**微哨校园开放平台-轻应用接口手册**

## 身份校验

### 获取凭证

**接口说明:**

1.拥有一个微哨登陆账号，通过登陆微哨PC客户端并点击“登陆微哨开放平台”打开微哨校园开放平台，添加应用后即可获得应用的appid和appsecret

2.通过获取凭证接口获取到access\_token

|  |
| --- |
| **接口（请求方式：get）** |
| **https://校园微哨通讯服务器/cgi-bin/oauth2/access\_token?grant\_type= client\_credentials&appid= APPID&appsecret=APPSECRET;** |

**参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| appid | 是 | 申请微哨应用时分配的Appid（通过登录微哨开放平台获取） |
| appsecret | 是 | 申请微哨应用时分配的AppSecret（通过登录微哨开放平台获取） |
| grant\_type | 是 | 内容必须为client\_credentials |

**协议说明：**

|  |
| --- |
| **协议** |
| **正确的Json返回结果:**  {  "access\_token":"2YotnFZFEjr1zCsicMWpAA",  "expires\_in":3600  }  其中参数如下：  access\_token 获取到的凭证  expires\_time 凭证到期时间  **错误的Json返回示例:**  {"errcode":40013,"errmsg":"invalid appid"} |

**注：**

1、access\_token是第三方访问微哨api资源的票据；

2、access\_token对应于公众号是全局唯一的票据，重复获取将导致上次获取的access\_token失效。

3、由于微哨服务器是配置在校园内部的，因此校园微哨通讯服务器、校园微哨资源服务器会因学校不同而不同，用户可以通过登录微哨开放平台获取。登陆开放平台后依次打开，“进入我的个人中心” “我的应用”“对应应用”“编辑”，显示如下界面：



红色框中即为校园微哨通讯服务器和校园微哨资源服务器的地址。

### URL有效性验证

开发者创建应用时，应提交处理微哨请求的URL。为了确认开发者提交的URL可用，用户提交轻应用通讯URL后，微哨服务器将发送GET请求到填写的URL上，并且带上四个参数：

|  |  |
| --- | --- |
| **参数** | **描述** |
| **signature** | 微哨加密签名，由token、timestamp、 nonce计算得来。 |
| **timestamp** | 时间戳 |
| **nonce** | 随机数 |
| **echostr** | 随机字符串 |

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

加密/校验流程：

1. 将token（轻应用通讯凭证）、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哨服务器

注：

1.微哨服务器与第三方服务器 校验通过后保持长连接 交换数据。若长时间没有数据交换，5分钟后自动断开服务器连接，重复校验过程以再次获得连接；

2. token（轻应用通讯凭证）是在登陆微哨开放平台后，创建应用时开发者自己填写的内容，必须保证代码中的token与开放平台的一致。并且限制为英文字母和阿拉伯数字，长度限制为0-32位。

**此后，每次应用接收用户消息的时候，微哨也都会带上这前三个参数访问开发者设置的URL，开发者依然通过对签名的效验判断此条消息的真实性。效验方式与首次提交验证申请一致（参见1.3）。**

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);

$tmpStr = implode( $tmpArr );

$tmpStr = sha1( $tmpStr );

if( $tmpStr == $signature ){

return true;

}else{

return false;

}

}

### 消息合法性校验

客户端向轻应用发送消息时，微哨服务器将消息发送到填写的URL上，并且带上三个参数：

|  |  |
| --- | --- |
| **参数** | **描述** |
| **signature** | 微哨加密签名，由token、timestamp、 nonce计算得来。 |
| **timestamp** | 时间戳 |
| **nonce** | 随机数 |

开发者通过检验signature对请求进行校验（校验方式参考1.2的加密/校验流程），用于确认请求来自微哨服务器。

## 自定义菜单

简介

开发者获取使用凭证后，可以使用该凭证对公众账号的自定义菜单进行创建、查询和删除等操作。 自定义菜单接口可实现以下类型按钮：

click（点击事件）： 用户点击click类型按钮后，微哨服务器会通过消息接口(event类型)推送点击事件给开发者，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值进行消息回复。

view（访问网页）： 用户点击view类型按钮后，会直接跳转到开发者指定的url中。

### 2.1设置菜单

|  |
| --- |
| **接口（请求方式：input）** |
| [https:// 校园微哨通讯服务器/lightapp/menu/set?appid=appid&access\_token=ACCESS\_TOKEN](https://domain/lightapp/menu/set?appid=appid&access_token=ACCESS_TOKEN) |

|  |  |
| --- | --- |
| **参数** | |
| app\_id | 应用id |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

|  |
| --- |
| **菜单示例** |
| {  "button":[  {  "type":"click",  "name":"今日歌曲",  "key":"V1001\_TODAY\_MUSIC"  },  {  "type":"view",  "name":"微哨官网",  "url":"http://www.weishao.com.cn/"  },  {  "name":"菜单",  "sub\_button":[  {  "type":"click",  "name":"hello word",  "key":"V1001\_HELLO\_WORLD"  },  {  "type":"click",  "name":"赞一下我们",  "key":"V1001\_GOOD"  }  ]  }  ]  } |

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| **button** | 是 | 一级菜单，数目不大于3个 |
| **sub\_button** | 否 | 二级菜单，数目不大于5个 |
| **type** | 是 | button的类型，必须是click或者view，为click时需要定义key值。为view时需要填写url， |
| **name** | 是 | button的标题，一级菜单标题长度不超过8个字符（4个汉字），二级菜单标题长度不超过16个字符（8个汉字）。 |
| **key** | 菜单button的type为click时必须 | 菜单key值，用于消息接口推送，不超过128字节 |
| **url** | 菜单button的type为view时必须 | 网页链接，用户点击菜单可打开链接，不超过256字节，url在pc端会在本地的默认浏览器中打开，在手机端会在微哨内置浏览器中打开。 |

**注：进行菜单设置时，应注意，该应用必须处于调试状态，应用开放后，将不能设置菜单！**

|  |
| --- |
| **返回说明** |
| 正确的Json返回结果:  {"errcode":0,"errmsg":"ok"}  错误的Json返回结果  {"errcode":40018,"errmsg":"invalid button name size"} |

### 2.2查询菜单

|  |
| --- |
| **接口（请求方式：get）** |
| [https://校园微哨通讯服务器/lightapp/menu/view?appid=appid&access\_token= ACCESS\_TOKEN](https://校园微哨通讯服务器/lightapp/menu/view?appid=appid&access_token=%20ACCESS_TOKEN) |

|  |  |
| --- | --- |
| **参数** | |
| appid | 应用编号 |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

|  |
| --- |
| **协议** |
| 返回设置的菜单内容，参考2.1.1 |

### 2.3删除菜单

|  |
| --- |
| **接口（请求方式：get）** |
| [https://校园微哨通讯服务器/lightapp/menu/delete?appid=appid&access\_token= ACCESS\_TOKEN](https://校园微哨通讯服务器/lightapp/menu/delete?appid=appid&access_token=%20ACCESS_TOKEN) |

|  |  |
| --- | --- |
| **参数** | |
| appid | 应用编号 |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

|  |
| --- |
| **协议** |
| 暂无 |

|  |
| --- |
| **返回说明** |
| 删除完成后 正确的Json返回结果:  {"errcode":0,"errmsg":"ok"} |

## . 获取用户信息

|  |
| --- |
| **接口（请求方式：get）** |
| https://校园微哨通讯服务器/api/user/get\_info?access\_token=asydf32r3221&appid=app231&student\_number =4542233 |

|  |  |
| --- | --- |
| **参数** | |
| appid | 应用编号 |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |
| student\_number | 用户学号，学号可以在接收消息时得到（消息中的fromstudentnumber字段）。获取用户信息同时支持使用消息中的fromusername获取。 |

|  |
| --- |
| **返回说明** |
| 调用成功返回用户信息：  {  "retcode": "0",  "data": {  "photo\_credential": "Nvf4luqrXoC.jpg",  "sex": "boy",  "organization": "计算机学院/\*\*\*\*计技本\*班",  "student\_number": "\*\*\*\*\*\*\*\*\*\*",  "name": "\*\*\*\*\*",  "aid": "\*\*\*\*\*@dgut.edu.cn"  },  "errmsg": "ok",  "errcode": "0"  }   |  |  | | --- | --- | | **参数** | **说明** | | **name** | 姓名 | | **aid** | 用户微哨号@域名，如：\*\*\*\*\*@dgut.edu.cn | | **organization** | 组织，如“外语学院/102班” | | **sex** | 性别: boy：男 ，girl：女 | | **photo\_credential** | 头像 | | **student\_number** | 学号 | |

## .收消息

当微哨用户向公众账号发消息时，微哨服务器会将用户消息已XML格式POST 到开发者填写的URL上。各消息类型的推送XML数据包结构如下。

### 4.1 hello 报文消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<createtime>1348831860</createtime>

<msgtype><![CDATA[hello]]></msgtype>

<helloid><![CDATA[hello id]]></helloid>

<msgid>1234567890123456</msgid>

</xml>

相关参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| createtime | 消息创建时间 （整型） |
| msgtype | hello |
| helloid | Hello报文ID |
| msgid | 消息id，64位整型 |

说明：hello报文在用户打开应用时发送，开发者可以感知到用户的打开行为，并作出响应。当用户在同一设备登录时，helloid是唯一的。用户不同或者同一用户登陆设备不同则helloid不同。

### 4.2文本消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<createtime>1348831860</createtime>

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

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

<msgid>1234567890123456</msgid>

</xml>

相关参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| 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 |

### 4.3声音信息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<createtime>1348831860</createtime>

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

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

<msgid>1234567890123456</msgid>

</xml>

相关参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| createtime | 消息创建时间 （整型） |
| msgtype | voice声音文件 |
| voiceurl | 声音文件链接 |
| msgid | 消息id，64位整型 |

### 4.4 图片消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<createtime>1348831860</createtime>

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

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

<msgid>1234567890123456</msgid>

</xml>

相关参数说明：

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| createtime | 消息创建时间 （整型） |
| msgtype | image |
| picurl | 图片链接, |
| msgid | 消息id，64位整型 |

### 4.5地理位置消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<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 | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| createtime | 消息创建时间 （整型） |
| msgtype | location |
| location\_x | 地理位置纬度 |
| 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 |

### 4.6事件推送

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<fromstudentnumber><![CDATA[fromstudentnumber]]></fromstudentnumber>

<createtime>123456789</createtime>

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

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

<eventkey><![CDATA[eventkey]]></eventkey>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 微哨轻应用ID (appid) |
| fromusername | 发送方帐号（一个JID） |
| fromstudentnumber | 学工号/教工号 |
| createtime | 消息创建时间 |
| msgtype | 消息类型，event |
| event | 事件类型，CLICK(自定义菜单点击事件) |
| eventkey | 事件KEY值，与自定义菜单接口中KEY值对应 |

## .消息回复

对于每一个POST请求，开发者在响应包中返回特定xml结构，对该消息进行响应（现支持回复文本、图文、语音、视频、**音乐**）。

微哨服务器在**五秒内**收不到响应会断掉连接。

回复xml结构如下：

#### 4.1回复文本消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<createtime>12345678</createtime>

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

<content><![CDATA[content]]></content>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 接收方帐号（收到的JID） |
| fromusername | 微哨轻应用ID (appid) |
| createtime | 消息创建时间 |
| msgtype | text |
| content | 回复的消息内容,长度不超过2048字节 |

#### 4.2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>

**参数说明：**

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 接收方帐号（收到的JID） |
| fromusername | 微哨轻应用ID (appid) |
| createtime | 消息创建时间 |
| msgtype | news |

|  |  |
| --- | --- |
| articlecount | 图文消息个数，限制为10条以内 |
| articles | 多条图文消息信息，默认第一个item为大图 |
| title | 图文消息标题 |
| description | 图文消息描述 |
| picurl | 图片链接(图片全路径)，如果是上传到微哨服务器的资源，则对应的url为 http://校园微哨资源服务器/file\_uri。  其中file\_uri为上传资源后返回的数据。支持JPG、PNG格式，较好的效果为大图640\*320，小图80\*80。 |
| url | 点击图文消息跳转链接 |

## 6．消息推送

### 6.1查询剩余广播消息条数

|  |
| --- |
| **接口（请求方式：get）** |
| https:// 校园微哨通讯服务器/lightapp/push?type=restcount&appid=appid&access\_token=ACCESS\_TOKEN |

|  |  |
| --- | --- |
| **参数** | |
| type | 消息类型，值必须为”restcount” |
| app\_id | 应用id |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

|  |
| --- |
| **返回说明** |
| 错误消息：  {"error\_code":"5","error":"invalid token"}  正确响应：  {"error\_code":"0","error":"ok","timestamp":"0","msg\_count":"10"} |

### 6.2广播消息(发送给所有订阅用户)

|  |
| --- |
| **接口（请求方式：post）** |
| https:// 校园微哨通讯服务器/lightapp/push?type= broadcast &from=appid&access\_token=ACCESS\_TOKEN |

Post的消息内容参见第五部分“消息回复”

|  |  |
| --- | --- |
| **参数** | |
| type | 消息类型，值必须为”broadcast” |
| from | 发送方appid |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

### 6.3定向消息(发送给特定用户)

|  |
| --- |
| **接口（请求方式：post）** |
| https:// 校园微哨通讯服务器/lightapp/push?type=user&from=appid&to=123456@dgut.edu.cn&access\_token=ACCESS\_TOKEN |

Post的消息内容参见第五部分“消息回复”

|  |  |
| --- | --- |
| **参数** | |
| type | 消息类型，值必须为”user” |
| from | 发送方appid |
| to | 接收方AID, 例如123456@dgut.edu.cn |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

### 6.4 消息推送格式

#### 6.4.1 文本消息

<xml>

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

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

<createtime>12345678</createtime>

<expirationtime>12345678</expirationtime>

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

<content><![CDATA[content]]></content>

</xml>

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 接收方帐号（收到的JID） |
| fromusername | 微哨轻应用ID (appid) |
| createtime | 消息创建时间 |
| expirationtime | 消息超期时间, 超期自动删除 |
| msgtype | text |
| content | 回复的消息内容,长度不超过2048字节 |

#### 6.4.2 图文消息

<xml>

<tousername><![CDATA[touser]]></tousername>

<fromusername><![CDATA[fromuser]]></fromusername>

<createtime>12345678</createtime>

<expirationtime>12345678</expirationtime>

<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>

**参数说明：**

|  |  |
| --- | --- |
| **参数** | **描述** |
| tousername | 接收方帐号（收到的JID） |
| fromusername | 微哨轻应用ID (appid) |
| createtime | 消息创建时间 |
| expirationtime | 消息超期时间, 超期自动删除 |
| msgtype | news |
| articlecount | 图文消息个数，限制为10条以内 |
| articles | 多条图文消息信息，默认第一个item为大图 |
| title | 图文消息标题 |
| description | 图文消息描述 |
| picurl | 图片链接(图片全路径)，如果是上传到微哨服务器的资源，则对应的url为 http://校园微哨资源服务器/file\_uri。  其中file\_uri为上传资源后返回的数据。支持JPG、PNG格式，较好的效果为大图640\*320，小图80\*80。 |
| url | 点击图文消息跳转链接 |

## 文件上传下载

### 7.1 文件上传

**接口说明：**

|  |
| --- |
| **接口（请求方式：**POST**）** |
| http://校园微哨资源服务器/upload?access\_token=3Zkx2gyQVaeIbS |

**参数说明：**

|  |  |
| --- | --- |
| **参数** | |
| access\_token | 所有请求，必须带有access\_token参数，access\_token获取方式参见第1部分--身份校验 |

**协议说明：**

|  |
| --- |
| **协议** |
| 上传成功后返回结果为：  {  “file\_uri”:” xxxxxxxx.jpg”,  "file\_id":"xxxxxxxxxxxxxxxx",  "create\_time " : 1335415895  }  返回值说明：  file\_uri:文件在资源服务器上的uri  file\_id:文件的唯一id（md5sum）  create\_time：文件创建时间，单位为秒 |

|  |
| --- |
| **请求样例（使用curl工具）：** |
| curl -K -F "file=@p45fw.jpg" https:// 校园微哨资源服务器/upload? appid=APPID&access\_token=ACCESS\_TOKEN |

### 文件下载

**接口说明：**

|  |
| --- |
| **接口（请求方式：**get**）** |
| http://校园微哨资源服务器/file\_uri |

**参数说明：**

|  |  |
| --- | --- |
| **参数** | |
| uri | 文件uri |

**协议说明：**

|  |
| --- |
| **协议** |
| 请求错误时的返回信息：  {"errcode":40007,"errmsg":"invalid uri"} |

|  |
| --- |
| **请求样例（使用curl工具）：** |
| curl –O https://校园微哨资源服务器 /file\_uri |

## 8. 注意事项

1.微哨开放平台不对外开放注册，登陆开放平台需先登录微哨客户端，然后点击开放平台菜单才可以进入。

2.微哨服务器请使用80端口

## 9. 返回码说明

|  |  |
| --- | --- |
| **返回码** | **说明** |
| -1 | 系统繁忙 |
| 0 | 请求成功 |
| 40001 | 验证失败 |
| 40002 | 不合法的凭证类型 |
| 40003 | 不合法的OpenID |
| 40004 | 不合法的媒体文件类型 |
| 40005 | 不合法的文件类型 |
| 40006 | 不合法的文件大小 |
| 40007 | 不合法的媒体文件id |
| 40008 | 不合法的消息类型 |
| 40009 | 不合法的图片文件大小 |
| 40010 | 不合法的语音文件大小 |
| 40011 | 不合法的视频文件大小 |
| 40012 | 不合法的缩略图文件大小 |
| 40013 | 不合法的APPID |
| 40014 | 不合法的access\_token |
| 40014 | 不合法的access\_token |
| 40015 | 不合法的菜单类型 |
| 40016 | 不合法的按钮个数 |
| 40017 | 不合法的按钮个数 |
| 40018 | 不合法的按钮名字长度 |
| 40019 | 不合法的按钮KEY长度 |
| 40020 | 不合法的按钮URL长度 |
| 40021 | 不合法的菜单版本号 |
| 40022 | 不合法的子菜单级数 |
| 40023 | 不合法的子菜单按钮个数 |
| 40024 | 不合法的子菜单按钮类型 |
| 40025 | 不合法的子菜单按钮名字长度 |
| 40026 | 不合法的子菜单按钮KEY长度 |
| 40027 | 不合法的子菜单按钮URL长度 |
| 40028 | 不合法的自定义菜单使用用户 |
| 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 |
| 42001 | access\_token超时 |
| 43001 | 需要GET请求 |
| 43002 | 需要POST请求 |
| 43003 | 需要HTTPS请求 |
| 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 |
| 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 |
| 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 |
| 47001 | 解析JSON/XML内容错误 |