

# WeHub API接口规范

## 修改记录

修改时间	作者	修改内容
2018.7.30	方清	规范action的业务名 以及request 和response的数据结构
2018.8.8	方清	重新定义回调接口下发的任务格式
2018.8.13	方清	新增(4,5,6,7,8)等5种新的任务,新增report_room_member_info 业务名
2018.8.22	方清	pull task的定时轮询时间可在wehub上设置;上报链接消息;修正wehub 多实例运行中的bug
2018.8.29	方清	新增文件上传功能(wehub上可以设置上传文件的类型),groupinfo 结构增加member_wxid_list字段,群成员发生变化时增量上报 (report_room_member_change);
2018.9.18	方清	新增"小程序,转账,发文件,个人名片,表情,语音,视频,微信系统消息"等消息事件的上报;新增report_friend_add_request,report_new_room两个action;新增发群公告,个人名片,删除好友,自动通过好友验证等4种新任务类型; 新增视频文件上传(微信版本号必须 >=2.6.4.56);相关数据结构有微调(格式向下兼容)

## 概述

weTool:

推宝科技在2017年推出的一款微信PC客户端的辅助产品,详情见[wetool网站]  
(<https://www.wxb.com/wetool>)

weHub:

weHub是一款类似于weTool的产品,它除了保留原来wetool就具备的各种功能,还提供了对接企业API的能力.第三方企业/个人(以下简称为第三方)需要开发一套符合weHub数据应答格式的web接口(以下简称为回调接口). weHub和回调接口采用c/s的方式进行应答,weHub向回调接口主动发起http request(post方式),回调接口返回http response

appid:

使用weHub服务的第三方需在weHub中配置一个appid和回调接口,appid需要第三方向推宝科技申请,申请时需提交第三方自己的回调接口地址,推宝科技会对该地址做审核,weHub会将相关的request数据post到这个回调接口上.第三方在使用weHub时首先要在weHub中配置appid和回调接口,weHub验证通过后会才post数据到该回调接口

wxid:

每一个微信号或者微信群,微信系统都定义了唯一的标识字符串.对于微信群,其唯一标识格式为xxxxxx@chatroom(如8680025352@chatroom);对于个人微信号,其格式wxid\_xxxxxxx(以wxid\_开头,如wxid\_p9597egc5j1c21)或者 xxxxxx(不以wxid\_开头,在注册微信时由注册者指定,如 fangqing\_hust). 本文档中所有数据结构中的"wxid"/"room\_wxid"字段即代表微信号/群的唯一标识字符串.

## 基本的数据结构(request/response)

wehub主动发起的数据(简称为:request)json格式为:

```
{
  "action": "具体业务名",
  "appid": "第三方申请的id",
  "wxid": "当前登录的wxid",
  "data": {具体业务的相关数据}
}
```

回调接口返回的数据(简称为:response)json格式为:

```
{
  "error_code": 0,           //0代表没有错误,其他的第三方自定义
  "error_reason": "",        //如果出错,这里写出错原因的描述,否则留空
  "ack_type": "xxxx_ack",    //ack_type 字段不能缺失.为上面的业务名称+"_ack"
                              //比如logout对应logout_ack,
                              //report_new_friend对应report_new_friend_ack
  "data": {
    xxxxx:xxxxxx             //具体的附带的数据,不同的业务返回的数据格式见下面具体的业务描述,
                              //如果没有要附带的数据,则为空
  }
}
```

对于response中带双引号的数据域,如果其语义为字符串,则双引号不可缺少

如:"ack\_type":"login\_ack"

ack\_type字段其值代表具体的业务名称(字符串), 因此login\_ack前后的""符号不能省略

又如: "error\_code": 0,

error\_code其值语义为一个具体的错误码(数字),因此0前后不需要""符号

注意:wehub发送的request 以utf-8编码,回调接口返回的response 中的json格式数据 wehub 也以utf-8编码来解析, 文档的示例代码中出现的 "\$xxx" 符号代表这里应该出现一个名为"xxx"结构的数据域,如 \$memberInfo,\$task,\$report\_msgunit等

## 业务数据定义

当前的支持的业务名有:login,logout,report\_contact,report\_new\_friend, report\_new\_msg,pull\_task,report\_task\_result , report\_room\_member\_info,report\_room\_member\_change,report\_friend\_add\_request,report\_new\_room. 这些业务名代表当前wehub支持的业务能力

## 微信登录通知/login

数据格式见上面的例子中的描述,这是appid验证通过并且微信登陆后回调接口受到的第一个request

```
{
  "action" : "login",           //登录的业务名为"login"
  "appid": "xxxx",             //申请的appid
  "wxid" : "wxid_fo1039029348sfj", //当前登陆的微信账号的wxid
  "data" : {
    "nickname": "Bill",        //微信昵称
    "wx_alias": "mccbill",     //微信号(有可能为空)
    "head_img": "http://xxxxxx" //微信的头像地址
  }
}
```

回调接口返回(response):

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "login_ack",
  "data": {}
}
```

## 微信退出通知/logout

request格式:

```
{
  "action": "logout",
  "appid": "xxxxxx",           //申请的appid
  "wxid": "wxid_xxxxxxx",
  "data": {}
}
```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "logout_ack",
  "data": {}
}
```

## 上报当前好友列表和群列表/report\_contact

这是紧接着微信登录通知之后发送的request

request格式

```
{
  "action": "report_contact",
  "appid": "xxxxxx",
  "wxid": "wxid_xxxxxxx",
  "data": {
    "group_list": [$groupinfo, $groupinfo, $groupinfo, .....],
    "friend_list": [$userinfo, $userinfo, $userinfo, ...]
  }
}
```

data中相关字段描述

- groupinfo(群信息结构):
 

```
{
        "wxid": "xxxxxx",           //群的wxid,格式为 xxxxx@chatroom
        "name": "xxxxxx",          //群名称
        "owner_wxid": "xxxxxx",    //群主的wxid
        "member_count": 100,       //该群成员总数
        "head_img": "http://xxxxxx" //群的头像的url地址
        "member_wxid_list" : ['wxid_xxx1', 'wxid_xxx2', ...] //当前群的成员wxid的列表
      }
```
- userinfo(好友信息结构)
 

```
{
        "wxid": "wxid",           //wxid
        "wx_alias": "xxxxx",      //微信号(有可能为空)
        "nickname": "xxxxx",      //微信昵称
        "remark_name": "xxxx",    //好友备注
        "head_img": "http://xxxxxx" //头像的url地址
      }
```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_contact_ack",
  "data": {}
}
```

## 上报群成员详细信息/report\_room\_member\_info

触发时机: 由回调接口通过下发"任务"来被触发

```
{
  "action": "report_room_member_info",
  "appid": "xxxxxxx",           //申请的appid
  "wxid": "wxid_fo1039029348sfj",
  "data": {
    room_data_list: [
      {
        "room_wxid": "xxxxx1@chatroom", //群wxid
        "memberInfo_list": [$memberInfo, $memberInfo.....]
        //群内成员信息
      },
      {
        "room_wxid": "xxxxx2@chatroom",
        "memberInfo_list": [$memberInfo, $memberInfo.....]
      },
      .....
    ]
  }
}

memberInfo 结构
{
  "wxid": "wxid",           //wxid
  "wx_alias": "xxxxx",      //微信号(有可能为空)
  "room_nickname":          //这个微信号的群昵称
  "nickname": "xxxxx",      //微信昵称
  "head_img": "http://xxxxxxx" //头像的url地址
}
```

## 上报群成员变化/report\_room\_member\_change

触发: 群成员增加或减少时上报

```
request
{
  "action": "report_room_member_change",
  "appid": "xxxxxxx",
  "wxid": "wxid_xxxxxxx",
  "data": {
    "room_wxid": "xxxxxxx@chatroom",
    "wxid_list": ['xxxxxx', 'xxxxx'], //变化的成员的wxid列表
    "flag": flag //0, 群成员减少; 1, 群成员增加
  }
}

response
{
  "error_code": 0,
  "error_reason": "",
}
```

```
"ack_type": "report_room_member_change_ack",
"data": {}
}
```

## 上报新群/report\_new\_room

```
request
{
  "action": "report_new_room",
  "appid": "xxxxxxx",
  "wxid": "wxid_xxxxxxx",
  "data": {
    "wxid": "xxxxx",    //新群的wxid
    "name": "xxxx",    //群名(可能为空)
    "owner_wxid": "xxxxx", //群主的wxid
    "head_img": "xxxx", //群头像的url地址
    "memberInfo_list": [$memberInfo, $memberInfo, .....] //见memberInfo结构
  }
}

response
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_new_room_ack",
  "data": {}
}
```

## 上报新的聊天消息/report\_new\_msg

触发时机:当收到私聊消息或所在的某个群内有人发言(含自己发送的消息)

**对于做了自动回复功能的第三方接口,需要过滤自己发的消息.否则会陷入"回调接口下发自动回复-->wehub发消息-->微信消息事件回调--->wehub上报刚才自己发的消息--->回调接口又下发聊天任务"的死循环**

一旦陷入死循环,容易导致wehub高频率地发消息,这极易触发微信系统的安全提醒甚至被封号

如何过滤自己发的消息?

若使用的wehub版本<0.1.4,回调接口在收到report\_new\_msg时,判断msg中的wxid是否为这个wehub上登陆的wxid.若是,则不要下发task\_type=1的发消息的任务

若使用的wehub版本>=0.1.4,回调接口在收到report\_new\_msg时,判断msg中的wxid\_from是否为这个wehub上登陆的wxid.若是,则不要下发task\_type=1的发消息的任务

request

```

{
  "action" : "report_new_msg",
  "appid": "xxxxxxx",           //申请的appid
  "wxid" : "wxid_fo1039029348sfj",
  "data" : {
    "msg": $report_msgunit      //上报单条消息
    //report_msgunit格式见[上报的消息单元的格式]
  }
}

```

response (为便于扩展,回复的内容以**"任务"**的形式返回,"任务"的种类之后会扩展)

```

{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_new_msg_ack",
  "data": {
    "reply_task_list": [$task, $task, ...] //任务列表,支持多个任务,但不要超过3个任务
    //如果没有要回复的任务,则列表为空
    //$task格式见[response中下发的任务格式]
  }
}

```

#### 上报的消息单元的格式(\$report\_msgunit)

聊天消息类型	类型值msg_type	是否支持任务下发	是否支持上传消息中的文件
文本消息	1	支持	----
图片消息	3	支持	支持消息中的图片文件上传
个人名片	42	支持	----
语音	34	暂不支持	暂不支持
视频	43	支持	支持消息中的视频文件上传
表情消息	47	暂不支持	不支持
链接消息	49	支持	----
小程序	4901	暂不支持	----
转账	4902	暂不支持	----
文件	4903	暂不支持	暂不支持
微信系统通知	10000	不支持	----

```

- 文本消息
{
  "msg_type": 1,                //1 代表文本消息
  "room_wxid": "xxxxxxx@chatroom", //聊天消息发生在哪个群(如果是私聊则为空)
}

```

```

    "wxid_from": "wxid_xxxxxx", //消息发送者的wxid
    //如果是自己发的消息这里的wxid就是自己的微信号
    "wxid_to": "wxid_xxxxx", //消息的接收者的wxid
    //如果发往群的消息,这个值就是群的wxid
    //如果是别人私聊给自己的消息,这里就是自己的微信号
    "msg": "xxxxxxxx" //具体的文本内容
}

```

例如:

A用户在B群里发了一条消息:

```

"room_wxid": "B群wxid",
"wxid_from": "A的wxid",
"wxid_to": "B群wxid",

```

A给我私聊了一条消息:

```

"room_wxid": "",
"wxid_from": "A的wxid",
"wxid_to": "我的wxid",

```

我在B群里发了一条消息:

```

"room_wxid": "B群wxid",
"wxid_from": "我的wxid",
"wxid_to": "B群wxid",

```

我向A发了一条私聊消息:

```

"room_wxid": "",
"wxid_from": "我的wxid",
"wxid_to": "A的wxid",

```

- 图片消息

```

{
    "msg_type": 3, //3 代表图片消息
    "room_wxid": "xxxxxxxx@chatroom", //同文本消息
    "wxid_from": "wxid_xxxxxx", //同文本消息
    "wxid_to": "wxid_xxxxx", //同文本消息
    "file_index": "xxxxxx" //图片文件的唯一索引(由wehub生成)
    //该字段在wehub上报消息时有效
}

```

- 链接消息(分享某个网页链接)

```

{
    "msg_type": 49, //49 代表链接消息
    "room_wxid": "xxxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "link_url": "http://xxxxx", //分享链接的url
    "link_title": "标题", //链接标题
    "link_desc": "副标题", //链接描述(副标题)
    "link_img_url": "http://xxxxxx" //链接的缩略图的url, jpg或者png格式
}

```

- 表情消息

```

{
    "msg_type": 47,
    "room_wxid": "xxxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
}

```



```

    "emoji_url": "xxxxxxxx" //表情的url地址(若有需要,请回调接口自行下载该文件)
    "raw_msg": "xxxxxxx"
}

- 小程序
{
    "msg_type": 4901,
    "room_wxid": "xxxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "raw_msg": "xxxxxxx" //微信中的小程序信息的原始数据,xml格式,请自行解析
                        //username,nickname 为关键字段
}

- 转账事件
触发时机:
情况一:我转账给他人
    1.发起转账时:wxid_from='我的wxid',wxid_to='他人的wxid',paysubtype=1
    2.对方确认收账时:wxid_from='他人的wxid',wxid_to='我的wxid',paysubtype=3
情况二:他人转账给我:
    1.发起转账时:wxid_from='他人的wxid',wxid_to='我的wxid',paysubtype=1
    (只有这种情况下才能自动收账,格式见自动收账任务)
    2.我确认收账时:wxid_from='我的wxid',wxid_to='他人的wxid',paysubtype=3
{
    "msg_type": 4902,
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "transferid": "xxxxxxx" //转账的ID
    "paysubtype": paysubtype, //这笔账单的状态
    "raw_msg": "xxxxxxx" //微信中的转账事件的原始数据,xml格式
}

- 发送文件
(目前还不能获取到消息中文件的内容,暂时不支持上传,file_index预留为空值)
{
    "msg_type": 4903,
    "room_wxid": "xxxxxxxx@chatroom", //发生在那个群里
    "wxid_from": "wxid_xxxxxx", //文件发送者
    "wxid_to": "wxid_xxxxxx", //文件接收者
    "file_index": "",
    "raw_msg": "xxxxxxx" //微信中的文件的原始消息,xml格式,请自行解析
}

- 个人名片
{
    "msg_type": 42,
    "room_wxid": "xxxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "raw_msg": "xxxxxxx" //微信中的名片信息的原始数据,xml格式,请自行解析
}

- 语音消息
(目前还不能获取到消息中语音文件的内容,暂时不支持上传,file_index预留为空值)
{

```

```

    "msg_type":34,
    "room_wxid": "xxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "file_index": "",
    "raw_msg": "xxxxxxx"    //微信中的原始消息,xml格式
}
- 视频消息
{
    "msg_type":43,
    "room_wxid": "xxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "file_index": "xxxxxx",    //视频文件的索引
    "raw_msg": "xxxxxxx",    //视频文件详细信息(文件大小length,播放时长playlength),
                                //需服务端自行解析;可根据文件大小来判断是否要上传
}

- 微信系统通知
{
    "msg_type":10000,
    "room_wxid": "xxxxxxx@chatroom",
    "wxid_from": "wxid_xxxxxx",
    "wxid_to": "wxid_xxxxxx",
    "raw_msg": "xxxxxxx"    //具体的通知内容,纯文本格式
}
无需对系统通知做自动回复
eg:
1.发消息-被对方拉黑之后,raw_msg 为"消息已发出,但被对方拒收了"
2.有红包出没时:"发出红包,请在手机上查看"
3.修改群名称后:"你修改群名为"新的群名称123ABcabc""
4.修改群公告后:""设置群公告内容xxxxxx""
其他:
    "群主已恢复默认进群方式。"
    "群主已启用“群聊邀请确认”,群成员需群主确认才能邀请朋友进群。"
    "你已成为新群主"
    ""老情人"已成为新群主"
    "你邀请xxxx加入了群聊"
    "xxxx邀请xxxx加入了群聊"
    "xxxxx通过扫描你分享的二维码加入群聊"
    "xxxxx通过扫描xxxxxx分享的二维码加入群聊"

```

## 文件上传

基本流程:

1. wehub收到图片消息/视频消息,上报基本信息(此时wehub仅仅通过report\_new\_msg上报该图片的索引file\_index值,不上传具体文件的二进制信息)
2. 回调接口根据file\_index的值查询当前服务端文件存储系统中是否已经存在该索引的文件,若需要wehub上传,在report\_new\_msg\_ack的response中携带文件上传的任务通知wehub上传二进制文件,格式见[response中下发的任务格式].若在wehub客户端设置了不上传某类型的文件,即使收到了上传指令wehub也不上传
3. wehub收到指令后通过第三方自定义的上传接口上传文件(post 方式)
4. 上传接口返回文件处理的结果

```
{
  'error_code':0,
  'error_reason':'',
  'ack_type':'upload_file_ack',
  'file_index':file_index      //接收到的文件的file_index
}
```

以下为 wehub向上传接口上传图片文件的http request示例

```
Content-Type: multipart/form-data; boundary="boundary_.oOo._OTg2Ng==MzU3Mg==MjEwNzE="
MIME-Version: 1.0
Content-Length: 46718
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en,*
User-Agent: Mozilla/5.0
Host: localhost.:5678

--boundary_.oOo._OTg2Ng==MzU3Mg==MjEwNzE=
Content-Type: text/plain
Content-Disposition: form-data; name="file_index"

3053020100044c304a02010002041cdc709b02032f56c10204a2e5e77302045b87dcfa0425617570696d675
f356665376666383735333737623337355f313533353633303538353633390204010400020201000400
--boundary_.oOo._OTg2Ng==MzU3Mg==MjEwNzE=
Content-Type: image/jpeg
Content-Disposition: form-data;
name="file";filename="6a0b2e8d81857cd9ac7cf4fcb6ac271fd409fc1d.jpg"

xxxxxxxxxxxxxxxxxxxx..... //图片的2进制字节流
xxxxxxxxxxxxxxxxxxxx.....
```

上传视频

```
Content-Type: multipart/form-data; boundary="boundary_.oOo._MTQ5NDU=Mjg1Nzk=MTAyNzE="
MIME-Version: 1.0
Content-Length: 1351338
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en,*
User-Agent: Mozilla/5.0
Host: localhost.:5678

--boundary_.oOo._MTQ5NDU=Mjg1Nzk=MTAyNzE=
```

```

Content-Type: text/plain
Content-Disposition: form-data; name="file_index"

306b0201000464306202010002041cdc709b02032f56c10204cde5e77302045b99fc32043d6175707669646
56f5f356665376666383735333737623337355f313533363831383232345f31373537303431333039313836
37613066633434323433300204010400040201000400
--boundary_.oOo._MTQ5NDU=Mjg1Nzk=MTAyNZE=
Content-Type: video/mp4
Content-Disposition: form-data;
name="file";filename="c899cebad9877280af73d4e595f5d1e41e7b1ed8.mp4"

xxxxxxxxxxxxxxxxxxxx..... //视频文件的2进制字节流
xxxxxxxxxxxxxxxxxxxx.....

```

注意: 服务端的上传接口接收到wehub的请求后需要取出 request中 file\_index的值

目前wehub只上传图片/视频类型的文件(之后上传的文件类型会拓展)

## 新的加好友请求/report\_friend\_add\_request

收到添加好友的请求(此时对方还不是我的好友,不能给对方发消息) 微信系统对每个账号每天通过的好友请求有限制(每天200左右) 服务端需要储存(v1,v2) 值, 以便通过任务下发的方式通过好友验证  
request格式

```

{
  "action" : "report_friend_add_request",
  "appid": "xxxxxxx",
  "wxid" : "wxid_fo1039029348sfj",
  "data" : {
    "v1": "xxxxxx",           //若要自动通过,请在ack中回传
    "v2": "xxxxxx",           //若要自动通过,请在ack中回传
    "raw_msg": "xxxxxxxxxx"   //微信中的原始消息,xml格式(xml中的content字段为打招呼的内容)
  }
}

```

response格式

```

{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_friend_add_request_ack",
  "data": {
    "op_code": 1 //1:通过好友请求(此时必须有v1和v2);其他值,不处理
    "v1": "xxxxxx",
    "v2": "xxxxxx"
  }
}

```

## 新好友通知/report\_new\_friend

每当有新的好友时,上报新好友的个人信息(对方已经成为了我的好友)

request格式

```
{
  "action" : "report_new_friend",
  "appid": "xxxxxxx",
  "wxid" : "wxid_fo1039029348sfj",
  "data" : {
    "fans_wxid": "wxid_ljsdlfjslfjl",          // 新好友的wxid
    "nickname": "Jerry",                      // 新好友的昵称
    "wx_alias": "jerry"                       // 新好友的微信号,可能为空
    "head_img": "xxxxx",                      //头像url
    "notice_word": "xxxxxxx"                  // 新好友加我时的打招呼的内容,可能为空
    "sourceusername": "xxxxx"                  //推荐人的wxid,可能为空
    "sourcenickname": "xxxxxxx"                //推荐人昵称,可能为空
  }
}
```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_new_friend_ack",
  "data": {
    "reply_task_list": [$task, $task, ...]      //任务列表
    // $task格式见[response中下发的任务格式]
  }
}
```

## 任务格式 \$task(response中下发的任务格式)

(将来会有更多的任务格式支持)

任务类型	类型值task_type
发消息任务	1(只支持文字,图片,链接,视频,个人名片)
踢人任务	2
拉群任务(邀请入群)	3
上报群成员信息	4
加群成员为好友	5
修改好友备注	6
修改群昵称	7
退群	8
上传文件	9 //通过report_new_msg_ack来下发
发群公告	10
自动收账	11
删除好友	12
通过好友验证	13

- 发消息任务：

(向一个微信群或一个微信号发一组消息单元)

```
{
  "task_type": 1,
  "task_dict": {
    //若发群消息:room_wxid为群对象,wxid为要@的对象(仅文字消息时生效)
    //若发私聊,room_wxid 为空,wxid为具体的私聊对象
    "room_wxid": "xxxxxx",
    "wxid": "xxxxxxx",
    "msg_list": [$push_msgunit,$push_msgunit,...]
    //待发送的消息单元列表
  }
}
```

发消息任务中的\$push\_msgunit格式

(1)文字消息

```
{
  'msg_type': 1,
  'msg': "xxxxxx"  发送的文字
}
```

(2)图片消息

```
{
  'msg_type': 3,
  'msg': "xxxx"  //图片的url绝对地址:http://xxxxxxx/xx.jpg或png
}
```

(3)链接消息

```

{
    "msg_type":49,                //49 代表链接消息
    "link_url":"http://xxxxx",    //分享链接的url
    "link_title":"标题",          //链接标题
    "link_desc": "副标题",        //链接描述（副标题）
    "link_img_url": "http://xxxxxxx" //链接的缩略图的url,jpg或者png格式
}
(4)视频消息
{
    "msg_type":43,
    "video_url":"http://xxxxxxx/xx.mp4" //回调接口推送给用户的视频的url地址, mp4格式
}
(5)个人名片
{
    "msg_type":42,
    "wxid_card":"xxxxxx"          //发送谁的个人名片
}

```

- 踢人任务:

(把一个微信号从指定的群踢出,当前微信必须有群主权限)

```

{
    "task_type":2,
    "task_dict":
    {
        "room_wxid":"xxxxx@chatroom", //被踢者所在的群,如果为空,则从所有的群踢出
        "wxid":"xxxxxxx"              //被踢者的wxid
    }
}

```

- 拉群任务:

(向一个好友发入群邀请,注意必须是自己的好友)

```

{
    "task_type":3,
    "task_dict":
    {
        "room_wxid":"xxxxx@chatroom", //目标群
        "wxid":"xxxxxxx"              //被拉进群的wxid
    }
}

```

- 上报群成员信息:

(上报某个群里所有的群成员的详细信息,如果群成员比较多,上报的数据量会比较大)

```

{
    "task_type":4,
    "task_dict":
    {
        //要上报的群列表
        "room_wxid_list":["xxxxx@chatroom","xxxxx2@chatroom"....]
    }
}

```

wehub 通过report\_room\_member\_info来主动上报,详情见[上报群成员详细信息]

- 加群成员为好友:

```

{
    "task_type":5,

```

```

    "task_dict":
    {
        "room_wxid":"xxxxx@chatroom",    //群
        "wxid":"xxxxxxx"                //群成员的wxid
        "msg":"xxxxxx"                  //打招呼消息,文本
    }
}
- 修改好友备注:
{
    "task_type":6,
    "task_dict":
    {
        "wxid":"xxxxxx",    //好友微信
        "remark_name":"xxxxxx" //好友备注
    }
}
- 修改群昵称:
(修改当前微信号在某个群里的群昵称)
{
    "task_type":7,
    "task_dict":
    {
        "room_wxid":"xxxxxx",    //微信群wxid
        "room_nickname":"xxxxxx" //我在这个群里的昵称
    }
}
- 退群:
{
    "task_type":8,
    "task_dict":
    {
        "room_wxid":"xxxxxx",    //要退出的微信群wxid
    }
}
-上传文件
{
    "task_type":9,
    "task_dict":
    {
        "file_index":"xxxxxxx",    //file_index的值由wehub维护
        "flag":1    //若为0代表不用上传,1代表需要上传
    }
}
//第三方需要将wehub上传的文件保存起来,建立file_index与上传文件的对应关系

- 发群公告
{
    "task_type":10,
    "task_dict":
    {
        "room_wxid":"xxxxxx",    //微信群
        "msg":"xxxxxx" //群公告的内容
    }
}

```



```

}

- 自动收账（只能收取别人发给自己的转账）
{
  "task_type":11,
  "task_dict":
  {
    "wxid_from": "xxxxxx"    //转账发起者wxid
    "transferid":"xxxxxx"    //transferid:自动收哪一笔转账
  }
}

- 删除好友
{
  "task_type":12,
  "task_dict":
  {
    "wxid_delete": "xxxxx"    //要被删除的好友的wxid
  }
}

-通过好友验证
{
  "task_type":12,
  "task_dict":
  {
    "v1": "xxxxx",
    "v2": "xxxxx"
  }
}
}

```

## 向回调接口请求一个任务(pull\_task)

wehub在appid验证通过以后,每间隔x秒请求一次(时间间隔可在wehub上设置,最少1秒,默认是5秒) request格式

```

{
  "action": "pull_task",
  "appid": "xxxxxx",
  "wxid" : "wxid_xxxxxxxxx",
  "data":{}
}

```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "pull_task_ack",
  "data": {
    //wehub通过task_id来识别不同的任务(task_id其值是由回调接口生成的字符串,请保证有唯一性)
    "task_id": "任务id",
    "task_data": $task    //单个任务
    //$task格式见[response中下发的任务格式]
    //wehub取到任务以后,会立即开始执行,执行完成后会把结果异步地反馈给回调接口
  }
}
```

## 向回调接口反馈任务执行的结果/report\_task\_result

request格式

```
{
  "action": "report_task_result",
  "appid": "xxxxxxx",
  "wxid" : "wxid_xxxxxxxx"
  "data":
  {
    "task_id": "任务id",
    "task_result": 1,    //0,任务执行失败,1任务执行成功
    "error_reason": ""   //为什么执行失败,若任务执行成功则为空
  }
}
```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_task_result_ack",
  "data": {
    "task_id": "任务id"
  }
}
```

流程图

