回成功时，意味着群发任务提交成功，并不意味着此时群发已经结束**，所以，仍有可能在后续的发送过程中出现异常情况导致用户未收到消息，如消息有时会进行审核、服务器不稳定等。此外，**群发任务一般需要较长的时间才能全部发送完毕，请耐心等待**。

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### 根据OpenID列表群发【订阅号不可用，服务号认证后可用】

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/message/mass/send?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

图文消息（注意图文消息的media\_id需要通过上述方法来得到）：

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

   "mpnews":{

      "media\_id":"123dsdajkasd231jhksad"

   },

    "msgtype":"mpnews"，

    "send\_ignore\_reprint":0

}

文本：

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

    "msgtype": "text",

    "text": { "content": "hello from boxer."}

}

语音：

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

   "voice":{

      "media\_id":"mLxl6paC7z2Tl-NJT64yzJve8T9c8u9K2x-Ai6Ujd4lIH9IBuF6-2r66mamn\_gIT"

   },

    "msgtype":"voice"

}

图片：

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

   "image":{

      "media\_id":"BTgN0opcW3Y5zV\_ZebbsD3NFKRWf6cb7OPswPi9Q83fOJHK2P67dzxn11Cp7THat"

   },

    "msgtype":"image"

}

视频：

请注意，此处视频的media\_id需通过POST请求到下述接口特别地得到： https://api.weixin.qq.com/cgi-bin/media/uploadvideo?access\_token=ACCESS\_TOKEN POST数据如下（此处media\_id需通过基础支持中的上传下载多媒体文件来得到）：

{

  "media\_id": "rF4UdIMfYK3efUfyoddYRMU50zMiRmmt\_l0kszupYh\_SzrcW5Gaheq05p\_lHuOTQ",

  "title": "TITLE",

  "description": "Description"

}

返回将为

{

  "type":"video",

  "media\_id":"IhdaAQXuvJtGzwwc0abfXnzeezfO0NgPK6AQYShD8RQYMTtfzbLdBIQkQziv2XJc",

  "created\_at":1398848981

}

然后，POST下述数据（将media\_id改为上一步中得到的media\_id），即可进行发送

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

   "mpvideo":{

      "media\_id":"123dsdajkasd231jhksad",

      "title":"TITLE",

      "description":"DESCRIPTION"

   },

    "msgtype":"mpvideo"

}

卡券：

{

   "touser":[

    "OPENID1",

    "OPENID2"

   ],

        "wxcard": {"card\_id":"123dsdajkasd231jhksad"}

        "msgtype":"wxcard"

}

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| touser | 是 | 填写图文消息的接收者，一串OpenID列表，OpenID最少2个，最多10000个 |
| mpnews | 是 | 用于设定即将发送的图文消息 |
| media\_id | 是 | 用于群发的图文消息的media\_id |
| msgtype | 是 | 群发的消息类型，图文消息为mpnews，文本消息为text，语音为voice，音乐为music，图片为image，视频为video，卡券为wxcard |
| title | 否 | 消息的标题 |
| description | 否 | 消息的描述 |
| thumb\_media\_id | 是 | 视频缩略图的媒体ID |
| send\_ignore\_reprint | 是 | 图文消息被判定为转载时，是否继续群发。  1为继续群发（转载），0为停止群发。  该参数默认为0。 |

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

   "errcode":0,

   "errmsg":"send job submission success",

   "msg\_id":34182,

   "msg\_data\_id": 206227730

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| type | 媒体文件类型，分别有图片（image）、语音（voice）、视频（video）和缩略图（thumb），次数为news，即图文消息 |
| errcode | 错误码 |
| errmsg | 错误信息 |
| msg\_id | 消息发送任务的ID |
| msg\_data\_id | 消息的数据ID，，该字段只有在群发图文消息时，才会出现。可以用于在图文分析数据接口中，获取到对应的图文消息的数据，是图文分析数据接口中的msgid字段中的前半部分，详见图文分析数据接口中的msgid字段的介绍。 |

请注意：**在返回成功时，意味着群发任务提交成功，并不意味着此时群发已经结束**，所以，仍有可能在后续的发送过程中出现异常情况导致用户未收到消息，如消息有时会进行审核、服务器不稳定等。此外，**群发任务一般需要较长的时间才能全部发送完毕，请耐心等待**。

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### 删除群发【订阅号与服务号认证后均可用】

群发之后，随时可以通过该接口删除群发。

**接口调用请求说明**

http请求方式: POST

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

**POST数据说明**

POST数据示例如下：

{

   "msg\_id":30124,

   "article\_idx":2

}

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| msg\_id | 是 | 发送出去的消息ID |
| article\_idx | 否 | 要删除的文章在图文消息中的位置，第一篇编号为1，该字段不填或填0会删除全部文章 |

请注意：

1、只有已经发送成功的消息才能删除

2、删除消息是将消息的图文详情页失效，已经收到的用户，还是能在其本地看到消息卡片。

3、删除群发消息只能删除图文消息和视频消息，其他类型的消息一经发送，无法删除。

4、如果多次群发发送的是一个图文消息，那么删除其中一次群发，就会删除掉这个图文消息也，导致所有群发都失效

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

   "errcode":0,

   "errmsg":"ok"

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| errcode | 错误码 |
| errmsg | 错误信息 |

错误时微信会返回错误码等信息，请根据错误码查询错误信息

### 预览接口【订阅号与服务号认证后均可用】

开发者可通过该接口发送消息给指定用户，在手机端查看消息的样式和排版。**为了满足第三方平台开发者的需求，在保留对openID预览能力的同时，增加了对指定微信号发送预览的能力，但该能力每日调用次数有限制（100次），请勿滥用。**

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/message/mass/preview?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

图文消息（其中media\_id与根据分组群发中的media\_id相同）：

{

   "touser":"OPENID",

   "mpnews":{

            "media\_id":"123dsdajkasd231jhksad"

             },

   "msgtype":"mpnews"

}

文本：

{

    "touser":"OPENID",

    "text":{

           "content":"CONTENT"

           },

    "msgtype":"text"

}

语音（其中media\_id与根据分组群发中的media\_id相同）：

{

    "touser":"OPENID",

    "voice":{

            "media\_id":"123dsdajkasd231jhksad"

            },

    "msgtype":"voice"

}

图片（其中media\_id与根据分组群发中的media\_id相同）：

{

    "touser":"OPENID",

    "image":{

            "media\_id":"123dsdajkasd231jhksad"

            },

    "msgtype":"image"

}

视频（其中media\_id与根据分组群发中的media\_id相同）：

{

    "touser":"OPENID",

    "mpvideo":{  "media\_id":"IhdaAQXuvJtGzwwc0abfXnzeezfO0NgPK6AQYShD8RQYMTtfzbLdBIQkQziv2XJc",

               },

    "msgtype":"mpvideo"

}

卡券：

{ "touser":"OPENID",

  "wxcard":{

           "card\_id":"123dsdajkasd231jhksad",

            "card\_ext": "{\"code\":\"\",\"openid\":\"\",\"timestamp\":\"1402057159\",\"signature\":\"017bb17407c8e0058a66d72dcc61632b70f511ad\"}"

            },

  "msgtype":"wxcard"

}

请注意，上述JSON数据中的touser字段都可以改为towxname，这样就可以针对微信号进行预览（而非openID），towxname和touser同时赋值时，以towxname优先。修改后JSON数据如下（以图文消息为例）： 图文消息：

{

   "towxname":"示例的微信号",

   "mpnews":{

            "media\_id":"123dsdajkasd231jhksad"

             },

   "msgtype":"mpnews"

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| touser | 接收消息用户对应该公众号的openid，该字段也可以改为towxname，以实现对微信号的预览 |
| msgtype | 群发的消息类型，图文消息为mpnews，文本消息为text，语音为voice，音乐为music，图片为image，视频为video，卡券为wxcard |
| media\_id | 用于群发的消息的media\_id |
| content | 发送文本消息时文本的内容 |

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

   "errcode":0,

   "errmsg":"preview success",

   "msg\_id":34182

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| errcode | 错误码 |
| errmsg | 错误信息 |
| msg\_id | 消息ID |

### 查询群发消息发送状态【订阅号与服务号认证后均可用】

**接口调用请求说明**

http请求方式: POST

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

**POST数据说明**

POST数据示例如下：

{

   "msg\_id": "201053012"

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| msg\_id | 群发消息后返回的消息id |

**返回说明**

返回数据示例（正确时的JSON返回结果）：

{

     "msg\_id":201053012,

     "msg\_status":"SEND\_SUCCESS"

}

|  |  |
| --- | --- |
| 参数 | 说明 |
| msg\_id | 群发消息后返回的消息id |
| msg\_status | 消息发送后的状态，SEND\_SUCCESS表示发送成功 |

### 事件推送群发结果

由于群发任务提交后，群发任务可能在一定时间后才完成，因此，群发接口调用时，仅会给出群发任务是否提交成功的提示，若群发任务提交成功，则在群发任务结束时，会向开发者在公众平台填写的开发者URL（callback URL）推送事件。

**需要注意，由于群发任务彻底完成需要较长时间，将会在群发任务即将完成的时候，就推送群发结果，此时的推送人数数据将会与实际情形存在一定误差**

推送的XML结构如下（发送成功时），已增加原创校验结果：

<xml>

<ToUserName><![CDATA[gh\_4d00ed8d6399]]></ToUserName>

<FromUserName><![CDATA[oV5CrjpxgaGXNHIQigzNlgLTnwic]]></FromUserName>

<CreateTime>1481013459</CreateTime>

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

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

<MsgID>1000001625</MsgID>

<Status><![CDATA[err(30003)]]></Status>

<TotalCount>0</TotalCount>

<FilterCount>0</FilterCount>

<SentCount>0</SentCount>

<ErrorCount>0</ErrorCount>

<CopyrightCheckResult>

<Count>2</Count>

<ResultList>

<item>

<ArticleIdx>1</ArticleIdx>

<UserDeclareState>0</UserDeclareState>

<AuditState>2</AuditState>

<OriginalArticleUrl><![CDATA[Url\_1]]></OriginalArticleUrl>

<OriginalArticleType>1</OriginalArticleType>

<CanReprint>1</CanReprint>

<NeedReplaceContent>1</NeedReplaceContent>

<NeedShowReprintSource>1</NeedShowReprintSource>

</item>

<item>

<ArticleIdx>2</ArticleIdx>

<UserDeclareState>0</UserDeclareState>

<AuditState>2</AuditState>

<OriginalArticleUrl><![CDATA[Url\_2]]></OriginalArticleUrl>

<OriginalArticleType>1</OriginalArticleType>

<CanReprint>1</CanReprint>

<NeedReplaceContent>1</NeedReplaceContent>

<NeedShowReprintSource>1</NeedShowReprintSource>

</item>

</ResultList>

<CheckState>2</CheckState>

</CopyrightCheckResult>

</xml>

|  |  |
| --- | --- |
| 参数 | 说明 |
| ToUserName | 公众号的微信号 |
| FromUserName | 公众号群发助手的微信号，为mphelper |
| CreateTime | 创建时间的时间戳 |
| MsgType | 消息类型，此处为event |
| Event | 事件信息，此处为MASSSENDJOBFINISH |
| MsgID | 群发的消息ID |
| Status | 群发的结构，为“send success”或“send fail”或“err(num)”。但send success时，也有可能因用户拒收公众号的消息、系统错误等原因造成少量用户接收失败。err(num)是审核失败的具体原因，可能的情况如下：  err(10001), //涉嫌广告 err(20001), //涉嫌政治 err(20004), //涉嫌社会 err(20002), //涉嫌色情 err(20006), //涉嫌违法犯罪 err(20008), //涉嫌欺诈 err(20013), //涉嫌版权 err(22000), //涉嫌互推(互相宣传) err(21000), //涉嫌其他  err(30001) // 原创校验出现系统错误且用户选择了被判为转载就不群发  err(30002) // 原创校验被判定为不能群发  err(30003) // 原创校验被判定为转载文且用户选择了被判为转载就不群发 |
| TotalCount | tag\_id下粉丝数；或者openid\_list中的粉丝数 |
| FilterCount | 过滤（过滤是指特定地区、性别的过滤、用户设置拒收的过滤，用户接收已超4条的过滤）后，准备发送的粉丝数，原则上，FilterCount = SentCount + ErrorCount |
| SentCount | 发送成功的粉丝数 |
| ErrorCount | 发送失败的粉丝数 |
| ResultList | 各个单图文校验结果 |
| ArticleIdx | 群发文章的序号，从1开始 |
| UserDeclareState | 用户声明文章的状态 |
| AuditState | 系统校验的状态 |
| OriginalArticleUrl | 相似原创文的url |
| OriginalArticleType | 相似原创文的类型 |
| CanReprint | 是否能转载 |
| NeedReplaceContent | 是否需要替换成原创文内容 |
| NeedShowReprintSource | 是否需要注明转载来源 |
| CheckState | 整体校验结果 |
| CheckState | 1-未被判为转载，可以群发，2-被判为转载，可以群发，3-被判为转载，不能群发 |

### 使用 clientmsgid 参数，避免重复推送

一、群发接口新增 clientmsgid 参数，开发者调用群发接口时可以主动设置 clientmsgid 参数，避免重复推送。

群发时，微信后台将对 24 小时内的群发记录进行检查，如果该 clientmsgid 已经存在一条群发记录，则会拒绝本次群发请求，返回已存在的群发msgid，开发者可以调用“查询群发消息发送状态”接口查看该条群发的状态。

二、新增返回码

|  |  |
| --- | --- |
| 返回码 | 结果 |
| 45065 | 相同 clientmsgid 已存在群发记录，返回数据中带有已存在的群发任务的 msgid |
| 45066 | 相同 clientmsgid 重试速度过快，请间隔1分钟重试 |
| 45067 | clientmsgid 长度超过限制 |

三、接口示例及参数描述

|  |
| --- |
| {     "filter":{        "is\_to\_all":false,        "tag\_id":2     },     "mpnews":{        "media\_id":"123dsdajkasd231jhksad"     },      "msgtype":"mpnews",      "send\_ignore\_reprint":0,  "clientmsgid":"send\_tag\_2"  } |

参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| clientmsgid | 否 | 开发者侧群发msgid，长度限制64字节，如不填，则后台默认以群发范围和群发内容的摘要值做为clientmsgid |

返回说明

|  |
| --- |
| clientmsgid 冲突时的返回示例：  {     "errcode":45065,     "errmsg":"clientmsgid exist",     "msg\_id":123456  } |

模板消息接口

模板消息仅用于公众号向用户发送重要的服务通知，只能用于符合其要求的服务场景中，如信用卡刷卡通知，商品购买成功通知等。不支持广告等营销类消息以及其它所有可能对用户造成骚扰的消息。

关于使用规则，请注意：

1、所有服务号都可以在功能->添加功能插件处看到申请模板消息功能的入口，但只有认证后的服务号才可以申请模板消息的使用权限并获得该权限；

2、需要选择公众账号服务所处的2个行业，每月可更改1次所选行业；

3、在所选择行业的模板库中选用已有的模板进行调用；

4、每个账号可以同时使用25个模板。

5、当前每个账号的模板消息的日调用上限为10万次，单个模板没有特殊限制。【2014年11月18日将接口调用频率从默认的日1万次提升为日10万次，可在MP登录后的开发者中心查看】。当账号粉丝数超过10W/100W/1000W时，模板消息的日调用上限会相应提升，以公众号MP后台开发者中心页面中标明的数字为准。

关于接口文档，请注意：

1、模板消息调用时主要需要模板ID和模板中各参数的赋值内容；

2、模板中参数内容必须以".DATA"结尾，否则视为保留字；

3、模板保留符号"{{ }}"。

目录

[1 设置所属行业](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#1)

[2 获取设置的行业信息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#2)

[3 获得模板ID](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#3)

[4 获取模板列表](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#4)

[5 删除模板](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#5)

[6 发送模板消息](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#6)

[7 事件推送](https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.9374669316884718#7)

设置所属行业

设置行业可在微信公众平台后台完成，每月可修改行业1次，帐号仅可使用所属行业中相关的模板，为方便第三方开发者，提供通过接口调用的方式来修改账号所属行业，具体如下：

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/template/api\_set\_industry?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

      {

          "industry\_id1":"1",

          "industry\_id2":"4"

       }

参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 接口调用凭证 |
| industry\_id1 | 是 | 公众号模板消息所属行业编号 |
| industry\_id2 | 是 | 公众号模板消息所属行业编号 |

**行业代码查询**

|  |  |  |
| --- | --- | --- |
| 主行业 | 副行业 | 代码 |
| IT科技 | 互联网/电子商务 | 1 |
| IT科技 | IT软件与服务 | 2 |
| IT科技 | IT硬件与设备 | 3 |
| IT科技 | 电子技术 | 4 |
| IT科技 | 通信与运营商 | 5 |
| IT科技 | 网络游戏 | 6 |
| 金融业 | 银行 | 7 |
| 金融业 | 基金|理财|信托 | 8 |
| 金融业 | 保险 | 9 |
| 餐饮 | 餐饮 | 10 |
| 酒店旅游 | 酒店 | 11 |
| 酒店旅游 | 旅游 | 12 |
| 运输与仓储 | 快递 | 13 |
| 运输与仓储 | 物流 | 14 |
| 运输与仓储 | 仓储 | 15 |
| 教育 | 培训 | 16 |
| 教育 | 院校 | 17 |
| 政府与公共事业 | 学术科研 | 18 |
| 政府与公共事业 | 交警 | 19 |
| 政府与公共事业 | 博物馆 | 20 |
| 政府与公共事业 | 公共事业|非盈利机构 | 21 |
| 医药护理 | 医药医疗 | 22 |
| 医药护理 | 护理美容 | 23 |
| 医药护理 | 保健与卫生 | 24 |
| 交通工具 | 汽车相关 | 25 |
| 交通工具 | 摩托车相关 | 26 |
| 交通工具 | 火车相关 | 27 |
| 交通工具 | 飞机相关 | 28 |
| 房地产 | 建筑 | 29 |
| 房地产 | 物业 | 30 |
| 消费品 | 消费品 | 31 |
| 商业服务 | 法律 | 32 |
| 商业服务 | 会展 | 33 |
| 商业服务 | 中介服务 | 34 |
| 商业服务 | 认证 | 35 |
| 商业服务 | 审计 | 36 |
| 文体娱乐 | 传媒 | 37 |
| 文体娱乐 | 体育 | 38 |
| 文体娱乐 | 娱乐休闲 | 39 |
| 印刷 | 印刷 | 40 |
| 其它 | 其它 | 41 |

获取设置的行业信息

获取帐号设置的行业信息。可登录微信公众平台，在公众号后台中查看行业信息。为方便第三方开发者，提供通过接口调用的方式来获取帐号所设置的行业信息，具体如下:

**接口调用请求说明**

http请求方式：GET

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

**参数说明**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 接口调用凭证 |

**返回说明**

正确调用后的返回示例：

{

"primary\_industry":{"first\_class":"运输与仓储","second\_class":"快递"},

"secondary\_industry":{"first\_class":"IT科技","second\_class":"互联网|电子商务"}

}

返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| access\_token | 是 | 接口调用凭证 |
| primary\_industry | 是 | 帐号设置的主营行业 |
| secondary\_industry | 是 | 帐号设置的副营行业 |

获得模板ID

从行业模板库选择模板到帐号后台，获得模板ID的过程可在微信公众平台后台完成。为方便第三方开发者，提供通过接口调用的方式来获取模板ID，具体如下：

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/template/api\_add\_template?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

      {

           "template\_id\_short":"TM00015"

       }

参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 接口调用凭证 |
| template\_id\_short | 是 | 模板库中模板的编号，有“TM\*\*”和“OPENTMTM\*\*”等形式 |

**返回码说明**

在调用模板消息接口后，会返回JSON数据包。正常时的返回JSON数据包示例：

    {

           "errcode":0,

           "errmsg":"ok",

           "template\_id":"Doclyl5uP7Aciu-qZ7mJNPtWkbkYnWBWVja26EGbNyk"

       }

获取模板列表

获取已添加至帐号下所有模板列表，可在微信公众平台后台中查看模板列表信息。为方便第三方开发者，提供通过接口调用的方式来获取帐号下所有模板信息，具体如下:

**接口调用请求说明**

http请求方式：GET

https://api.weixin.qq.com/cgi-bin/template/get\_all\_private\_template?access\_token=ACCESS\_TOKEN

**参数说明**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 接口调用凭证 |

**返回说明**

正确调用后的返回示例：

{

 "template\_list": [{

      "template\_id": "iPk5sOIt5X\_flOVKn5GrTFpncEYTojx6ddbt8WYoV5s",

      "title": "领取奖金提醒",

      "primary\_industry": "IT科技",

      "deputy\_industry": "互联网|电子商务",

      "content": "{ {result.DATA} }\n\n领奖金额:{ {withdrawMoney.DATA} }\n领奖  时间:{ {withdrawTime.DATA} }\n银行信息:{ {cardInfo.DATA} }\n到账时间:  { {arrivedTime.DATA} }\n{ {remark.DATA} }",

      "example": "您已提交领奖申请\n\n领奖金额：xxxx元\n领奖时间：2013-10-10 12:22:22\n银行信息：xx银行(尾号xxxx)\n到账时间：预计xxxxxxx\n\n预计将于xxxx到达您的银行卡"

   }]

}

返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | **说明** |
| access\_token | 是 | 接口调用凭证 |
| template\_id | 是 | 模板ID |
| title | 是 | 模板标题 |
| primary\_industry | 是 | 模板所属行业的一级行业 |
| deputy\_industry | 是 | 模板所属行业的二级行业 |
| content | 是 | 模板内容 |
| example | 是 | 模板示例 |

删除模板

删除模板可在微信公众平台后台完成，为方便第三方开发者，提供通过接口调用的方式来删除某帐号下的模板，具体如下：

**接口调用请求说明**

http请求方式：POST

https://api.weixin.qq.com/cgi-bin/template/del\_private\_template?access\_token=ACCESS\_TOKEN

POST数据说明如下：

 {

     "template\_id" : "Dyvp3-Ff0cnail\_CDSzk1fIc6-9lOkxsQE7exTJbwUE"

 }

**参数说明**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 说明 |
| access\_token | 是 | 接口调用凭证 |
| template\_id | 是 | 公众帐号下模板消息ID |

**返回说明**

在调用接口后，会返回JSON数据包。正常时的返回JSON数据包示例：

{

   "errcode" : 0,

   "errmsg" : "ok"

}

发送模板消息

**接口调用请求说明**

http请求方式: POST

https://api.weixin.qq.com/cgi-bin/message/template/send?access\_token=ACCESS\_TOKEN

**POST数据说明**

POST数据示例如下：

      {

           "touser":"OPENID",

           "template\_id":"ngqIpbwh8bUfcSsECmogfXcV14J0tQlEpBO27izEYtY",

           "url":"http://weixin.qq.com/download",

           "miniprogram":{

             "appid":"xiaochengxuappid12345",

             "pagepath":"index?foo=bar"

           },

           "data":{

                   "first": {

                       "value":"恭喜你购买成功！",

                       "color":"#173177"

                   },

                   "keynote1":{

                       "value":"巧克力",

                       "color":"#173177"

                   },

                   "keynote2": {

                       "value":"39.8元",

                       "color":"#173177"

                   },

                   "keynote3": {

                       "value":"2014年9月22日",

                       "color":"#173177"

                   },

                   "remark":{

                       "value":"欢迎再次购买！",

                       "color":"#173177"

                   }

           }

       }

参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| touser | 是 | 接收者openid |
| template\_id | 是 | 模板ID |
| url | 否 | 模板跳转链接 |
| miniprogram | 否 | 跳小程序所需数据，不需跳小程序可不用传该数据 |
| appid | 是 | 所需跳转到的小程序appid（该小程序appid必须与发模板消息的公众号是绑定关联关系） |
| pagepath | 是 | 所需跳转到小程序的具体页面路径，支持带参数,（示例index?foo=bar） |
| data | 是 | 模板数据 |

**注：url和miniprogram都是非必填字段，若都不传则模板无跳转；若都传，会优先跳转至小程序。开发者可根据实际需要选择其中一种跳转方式即可。当用户的微信客户端版本不支持跳小程序时，将会跳转至url。**

**返回码说明**

在调用模板消息接口后，会返回JSON数据包。正常时的返回JSON数据包示例：

    {

           "errcode":0,

           "errmsg":"ok",

           "msgid":200228332

       }

**使用效果**



事件推送

在模版消息发送任务完成后，微信服务器会将是否送达成功作为通知，发送到开发者中心中填写的服务器配置地址中。

1、送达成功时，推送的XML如下：

           <xml>

           <ToUserName><![CDATA[gh\_7f083739789a]]></ToUserName>

           <FromUserName><![CDATA[oia2TjuEGTNoeX76QEjQNrcURxG8]]></FromUserName>

           <CreateTime>1395658920</CreateTime>

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

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

           <MsgID>200163836</MsgID>

           <Status><![CDATA[success]]></Status>

           </xml>

参数说明

|  |  |
| --- | --- |
| 参数 | 说明 |
| ToUserName | 公众号微信号 |
| FromUserName | 接收模板消息的用户的openid |
| CreateTime | 创建时间 |
| MsgType | 消息类型是事件 |
| Event | 事件为模板消息发送结束 |
| MsgID | 消息id |
| Status | 发送状态为成功 |

2、送达由于用户拒收（用户设置拒绝接收公众号消息）而失败时，推送的XML如下：

           <xml>

           <ToUserName><![CDATA[gh\_7f083739789a]]></ToUserName>

           <FromUserName><![CDATA[oia2TjuEGTNoeX76QEjQNrcURxG8]]></FromUserName>

           <CreateTime>1395658984</CreateTime>

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

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

           <MsgID>200163840</MsgID>

           <Status><![CDATA[failed:user block]]></Status>

           </xml>

参数说明

|  |  |
| --- | --- |
| 参数 | 说明 |
| ToUserName | 公众号微信号 |
| FromUserName | 接收模板消息的用户的openid |
| CreateTime | 创建时间 |
| MsgType | 消息类型是事件 |
| Event | 事件为模板消息发送结束 |
| MsgID | 消息id |
| Status | 发送状态为用户拒绝接收 |

3、送达由于其他原因失败时，推送的XML如下：

           <xml>

           <ToUserName><![CDATA[gh\_7f083739789a]]></ToUserName>

           <FromUserName><![CDATA[oia2TjuEGTNoeX76QEjQNrcURxG8]]></FromUserName>

           <CreateTime>1395658984</CreateTime>

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

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

           <MsgID>200163840</MsgID>

           <Status><![CDATA[failed: system failed]]></Status>

           </xml>

参数说明

|  |  |
| --- | --- |
| 参数 | 说明 |
| ToUserName | 公众号微信号 |
| FromUserName | 接收模板消息的用户的openid |
| CreateTime | 创建时间 |
| MsgType | 消息类型是事件 |
| Event | 事件为模板消息发送结束 |
| MsgID | 消息id |
| Status | 发送状态为发送失败（非用户拒绝） |

## 模板消息运营规范

模板消息用来帮助公众号进行业务通知，是开发者在模板内容中设定参数（参数必须以“{ {”开头，且以“.DATA} }”结尾），并在调用时为这些参数赋值并发送的消息。模板消息的定位是用户触发后的通知消息，不允许在用户没做任何操作或未经用户同意接收的前提下，主动下发消息给用户。目前在特殊情况下允许主动下发的消息只有故障类和灾害警示警告类通知，除此之外都要经过用户同意或用户有触发行为才能下发模板消息。公众号只能在模板库中按照自己的行业来选择模板。如果模板库中暂时没有你想要的模板，则请你仔细阅读以下内容后，在满足要求的情况下，可以为你所在的行业贡献新模板，帮助充实模板库。

一 目前允许发的模板消息

**允许发的模板消息必须是用户接受过帐号主体提供过服务的，严禁用户未接受服务而向其推送模板消息。发送模板消息的前提是内容不涉及广告营销骚扰用户，一经发现内容涉及营销骚扰将严厉处罚。**

1、服务即时通知类消息模板

此类模板消息具有即时性，在用户触发某个事件活动后，即时推送一条模板消息给用户，并告知用户相应内容。此类模板消息举例如下：

  资料变更类通知、政务服务即时类通知、物品（包含虚拟类）收取类通知、消费交易类通知、签到类通知、状态类通知、登陆提醒类通知等一些用户触发后的即时通知。

2、服务后未即时通知类消息模板

此类模板消息具有延时性，在用户触发某个事件或活动后，可能无法即时给用户推送结果，需要一定时间处理后才能回复用户，并告知相应结果的内容。此类模板消息的关键在于频率控制上，我们对消息推送的频率会有严格监控，一经发现频率过高可能骚扰到用户，或收到用户投诉，将会严厉惩处。此类模板消息举例如下：

2.1每月推送频率固定的模板消息

  此类消息主动下发前必须要经过用户同意过，否则不允许发。举例如下：

  月账单类通知消息，如话费通知，水电费通知，物业费通知等每月固定推送给用户的消息通知。

2.2推送频率不固定的模板消息

  除故障类、灾害警示警告类允许特殊情况主动下发，其它类型模板消息主动下发前必须要用户同意过，否则不允许发。举例如下：

  故障类通知、警示警告类通知、到期提醒类通知、缴费提醒类通知、会议提醒类通知、赛事提醒类通知、发车提醒类通知、航班延误提醒类通知、待办任务提醒类通知等对用户很重要的一些消息通知。

**注：此类通知必须是从用户角度来看非常重要，对用户有益的信息，活动公告通知或无关紧要的信息不允许发。特别强调到期提醒类要注意频率和使用场景；待办任务提醒类的内容要注意不能营销骚扰。**

2.3一般延时性通知

  此类模板消息举例如下：

  审核结果类通知、退款结果类通知、投标结果类通知、订单受理结果类通知、报名结果类通知、反馈类通知等一些具有延时性的对用户很重要的消息通知。

附目前允许发的模板示例下载：[点击下载](https://wximg.gtimg.com/shake_tv/mpwiki/template_allow_sample.zip)

二 目前不允许发的模板消息

**1、模板内容与服务场景（含标题、关键词）不一致的模板**

**此类模板消息的内容与服务场景不一致，属于滥用模板消息，不允许发，一经发现将会严厉处罚。**

2、涉嫌广告营销类消息模板

此类模板消息具有广告营销性质，有诱导用户消费的意图，一经发现将会严厉惩处，轻则封接口处理，重则封号处理。此类模板消息举例如下：

  消费优惠类通知、购物返利类通知、商品降价类通知、商品更新提醒类通知等一些涉及消费的营销类通知。

3、发送频率过高，有骚扰用户倾向类消息模板

此类模板消息具有骚扰用户，让用户反感的性质，我们会对该类模板消息监管，一经发现频率过高骚扰到用户，或收到用户投诉，将会严厉惩处，轻则封接口处理，重则封号处理。此类模板消息举例如下：

  频率过高的到期提醒类通知、频率过高的缴费提醒类通知、频率过高的留言提醒类通知、订阅提醒类通知等一些频率过高会骚扰到用户的消息通知。

4、涉及红包、卡券、优惠券、代金券、会员卡类消息模板

由于此类模板消息容易用做营销，今后将不允许发，此类模板的申请也不会通过。目前库内存在的模板暂时不做处理，会有后续处理规范发布。请接入微信卡券，实现券、卡相关的功能。

**注：目前模板库中若存在模板与规范内容相违背，请以此规范为准，以后不符合规范的模板将严格审核不能入库。若仍在使用库中不符合规范的模板，请运营者尽快取消使用，删除模板，我们会清理模板库内不合规范的模板，以后将不允许不符合规范的模板出现。**

附目前不允许发的模板示例下载：[点击下载](https://wximg.gtimg.com/shake_tv/mpwiki/template_not_allow_sample.zip)

三 模板审核标准

 1、符合上述允许发的模板消息里的各项要求。

 2、格式正确，模板标准格式（三段式模板）

  { {first.DATA} }

  关键词1：{ {keyword1.DATA} }

  关键词2：{ {keyword2.DATA} }

  关键词3：{ {keyword3.DATA} }

  { {remark.DATA} }

 3、内容：

①标题、关键词带有品牌或公司名等没有行业通用性的内容，不能通过

包括但不限于：某某公司服务通知、某某商品购买通知

②标题不能带标点或其它特殊符号，且必须以“通知”或“提醒”结尾，否则不能通过

包括不限于：领取红包！、天气提醒！

③易被用作群发，标题或关键词不能简要说明具体服务行为或使用场景的模板不能通过

包括但不限于：管理员类通知、公告类通知、系统通知

④易对用户骚扰的，发送频率较高的模板不能通过

包括但不限于 ：更新类通知、回复类通知、互动类通知

⑤推广宣传、营销类的模板不能通过

包括但不限于 ：红包类通知、优惠券类通知、活动类通知、积分类通知、奖励类通知

⑥模板库里已存在类似的模板不通过

四 填写规则

模板消息填写时，必须严格遵守以下规则，否则也不能审核通过：

 1、模板内容长度不能超过200个字符，且必须有至少10个固定文字或标点

 2、模版内容中，参数是可以在发送时赋值的，参数必须以“{ {”开头，以“.DATA} }”结尾）

 3、内容示例是将模板内容中的参数分别举例后的模板内容的副本，必须严格填写，用以帮助审核人员详细理解模板的用途

 4、参数中可以通过增加“\n”，来实现换行，所以，一般建议将参数紧贴上一行，在需要时用“\n”自行换行。通过该技巧，可以让模板消息的拓展能力变得非常强。

 5、模版内容中，第一句话需要为礼貌性、称谓性的用语，这句话统一用{ {first.DATA} }参数来概括，以免此后称谓习惯等修改后，需修改模版。

 6、模版内容的中部必须为2到5个“关键词名称：关键词内容参数”（中文冒号）的组合排列，这样显示在微信客户端中体验最佳。

 7、模版内容中，末尾必须有{ {remark.DATA} }参数，该参数的作用是，用户可以自行添加多行需要的内容。例如，模版中有姓名、时间这2个关键词，但你需要姓名、时间、地点3个，则可以将地点这个关键词的内容放入remark参数中（需要时可通过\n来换行）。这样就达到了用remark参数来扩充关键词的作用，极大增强了模版的通用性。

 8、直接相连在一起的参数，应简化为一个，避免模板内容过于复杂，因为赋值到两个参数中的内容可以简化地赋值到一个参数中。

 9、为了保持行业通用性，模板标题和模板内容中，不允许带有品牌等关键词，以免影响通用性。

五 处罚规则

**违规的判定原则**

1.发模板的行为：

①模板消息不能主动下发给没有接受过服务的接收者（故障报警、灾害报警和不涉及营销推广的通知除外）

例：某用户仅仅是关注公众号，没有和公众号及其所属主体有任何交互行为，却无故收到该公众号下发的模板消息，属于违规行为

②模板消息的发送频率不能太高骚扰接收者

例：某用户点击公众号的自定义菜单一次或其它触发操作，连续收到3条或更多重复模板消息，属于违规行为

2.模板的内容：

①模板消息内容不能做营销、推广、诱导分享及诱导下载APP

例：某用户购买某商品后，公众号下发模板推销其它商品，与用户此次接受的服务无关，属于违规

②模板内容与模板标题或关键词无关联

例：标题是刷卡成功通知，模板内容却是推销商品或活动通知，属于违规

③模板内容是营销性质的群发活动公告通知

例：标题小区物业通知，模板内容却是群发活动的营销信息

综合上述两项原则后判断，主动下发、内容涉及恶意营销、频率过高恶意骚扰、以及模板参数内容乱填写将被封接口处罚，首次封7天，第二次30天，第三次永久封禁。违规处罚将通过微信公众平台站内信告知运营者。

获取公众号的自动回复规则

开发者可以通过该接口，获取公众号当前使用的自动回复规则，包括关注后自动回复、消息自动回复（60分钟内触发一次）、关键词自动回复。

请注意：

1、第三方平台开发者可以通过本接口，在旗下公众号将业务授权给你后，立即通过本接口检测公众号的自动回复配置，并通过接口再次给公众号设置好自动回复规则，以提升公众号运营者的业务体验。  
2、本接口仅能获取公众号在公众平台官网的自动回复功能中设置的自动回复规则，若公众号自行开发实现自动回复，或通过第三方平台开发者来实现，则无法获取。  
3、认证/未认证的服务号/订阅号，以及接口测试号，均拥有该接口权限。  
4、从第三方平台的公众号登录授权机制上来说，该接口从属于消息与菜单权限集。  
5、本接口中返回的图片/语音/视频为临时素材（临时素材每次获取都不同，3天内有效，通过素材管理-获取临时素材接口来获取这些素材），本接口返回的图文消息为永久素材素材（通过素材管理-获取永久素材接口来获取这些素材）。

**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/get\_current\_autoreply\_info?access\_token=ACCESS\_TOKEN

**返回结果说明**

返回的JSON格式样例：（注意，格式化前的json使用‘\’作为转义符）

{   
  "is\_add\_friend\_reply\_open": 1,   
  "is\_autoreply\_open": 1,   
  "add\_friend\_autoreply\_info": {   
      "type": "text",   
      "content": "Thanks for your attention!"  
  },   
  "message\_default\_autoreply\_info": {   
      "type": "text",   
      "content": "Hello, this is autoreply!"  
  },   
  "keyword\_autoreply\_info": {   
      "list": [   
          {   
              "rule\_name": "autoreply-news",   
              "create\_time": 1423028166,   
              "reply\_mode": "reply\_all",   
              "keyword\_list\_info": [   
                  {   
                      "type": "text",   
                      "match\_mode": "contain",   
                      "content": "news测试"//此处content即为关键词内容  
                  }  
              ],   
              "reply\_list\_info": [   
                  {   
                      "type": "news",   
                      "news\_info": {   
                          "list": [   
                              {   
                                  "title": "it's news",   
                                  "author": "jim",   
                                  "digest": "it's digest",   
                                  "show\_cover": 1,  "cover\_url": "http://mmbiz.qpic.cn/mmbiz/GE7et87vE9vicuCibqXsX9GPPLuEtBfXfKbE8sWdt2DDcL0dMfQWJWTVn1N8DxI0gcRmrtqBOuwQH

  euPKmFLK0ZQ/0",   
                                  "content\_url": "http://mp.weixin.qq.com/s?\_\_biz=MjM5ODUwNTM3Ng==&mid=203929886&idx=1&sn=628f964cf0c6d84c026881b6959aea8b#rd",   
                                  "source\_url": "http://www.url.com"  
                              }  
                          ]  
                      }  
                  },   
                  {   
                      "type": "news",  
                      "content":"KQb\_w\_Tiz-nSdVLoTV35Psmty8hGBulGhEdbb9SKs-o",   
                      "news\_info": {   
                          "list": [   
                              {   
                                  "title": "MULTI\_NEWS",   
                                  "author": "JIMZHENG",   
                                  "digest": "text",   
                                  "show\_cover": 0,   
                                  "cover\_url": "http://mmbiz.qpic.cn/mmbiz/GE7et87vE9vicuCibqXsX9GPPLuEtBfXfK0HKuBIa1A1cypS0uY1wickv70iaY1gf3I1DTszuJoS3lAVLv

hTcm9sDA/0",   
                                  "content\_url": "http://mp.weixin.qq.com/s?\_\_biz=MjM5ODUwNTM3Ng==&mid=204013432&idx=1&sn=80ce6d9abcb832237bf86c87e50fda15#rd",   
                                  "source\_url": ""  
                              },  
                              {   
                                  "title": "MULTI\_NEWS4",   
                                  "author": "JIMZHENG",   
                                  "digest": "MULTI\_NEWSMULTI\_NEWSMULTI\_NEWSMULTI\_NEWSMULTI\_NEWSMULT",   
                                  "show\_cover": 1,   
"cover\_url": "http://mmbiz.qpic.cn/mmbiz/GE7et87vE9vicuCibqXsX9GPPLuEtBfXfKbE8sWdt2DDcL0dMfQWJWTVn1N8DxI0gcRmrtqBOuwQ

HeuPKmFLK0ZQ/0",   
                                  "content\_url": "http://mp.weixin.qq.com/s?\_\_biz=MjM5ODUwNTM3Ng==&mid=204013432&idx=5&sn=b4ef73a915e7c2265e437096582774af#rd",   
                                  "source\_url": ""  
                              }  
                          ]  
                      }  
                  }  
              ]  
          },   
          {   
              "rule\_name": "autoreply-voice",   
              "create\_time": 1423027971,   
              "reply\_mode": "random\_one",   
              "keyword\_list\_info": [   
                  {   
                      "type": "text",   
                      "match\_mode": "contain",   
                      "content": "voice测试"  
                  }  
              ],   
              "reply\_list\_info": [   
                  {   
                      "type": "voice",   
                      "content": "NESsxgHEvAcg3egJTtYj4uG1PTL6iPhratdWKDLAXYErhN6oEEfMdVyblWtBY5vp"  
                  }  
              ]  
          },   
          {   
              "rule\_name": "autoreply-text",   
              "create\_time": 1423027926,   
              "reply\_mode": "random\_one",   
              "keyword\_list\_info": [   
                  {   
                      "type": "text",   
                      "match\_mode": "contain",   
                      "content": "text测试"  
                  }  
              ],   
              "reply\_list\_info": [   
                  {   
                      "type": "text",   
                      "content": "hello!text!"  
                  }  
              ]  
          },   
          {   
              "rule\_name": "autoreply-video",   
              "create\_time": 1423027801,   
              "reply\_mode": "random\_one",   
              "keyword\_list\_info": [   
                  {   
                      "type": "text",   
                      "match\_mode": "equal",   
                      "content": "video测试"  
                  }  
              ],   
              "reply\_list\_info": [   
                  {   
                 "type": "video",   
"content": "http://61.182.133.153/vweixinp.tc.qq.com/1007\_114bcede9a2244eeb5ab7f76d951df5f.f10.mp4?vkey=7183E5C952B16C3AB1991BA8138673DE1037CB82A29801A504B64A77F691BF9DF7AD054A9B7FE683&sha=0&save=1"  
                  }  
              ]  
          }  
      ]  
  }  
}

**参数说明**

|  |  |
| --- | --- |
| 参数 | 说明 |
| is\_add\_friend\_reply\_open | 关注后自动回复是否开启，0代表未开启，1代表开启 |
| is\_autoreply\_open | 消息自动回复是否开启，0代表未开启，1代表开启 |
| add\_friend\_autoreply\_info | 关注后自动回复的信息 |
| type | 自动回复的类型。关注后自动回复和消息自动回复的类型仅支持文本（text）、图片（img）、语音（voice）、视频（video），关键词自动回复则还多了图文消息（news） |
| content | 对于文本类型，content是文本内容，对于图文、图片、语音、视频类型，content是mediaID |
| message\_default\_autoreply\_info | 消息自动回复的信息 |
| keyword\_autoreply\_info | 关键词自动回复的信息 |
| rule\_name | 规则名称 |
| create\_time | 创建时间 |
| reply\_mode | 回复模式，reply\_all代表全部回复，random\_one代表随机回复其中一条 |
| keyword\_list\_info | 匹配的关键词列表 |
| match\_mode | 匹配模式，contain代表消息中含有该关键词即可，equal表示消息内容必须和关键词严格相同 |
| news\_info | 图文消息的信息 |
| title | 图文消息的标题 |
| digest | 摘要 |
| author | 作者 |
| show\_cover | 是否显示封面，0为不显示，1为显示 |
| cover\_url | 封面图片的URL |
| content\_url | 正文的URL |
| source\_url | 原文的URL，若置空则无查看原文入口 |

<https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1433751299-->微信网页开发