

WeHub API接口规范v1.2

修改记录

修改时间	版本号	作者	修改内容
2018.7.30	v1.0	方清	规范action的业务名 以及request 和response的数据结构
2018.8.8	v1.0	方清	重新定义回调接口下发的任务格式
2018.8.13	v1.1	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	v1.1	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	v1.2	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);

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)]
(<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,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 xxxxxxx(不以wxid_开头,在注册微信时由注册者指定,如 fangqing_hust). 本文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

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发送的请求以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

以登录为例, 当微信登录, 则会向回调接口上报如下数据(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": {}
}
```

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 , 这些业务名代表当前wehub支持的业务能力

微信登录通知

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

微信退出通知

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

这是紧接着微信登录通知之后发送的request

(因为群中成员数据量太大了,所以这里没有上报群内部的成员列表)

request格式

```
{
  "action": "report_contact",
  "appid": "xxxxxx",
  "wxid": "wxid_xxxxxxx",
  "data": {
    "group_list": [
```

```

        $groupinfo1,
        $groupinfo2,
        $groupinfo3,
        .....
    ],
    "friend_list":[
        $userinfo1,$userinfo2,$userinfo3
    ]
}
}
data中相关字段描述
- groupinfo(群信息结构):
{
    "wxid": "xxxxxxx",                //群的wxid,格式为 xxxxx@chatroom
    "name": "xxxxxx",                //群名称
    "owner_wxid": "xxxxxxx",          //群主的wxid
    "member_count": 100,              //该群成员总数
    "head_img":"http://xxxxxxx"      //群的头像的url地址
    "member_wxid_list" :['wxid_xxx1','wxid_xxx2','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://xxxxxxx"     //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

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": [$memberInfo1,$memberInfo2.....]
            }
        ]
    }
}

```

```

        //群内成员信息
    },
    {
        "room_wxid":"xxxxx2@chatroom",
        "memberInfo_list":[$memberInfo1,$memberInfo2.....]
    },
    .....
]
}
}

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

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":{}
}

```

上报新的聊天消息

触发时机:当收到私聊消息或所在的某个群内有人发言,目前wehub上报文本消息 和链接消息

request

```
{
  "action" : "report_new_msg",
  "appid": "xxxxxxx",           //申请的appid
  "wxid" : "wxid_fo1039029348sfj",
  "data" : {
    "msg": $msgunit           //上报单条消息,格式见下方
  }
}
```

聊天消息单元的格式(\$msgunit)

聊天消息类型	类型值msg_type
文本消息	1
图片消息	3
链接消息	49

```
- 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": "wxid_xxxxxx",       //聊天消息发送者的wxid
                                //如果是自己发的消息这里的wxid就是当前登陆的微信号
  "msg": "xxxxxxx"             //具体的文本内容
}
```

```
- 图片消息
{
  "msg_type": 3,                //3 代表图片消息
  "room_wxid": "xxxxxxx@chatroom", //同文本消息
  "wxid": "wxid_xxxxxx",       //同文本消息
  "msg": "xxxxxxx"             //图片的url绝对地址:http://xxxxxxx/xx.jpg或png
                                //该字段在回调接口下发任务时有效
  "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
}
```

当wehub向回调接口上报图片消息时,忽略msg字段,file_index为图片文件的唯一索引,之后wehub通过文件上传接口上传该图片(具体见文件上传)

当回调接口下发图片消息任务时,忽略upload_image_index字段,msg为图片的绝对地址

```
- 链接消息(分享某个链接)
{
  "msg_type": 49,               //49 代表链接消息
  "room_wxid": "xxxxxxx@chatroom", //同文本消息
  "wxid": "wxid_xxxxxx",       //同文本消息
  "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
}
```

```
}
```

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

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": [$task1,$task2...] //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中的任务格式]
  }
}
```

文件上传

基本流程:

1. wehub收到图片消息,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,二进制相同的文件,其file_index是一样的)
2. 回调接口根据file_index的值查询第三方的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中的任务格式]
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向上传接口上传文件时的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"
```

```
3053020100044c304a02010002041cdc709b02032f56c10204a2e5e77302045b87dcfa0425617570696d675f356665376666383735333737623337355f313533353633303538353633390204010400020201000400
```

```
--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.....
```

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只上传图片类型的文件(之后上传的文件类型会拓展)

新好友通知

每当有新的好友时,上报新好友的个人信息

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
    "notice_word": "xxxxxxx"                   // 新好友加我时的打招呼的内容,可能为空
  }
}
```

response格式

```
{
  "error_code": 0,
  "error_reason": "",
  "ack_type": "report_new_friend_ack",
  "data": {
    "reply_task_list": [task1, task2...]
    // 这里的回复格式和report_new_msg_ack 一样
  }
}
```

response中的任务格式

(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
踢人任务	2
拉群任务	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 //只通过report_new_msg_ack来下发

- 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,      //任务类型
  "task_dict":
  {
    //若发群消息:room_wxid为群对象,wxid为要@的对象
    //若发私聊,room_wxid 为空,wxid为具体的私聊对象
    "room_wxid": "xxxxxx",
    "wxid": "xxxxxxx",
    "msg_list": [$msgunit, $msgunit2....]
    //将一条或多条消息单元发送给receiver_wxid
    // $msgunit中的room_wxid,wxid等信息将被忽略
    //格式见[聊天消息单元的格式($msgunit)]
  }
}
```

- 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与存储文件的索引
```

向回调接口请求一个任务(pull_task)

wehub在appid验证通过以后,每5秒请求一次 request格式

```
{
  "action": "pull_task",
  "appid": "xxxxxx",
  "wxid" : "wxid_xxxxxxxx",
  "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
    //具体的任务格式见[response中的任务格式]
    //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

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

