

微信公众平台开发

1. 关于公众号

网址: <https://mp.weixin.qq.com>

帐号分类



服务号

给企业和组织提供更强大的业务服务与用户管理能力,帮助企业快速实现全新的公众号服务平台。



订阅号

为媒体和个人提供一种新的信息传播方式,构建与读者之间更好的沟通与管理模式。



企业号

为企业或组织提供移动应用入口,帮助企业建立与员工、上下游供应链及企业应用间的连接。

2. 申请测试号: <https://mp.weixin.qq.com/wiki>



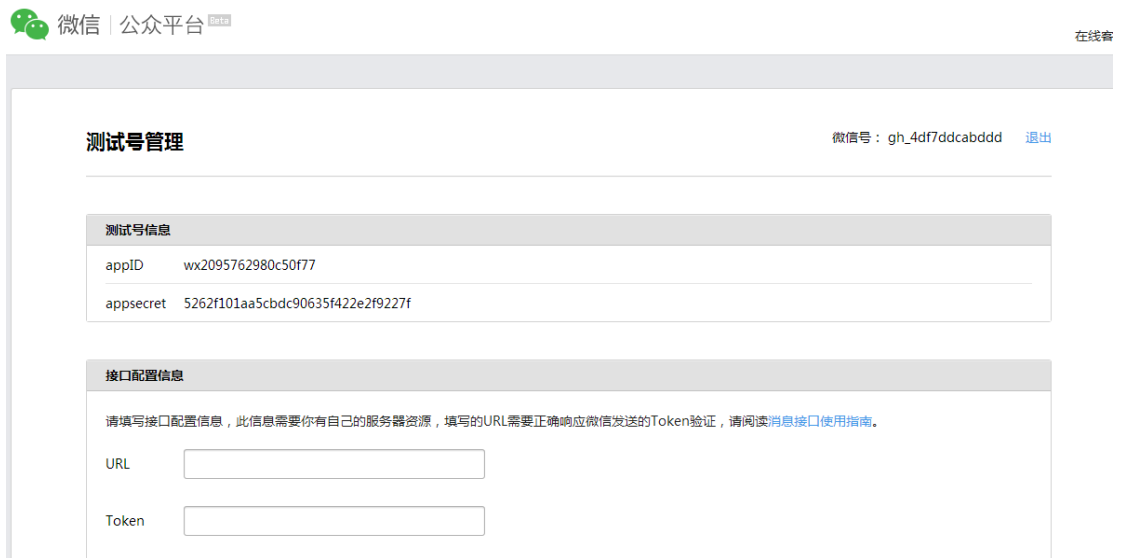
微信 | 公众平台开发者文档

<ul style="list-style-type: none">▶ 开发者必读▶ 开发者工具<ul style="list-style-type: none">接口测试号申请接口在线调试常见问题▶ 获取接口调用凭据▶ 接收消息▶ 发送消息▶ 消息加解密▶ 素材管理▶ 用户管理▶ 自定义菜单管理▶ 帐号管理▶ 数据统计接口▶ 微信JS-SDK▶ 微信小店接口▶ 微信卡券接口	<p>首页</p> <h3>微信公众平台开发概述</h3> <p>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。</p> <p>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。</p> <p>请开发者注意:</p> <ol style="list-style-type: none">1. 微信公众平台开发是指为微信公众平台进行业务开发,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,以及报管排查指引(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。 <p>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:</p> <ol style="list-style-type: none">1. 公众号消息会话
--	--

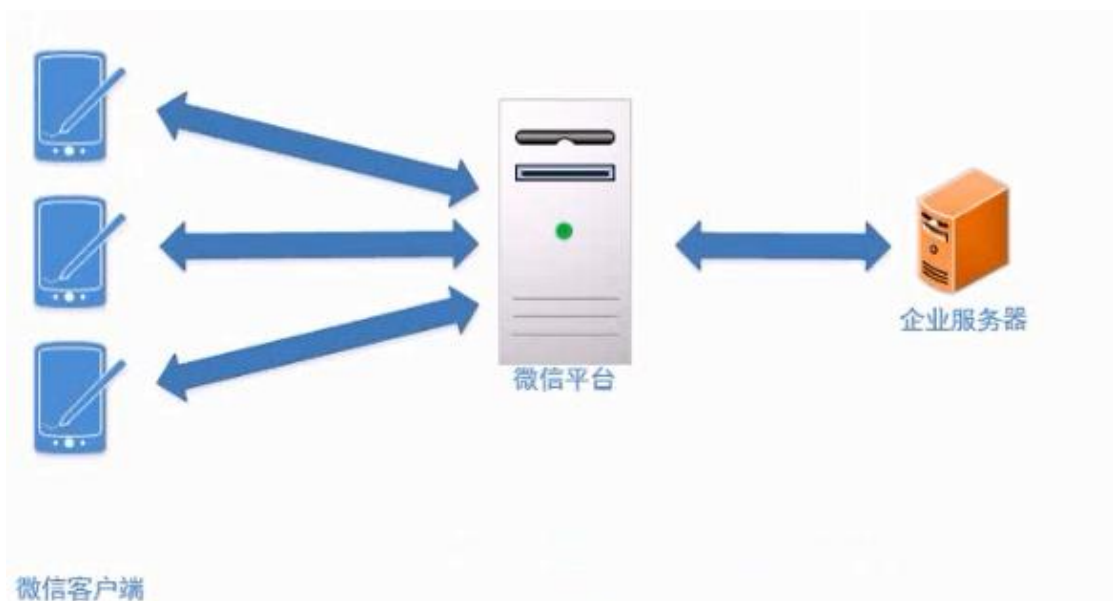
申请页面



通过扫描二维码后



微信工作流程



3. Access Token

access_token 是公众号的全局唯一票据，公众号调用各接口时都需使用 access_token。
如何获取 Access token?

接口调用请求说明

```
http请求方式: GET  
https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET
```

参数说明

参数	是否必须	说明
grant_type	是	获取access_token填写client_credential
appid	是	第三方用户唯一凭证
secret	是	第三方用户唯一凭证密钥，即appsecret

方法一:

```
https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=  
wx2095762980c50f77&secret= 5262f101aa5cbdc90635f422e2f9227f
```

在浏览器返回结果

```
{"access_token": "oS6HdSgIzdHfQd51PzA1BVnhrPhPcaWOSk71V0auBUzpvu-8HyP6  
MiTsCs53naRvkvVz6yYu4c805KCV0fWBUE1D18hDpNmcCi_dpfsbBUk", "expires_in"  
: 7200}
```

方法二：通过网页调试工具调试该接口获取

微信公众平台接口调试工具

此工具旨在帮助开发者检测调用【微信公众平台开发者API】时发送的请求参数是否正确，提交相关信息后可获得服务器的验证结果

使用说明：

- (1) 选择合适的接口。
- (2) 系统会生成该接口的参数表，您可以直接在文本框内填入对应的参数值。（红色星号表示该字段必填）
- (3) 点击检查问题按钮，即可得到相应的调试信息。

一、接口类型：

二、接口列表： 方法：GET

三、参数列表：

* grant_type：
 获取access_token填写client_credential
 校验通过

* appid：
 填写appid
 校验通过

* secret：
 填写appsecret
 校验通过

基础支持: 获取access_token接口 /token ✕

请求地址：https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=wx2095762980c50f77&secret=5262f101aa5cbdc90635f422e2f922f

返回结果：200 OK

```

Connection: keep-alive
Date: Wed, 09 Sep 2015 12:46:12 GMT
Server: nginx/1.8.0
Content-Type: application/json; encoding=utf-8
Content-Length: 144
{
  "access_token": "HUrJuy3fmXfpzYG2QGlL2K8mNNSHmf5-P4ITbLsbAN_tZ6gEx3DZDauO8I155vc2zk-W86AHz7hYKsf1s_ZpDdDcPGs_EF25HQ1PSSIA",
  "expires_in": 7200
}
    
```

提示：**Request successful**

一、接口类型：

二、接口列表： 方法：GET

三、参数列表：

* grant_type：
获取access_token填写client_credential
校验通过

* appid：
填写appid
校验通过

* secret：
填写appsecret
校验通过

基础支持: 获取access_token接口 /token

请求地址：https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=wx2095762980c50f77&secret=5262f101aa5cbdc90635f422e2f9227f

返回结果: 200 OK

Connection: keep-alive
Date: Wed, 09 Sep 2015 12:46:12 GMT
Server: nginx/1.8.0
Content-Type: application/json; encoding=utf-8
Content-Length: 144

```
{
  "access_token": "HUrJuy3fmXfpzYG2QGhL2K8mNNSHmf5-P4ITbLsbAN_tZ6gEx3DZDauO8I155vc2zk-W8s6AHtz7hYKsf1s_ZpDdDcPGs_EF25HQ1PS5IA",
  "expires_in": 7200
}
```

提示: Request successful

重点来学习用 php 来获取 access token
<?php

```
class WeChat{
    private $_appid;
    private $_appsecret;
    private $_token;

    public function __construct($_appid,$_appsecret,$_token){
        $this->_appid = $_appid;
```

```

        $this->_appsecret = $_appsecret;
        $this->token = $_token;
    }

    public function _request($curl,$https= true,$method ='GET',$data = null){
        $ch = curl_init();//初始化
        curl_setopt($ch, CURLOPT_URL, $curl);
        curl_setopt($ch, CURLOPT_HEADER, false);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        if($https){
            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
            curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, true);
        }
        if($method == 'POST'){
            curl_setopt($ch, CURLOPT_POST, true);
            curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
        }
        $content = curl_exec($ch);
        curl_close($ch);
        return $content;
    }

    public function _getAccessToken(){
        $curl =
'https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid='.$this->_appid.'&
secret='.$this->_appsecret;
        return $this->_request($curl);
    }
}

$wechat = new WeChat ('wx2095762980c50f77','5262f101aa5cbdc90635f422e2f9227f','');
//echo $wechat->_request('https://www.baidu.com');
echo $wechat->_getAccessToken();

?>

<?php

class WeChat{
    private $_appid;
    private $_appsecret;
    private $_token;

    public function __construct($_appid,$_appsecret,$_token){
        $this->_appid = $_appid;

```

```

        $this->_appsecret = $_appsecret;
        $this->token = $_token;
    }

    public function _request($curl,$https= true,$method ='GET',$data = null){
        $ch = curl_init();//初始化
        curl_setopt($ch, CURLOPT_URL, $curl);
        curl_setopt($ch, CURLOPT_HEADER, false);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        if($https){
            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
            curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, true);
        }
        if($method == 'POST'){
            curl_setopt($ch, CURLOPT_POST, true);
            curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
        }
        $content = curl_exec($ch);
        curl_close($ch);
        return $content;
    }

    public function _getAccessToken(){
        $file = './accesstoken';
        if(file_exists($file)){
            $content = file_get_contents($file);
            $content = json_decode($content);
            if(time() - filemtime($file) < $content->expires_in)
                return $content->access_token;
        }

        $curl =
'https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid='.$this->_appid.'&
secret='.$this->_appsecret;
        $content = $this->_request($curl);
        file_put_contents($file,$content);
        $content = json_decode($content);
        return $content->access_token;
    }
}

$wechat = new WeChat ('wx2095762980c50f77','5262f101aa5cbdc90635f422e2f9227f','');
//echo $wechat->_request('https://www.baidu.com');
echo $wechat->_getAccessToken();

```

?>

通过 php 程序来获取 Access Token

使用 curl 函数库

生成二维码（QRCode：快速响应码）

The screenshot shows the WeChat developer tools interface. On the left is a navigation menu with categories like '发送消息', '消息加解密', '素材管理', '用户管理', '自定义菜单管理', '帐号管理', '数据统计接口', '微信JS-SDK', '微信小店接口', and '微信卡券接口'. The '帐号管理' category is expanded, and '生成带参数的二维码' is highlighted with a red box. On the right, there is a text box titled '目前有两种类型的二维码：' containing two numbered points: 1. 临时二维码, 是有过期时间的, 最长可以设置为在二维码生成后的7天(即604800秒)后过期, 但能够生成较多数量. 临时二维码主要用于帐号绑定等不要求二维码永久保存的业务场景. 2. 永久二维码, 是无过期时间的, 但数量较少(目前为最多10万个). 永久二维码主要用于适用于帐号绑定、用户来源统计等场景. Below this is another text box titled '用户扫描带场景值二维码时, 可能推送以下两种事件：' with two numbered points: 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, 在用户扫描后会进入会话, 微信也会将带场景值扫描事件推送给开发者. Below that is a text box titled '获取带参数的二维码的过程包括两步, 首先创建二维码ticket, 然后凭借ticket到指定URL换取二维码.' followed by a section header '创建二维码ticket' and a paragraph: '每次创建二维码ticket需要提供一个开发者自行设定的参数(scene_id), 分别介绍临时二维码和永久二维码的创建二维码ticket过程.' Below that is a section header '临时二维码请求说明' and a text box containing: 'http请求方式: POST', 'URL: https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token=TOKEN', 'POST数据格式: json', and 'POST数据例子: {"expire_seconds": 604800, "action_name": "QR_SCENE", "action_info": {"scene": {"scene_id": 123}}}'. At the bottom, there is a section header '永久二维码请求说明' and a text box containing: 'http请求方式: POST', 'URL: https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token=TOKEN', 'POST数据格式: json', 'POST数据例子: {"action_name": "QR_LIMIT_SCENE", "action_info": {"scene": {"scene_id": 123}}}', and '或者也可以使用以下POST数据创建字符串形式的二维码参数: {"action_name": "QR_LIMIT_STR_SCENE", "action_info": {"scene": {"scene_str": "123}}}'.

第一步生成 ticket

创建二维码ticket

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

临时二维码请求说明

```
http请求方式: POST
URL: https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token=TOKEN
POST数据格式: json
POST数据例子: {"expire_seconds": 604800, "action_name": "QR_SCENE", "action_info": {"scene": {"scene_id": 123}}}
```

永久二维码请求说明

```
http请求方式: POST
URL: https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token=TOKEN
POST数据格式: json
POST数据例子: {"action_name": "QR_LIMIT_SCENE", "action_info": {"scene": {"scene_id": 123}}}
或者也可以使用以下POST数据创建字符串形式的二维码参数:
{"action_name": "QR_LIMIT_STR_SCENE", "action_info": {"scene": {"scene_str": "123}}}
```

第二步用 ticket 换取二维码

通过ticket换取二维码

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

请求说明

```
HTTP GET请求（请使用https协议）
https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=TICKET
提醒：TICKET记得进行urlencode
```

返回说明

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

```
HTTP头（示例）如下：
Accept-Ranges:bytes
Cache-control:max-age=604800
Connection:keep-alive
Content-Length:28026
Content-Type:image/jpeg
Date:Wed, 16 Oct 2013 06:37:10 GMT
Expires:Wed, 23 Oct 2013 14:37:10 +0800
Server:nginx/1.4.1
```

错误情况下（如ticket非法）返回HTTP错误码404。

```
public function _getTicket($sceneid, $type='temp', $expire_seconds=604800){
    //判断类型是否是临时
    if($type=='temp'){
        $data = '{"expire_seconds": %s, "action_name": "QR_SCENE", "action_info": {"scene": {"scene_id": %s}}}';
        $data = sprintf($data, $expire_seconds, $sceneid);
    }else{
        $data = '{"action_name": "QR_LIMIT_SCENE", "action_info": {"scene": {"scene_id": %s}}}';
        $data = sprintf($data, $sceneid);
    }
    //发送
    $curl = 'https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token='.$this->_getAccessToken();
    //发送请求
    $content = $this->_request($curl, true, 'POST', $data);
    $content = json_decode($content);
    //得到票据
    return $content->ticket;
}
//通过ticket换取二维码
public function _getQRCode($sceneid, $type = 'temp', $expire_seconds=604800){
    $ticket = $this->_getTicket($sceneid, $type, $expire_seconds); //获得参数
    $content = $this->_request('https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket='.$urlencode($ticket));
    return $content;
}
```

```
header('Content-type:image/jpeg');
echo $wechat->_getQRCode(30);
```

5.接入校验

体验接口权限表				
类目	功能	接口	每日调用上限/次	操作
	基础支持	获取access_token	2000	
		获取微信服务器IP地址	无上限	
	接收消息	验证消息真实性	无上限	
		接收普通消息	无上限	
		接收事件推送	无上限	
		接收语音识别结果	无上限	开启
		自动回复	无上限	

找到下载位置进行下载

- ▶ 微信JS-SDK
- ▶ 微信小店接口
- ▶ 微信卡券接口
- ▶ 微信门店接口
- ▶ 微信智能接口
- ▶ 设备功能介绍
- ▶ 多客服功能
- ▶ 摇一摇周边

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

加密/校验流程如下：

1. 将token、timestamp、nonce三个参数进行字典序排序
2. 将三个参数字符串拼接成一个字符串进行sha1加密
3. 开发者获得加密后的字符串可与signature对比，标识该请求来源于微信

检验signature的PHP示例代码：

```
private function checkSignature()
{
    $signature = $_GET["signature"];
    $timestamp = $_GET["timestamp"];
    $nonce = $_GET["nonce"];
    $token = TOKEN;
    $tmpArr = array($token, $timestamp, $nonce);
    sort($tmpArr, SORT_STRING);
    $tmpStr = implode( $tmpArr );
    $tmpStr = sha1( $tmpStr );
    if( $tmpStr == $signature ){
        return true;
    }else{
        return false;
    }
}
```

PHP示例代码下载：[下载](#)

使用下载的样本文件，修改其中的 TOKEN 常量,这个校验只是第一次接入微信平台使用。

```
<?php
/**
 * wechat php test
 * update time: 20141008
 */

//define your token
define("TOKEN", "weixinopen");
$wechatObj = new WeChat();
//$wechatObj->valid();//第一次接入微信平台，验证使用，以后不需要再验证
$wechatObj->responseMsg();

class WeChat
```

以后修改使用该文件中的 responseMsg 方法就可以完成和微信用户的交互。

```

public function responseMsg()
{
    //get post data, May be due to the different environments
    $postStr = $GLOBALS["HTTP_RAW_POST_DATA"];

    //extract post data
    if (!empty($postStr)){

        $postObj = simplexml_load_string($postStr, 'SimpleXMLElement', LIBXML_NOCDATA);
        $fromUsername = $postObj->FromUserName;
        $toUsername = $postObj->ToUserName;
        $keyword = trim($postObj->Content);
        $time = time();
        $textTpl = "<xml>
                <ToUserName><![CDATA[%s]]></ToUserName>
                <FromUserName><![CDATA[%s]]></FromUserName>
                <CreateTime>%s</CreateTime>
                <MsgType><![CDATA[%s]]></MsgType>
            ";
    }
}

```

我们的服务器通过微信平台发送的 xml 文档中的 MsgType 判断微信用户发送的消息类型。

创建菜单

菜单的形式

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。

创建菜单接口：

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

创建菜单的格式：

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

程序实现：

```
public function _createMenu($menu) {
    $curl = 'https://api.weixin.qq.com/cgi-bin/menu/create?access_token=' . $this->_getAccessToken();
    $content = json_decode($this->_request($curl, true, 'POST', $menu));
    if($content->errcode == 0)
        echo "菜单成功创建";
}
```

```
class.php createmenu.php
<?php
define('APPID','wx2095762980c50f77');
define('APPSECRET','5262f101aa5cbdc90635f422e2f9227f');
define('TOKEN','zclweixin');

require './wechat.class.php';
header("content-type:text/html;charset=utf-8");
//实例化
$wechat = new WeChat(APPID,APPSECRET,TOKEN);
$menu = '
    {
      "button": [
        {
          "type": "click",
          "name": "新闻",
          "key": "news"
        },
        {
          "name": "休闲",
          "sub_button": [
            {
              "type": "view",
              "name": "打鬼砸",
              "url": "http://flash1.7k7k.com/h5/2015/dgz/index.html"
            },
            {
              "type": "view",
              "name": "视频",
              "url": "http://v.qq.com/"
            },
            {
              "type": "click",
              "name": "赞一下我们",
              "key": "GOOD"
            }
          ]
        }
      ]
    }
';
$wechat->_createMenu($menu);
```

发送图文信息到微信用户
回复图文消息的格式:

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>
```