脸脸网服务器与手机端的开发接口API

YXY

2013 年 6 月 18 日

目录

注意事项：

1. 所有的链接，如果要获得json格式的数据，最好带上".json"的后缀。因为同一个链接以后可能返回多种格式的数据，比如xml/html/json等。

2. 当调用出错时，会在返回的json中包含"error"信息。

3. 只有开发调试时调用http的接口，发布时全部使用https接口。

1 初始化与升级

1.1 初始化

手机应用启动后，初次访问时调用此API。

GET	https://42.121.79.210/init/init		
参数	类型	必填	说明
model	字符串	必须	硬件型号
os	字符串	必须	手机操作系统版本
mac	字符串	必须	无线网卡MAC地址的MD5
hash	字符串	必须	hash验证码
ver	字符串	可选	软件版本

注意：本API直接通过IP地址“42.121.79.210”访问，不通过DNS。而后续所有API调用的IP地址根据本调用的返回的IP地址确定。在目前只有单台服务器的情况下，返回的ip地址也是“42.121.79.210”，但是以后会根据离用户的距离返回就近的IP地址。

hash的计算方式为：model+os+mac+”init”进行SHA1算法后取前32位，和登录时的hash算法［2.4］类似。

返回值：

```
{
"ip": "60.191.119.190",
"xmpp": "60.191.119.190",
"ver":1.0
}
```

后续的`http`访问使用返回的`ip`地址，`xmpp`协议的访问使用`xmpp`地址。

[2012-12-27] 新增ver，表示脸脸当前线上的最新版本。客户端根据这个版本号可以判断是否需要升级。如果需要升级，提醒用户一次，不要每次启动重复提醒。比如1.1版本时提醒一次，如果用户忽略了，那么只有当版本升级到了1.2时再提醒一次。

本API的作用一个是给客户端推荐IP地址，一个是统计应用的开机情况和操作系统分布。

TODO: android

1.2 获得最新的升级信息

如果客户端判断出版本不是最新的，调用本接口获得最新的升级信息。

GET	https://42.121.79.210/init/upgrade		
参数	类型	必填	说明
os	字符串	必须	手机操作系统版本
ver	字符串	必须	客户端软件版本

返回值：

["2.0.1","聊天室发图增加了分享到微信朋友圈和微信好友功能\n界面美化，更美观更清新",true]

分别代表：最新的软件版本、功能介绍、是否强制升级。

如果不强制升级，那么弹出升级对话框时有两个选择：立即升级 / 暂不升级。

如果强制升级，那么弹出升级对话框时只有一个选择：立即升级。强制升级用于发生不兼容改动时。强制升级也只应该提示一次，防止升级失败导致死循环。

2 登录和退出

2.1 网页登录接口：oauth2

POST	/oauth2/sina_callback		
参数	类型	必填	说明
code	字符串	必须	新浪返回的code

例子：

访问：/oauth2/sina_callback?code=...
该访问由新浪微博通过302转向发起。
返回值：

```
{
"id":1,
"password":"15c663b8ff620502",
"logo":"",
"name" : "name"
"gender" : 1
"wb_uid":"1884834632",
"expires_in":75693,
"token":"2.00aaZYDCMcnDPCb4dc439e06i2m_GC",
"expires_at":1338431259
}
```

其中，id和password是该用户在脸脸网的id和密码，在首次新浪授权时创建。该id加上"@dface.cn"就是openfire的jid。（目前还没有和jid关联上。）

后面几个字段都是新浪提供的，其中wb_uid是该用户的新浪微博的uid，token是授权的access_token，其它几个是授权有效期信息。

在首次登录获得用户的id和password以后，可以缓存到本地。以后用户每次登录，返回的id和password是不会改变的，除非服务器端重置密码（比如出现密码泄漏等安全情况）。

logo可以得知用户是否成功上传头像。对于第一次登录的用户，其logo为空，此时需要引导用户上传头像。

本接口不再需要，登录逻辑为：如果客户端安装了支持SSO的微博客户端，那么使用SSO的接口登录；否则使用xauth的接口登录。

2.2 客户端登录接口：xauth

POST	/oauth2/login		
参数	类型	必填	说明
name	字符串	必须	用户名
pass	字符串	必须	密码
mac	字符串	必须	网卡Mac地址
hash	字符串	必须	hash验证码
bind	整数	可选	bind为0表示正常登录，1表示绑定微博帐户，2表示已经绑定过微博且已经用qq登录时，同时登录微博。默认为0

本接口的输出和oauth2登录接口的输出相同。

hash的计算方式为：name+pass+mac+"dface"进行SHA1算法后取前32位。
比如用户名为name，密码为pa，mac地址为ss
那么待hash的字符串为"namepassdface",
其hash码为"11f4004a73a65117071bc4a7d3dfdf07"。

新的登录方式不需要调用新浪的Xauth API，直接调用本API即可以登录。客户端应该不需要保存AppSecret。通过预先保存的AppKey和本调用输出中包含的token应该就可以访问新浪微博的API了。

2.3 客户端SSO接口

POST	/oauth2/sso		
参数	类型	必填	说明
remind_in	字符串	必须	过期时间
expires_in	字符串	必须	过期时间
uid	字符串	必须	微博的uid
access_token	字符串	必须	微博的token
hash	字符串	必须	hash验证码
bind	整数	可选	bind为0表示正常登录，1表示绑定微博帐户，2表示已经绑定过微博且已经用qq登录时，同时登录微博。默认为0

手机客户端使用微博SSO登录后，调用本接口。hash的计算方式为：uid+access_token+"dface"进行SHA1算法后取前32位。

2.4 客户端QQ接口

POST	/oauth2/qq_client		
参数	类型	必填	说明
expires_in	字符串	必须	过期时间
openid	字符串	必须	QQ的openid
access_token	字符串	必须	QQ授权的token
hash	字符串	必须	hash验证码
bind	整数	可选	bind为0表示正常登录，1表示绑定qq帐户，2表示已经绑定过qq且已经用微博登录时，同时登录qq。默认为0

手机客户端使用QQ提供的SDK登录后，调用本接口。hash的计算方式为：openid+access_token+"dface"进行SHA1算法后取前32位。

2.5 退出

POST	/oauth2/logout		
参数	类型	必填	说明
pushtoken	浮点数	可选	当前设备的push消息token

当用户主动退出时，调用本接口并断开XMPP的连接。[2012-12-30] 新增pushtoken参数。

2.6 解除QQ绑定

POST	/oauth2/unbind_qq		
参数	类型	必填	说明
uid	字符串	必须	当前用户的id
qq	字符串	必须	QQ分配的openid

解除成功返回unbind: true，否则返回error信息。新浪微博和QQ只有都处在有效登录状态时，才能解除其中的一个的绑定。

2.7 解除新浪微博绑定

POST	/oauth2/unbind_sina		
参数	类型	必填	说明
uid	字符串	必须	当前用户的id
wb_uid	字符串	必须	新浪微博的uid

解除成功返回unbind: true，否则返回error信息。

2.8 登录与绑定的状态判断

目前，脸脸帐号登录相关方有四个：脸脸web服务器、脸脸xmpp服务器，新浪微博，QQ。

1. 当访问web接口时，如果返回的error信息是"not login"，此时代表要登录脸脸web服务器。但是脸脸不提供独立的帐户体系，要登陆脸脸Web服务器，必须登陆新浪微博或者QQ中的一个。当脸脸登录时，如果客户端缓存有未过期的新浪微博或者QQ认证信息，要丢弃。Web服务器认证信息保存在_session_id中。_session_id默认没有过期时间。

2. 如果脸脸_session_id包含登录信息（没返回not login）：

   (a) 如果新浪微博或者QQ中的一个处于有效登录状态(wb_token/qq_token处在有效期内)，需要使用另外一个服务时：

      i. 如果以前绑定过另外一个的帐号，使用bind=2参数登录。

      ii. 如果没有，此时要绑定帐号。要使用bind=1参数登录。此时会绑定帐号并登录。

   (b) 如果新浪微博和QQ都没有登录，此时要返回到登录界面。

   (c) 绑定操作（bind=1）只需要执行一次，

   (d) 当用户的个人信息中有wb_uid，代表绑定过微博；有qq_openid，代表绑定过QQ。

   (e) 当用户的个人信息中有wb_token，代表登录了微博；有qq_token，代表登录了QQ。

   (f) 绑定操作可以取消。而对于登录操作，如果新浪微博或者QQ都没有登录，则不能取消。

3. 关于_session_id：

(a) 首次安装应用调用init接口时，生成_session_id。以后所有的调用包括后来的init调用也要带上_session_id。

(b) 即使没有登录也分配_session_id。

(c) 调用logout时会变更_session_id。

(d) 任何调用，如果没带_session_id，会生成新的_session_id。这样用户原来的所有状态信息会丢失，这是不建议的。

4. 总的来说就是：没有_session_id通过init调用获得_session_id；每次调用总是带上_session_id，logout时更新_session_id。

5. xmpp服务器是单独登录的。每次应用打开的时候登录，退出或切换到后台时退出。

2.9 关于首次登录的判断

以前，脸脸判断用户是否首次登录是看TA是否有头像。如果没有头像，则走注册流程。

现在更改为：每种登录接口（sina xauth/sso, qq）都会在用户首次登录时返回字段：newuser:1。当有newuser时，走注册流程。

3 QQ登录用户看他人微博

由于看他人微博需要微博登陆，所以使用QQ登陆而没有绑定微博的用户无法直接查看他人的微博。获取他人微博的接口是：

```
https://api.weibo.com/2/statuses/user_timeline.json
```

把上面的域名api.weibo.com替换为www.dface.cn，且不传递access_token参数，就可以通过脸脸查看他人的微博了。比如：

```
http://www.dface.cn/2/statuses/user_timeline.json?uid=1884834632
```

其中uid指的是新浪微博的wb_uid，其他参数和新浪的接口完全一致。参见：http://open.weibo.com/wiki/2/statuses/user_timeline

4 首次使用脸脸时发送分享信息

POST	/oauth2/share		
参数	类型	必填	说明
ver	浮点数	可选	当前安装的版本
t	整数	可选	分享类型：0代表微博，1代表qq空间，默认为0

首次使用脸脸 / 或者以后重大版本更新时，以登陆用户个人的名义发送使用脸脸的分享微博，微博由服务器端发送。

5 现场与附近

5.1 根据经纬度获得可能的现场商家

GET	/aroundme/shops		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
speed	浮点数	可选	速度m/s
bssid	字符串	可选	wifi上网时的bssid，如果不是wifi上网则忽略
baidu	整数	可选	如果是百度坐标，则baidu=1，否则忽略

注意：一定要在accuracy小于200m或者调用获取GPS的API超过三次后才能调用本接口。

Gps获取位置是一个逐步精确的过程。很多时候首次获取的误差超过1000米。此时显然无法准确获得商家列表。此时建议等待0.5秒再次获取gps。要保证gps的精确度小于200m，或则定位三次以上确实不能更准确了。宁可等待的时间久一些，也要保证定位的准确性。

最多返回50条数据，也有可能没有数据（比如西部沙漠地区）。

例子：

访问：/aroundme/shops.json
返回值：

[
{
"name":"新紫轩花店",
"address":"文一路266号",
"lng":120.12763,
"id":40056,
"phone":"",
"lat":30.28691,
"lo":[30.297,120.1288],
"t":1,
"user":0,
"male",0,
"female",0
}
]

除了返回商家的id、名称、电话、经纬度以后，增加了在该商家的用户总数“user”、男“male”、女“female”数量。

[2012－8－7] 新增lo字段。这是商家的实际经纬度，以前的lat/lng是地图上显示的经纬度，两者有几百米的误差。计算商家和自己的距离时，要采用lo来计算。

[2012－8－8] 新增t字段，表示商家的类型。

[2013－5－6] 新增coupon字段，表示商家是否由优惠券。

5.2 根据经纬度获得附近的热点商家

GET	/shop/nearby		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
name	字符串	可选	商家的名称
type	整数	可选	商家类型
baidu	整数	可选	如果是百度坐标，则baidu=1，否则忽略

商家类型：

1. 酒吧•活动
2. 咖啡•茶馆
3. 餐饮•酒店
4. 休闲•娱乐
5. 购物•广场
6. 楼宇•社区

用户当前所在的现场只有一个，但是由于定位有误差，所以服务器返回最有可能的几个商家让用户选择；而附近的商家有很多，按距离由近到远排序，且支持查询、分类筛选和分页。

5.3 根据经纬度获得附近的热点用户

GET	/aroundme/hot_users		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
baidu	整数	可选	如果是百度坐标，则baidu=1，否则忽略

输出格式参考“根据商家ID获得商家用户”的接口。唯一的不同点是：“根据商家ID获得商家用户”返回该用户在商家出现的time, 而本接口返回用户当前的location。

5.4 获得附近的用户

GET	/aroundme/users		
参数	类型	必填	说明
gender	数字	必填	用户的性别，未设置0男1女2

输出格式和“根据商家ID获得商家用户”的接口一样。

本接口用于“应用安装的时候，用户上传头像的页面”。其中有8张头像，是本地存储4张 / 网络获取（通过本接口）4张。其中本地的4张立刻显示。

5.5 地点报错

GET	/aroundme/shop_report		
参数	类型	必填	说明
uid	字符串	必须	登录用户的uid
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
bssid	字符串	可选	wifi上网时的bssid，如果不是wifi上网则忽略
baidu	整数	可选	如果是百度坐标，则baidu=1，否则忽略

本接口需要通过浏览器以URL的方式直接调用。

6 根据商家ID获得商家的信息

6.1 根据商家ID获得商家的基本信息

GET	/shop/info		
参数	类型	必填	说明
id	整数	必须	商家的ID

例子：

```
{
    "name": "浙江科技产业大厦",
    "t": 6,
    "lat": 30.280254,
    "lng": 120.121824,
    "address": "古翠路80",
    "id": 4928288,
    "staffs": [
        "502e6303421aa918ba000001"
    ],
    "notice":"",
    "text":"",
    "photos": []
}
```

staffs是该商家的员工数组。在商家聊天室，商家和商家的员工发言时都加V并加黄色背景。只有该商家在商家聊天室才加V，其它商家不做特殊处理。

notice是该商家发布的公告，已取消。

新增photos字段，表示该商家最热的图片，最多4张。

[2013-04-09] 新增字段text，表示用户在分享图片时，默认带上的文字。

6.2 根据商家ID获得商家的图片墙

商家的图片来自个人在聊天室的发图。

GET	/shop/photos		
参数	类型	必填	说明
id	整数	必须	商家的ID
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20

例子：


```
[{"id":"509a061fbe4b197297000002",
"room":"1",
"desc":"图片说明",
"updated_at":"2013-01-13T10:11:42Z",
"user_id":"502e61bfbe4b1921da000005",
"weibo":false,
"logo":"http://oss.aliyuncs.com/dface_test/509a061fbe4b197297000002/0.jpg",
"logo_thumb2":"http://oss.aliyuncs.com/dface_test/509a061fbe4b197297000002/t2_0.j
"id":"509a061fbe4b197297000002"}]
```

6.3 根据商家ID获得当前在该商家的所有用户

GET	/shop/users		
参数	类型	必填	说明
id	整数	必须	商家的ID
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20

例子：

```
[
{"id":1,
"logo":"/phone2/images/namei2.gif",
"name":"name23",
"wb_uid":1884834632,
"gender":0,
"friend":true,
"follower":false,
"birthday":null}
]
```


"friend"表示该用户是否是当前用户的朋友；"follower"代表该用户是否关注了当前用户。

[2012－8－8] 新增time字段，表示用户在该商家的最后活跃时间。
[2012－10－30] 新增分页参数。
[2013－5－322] 新增lord，lord=1表示某个用户是这里的地主。

在商家的用户的统计数据根据签到来统计，而签到则是当用户进入商家聊天室时自动后台发送。

6.4 根据商家ID获得商家的聊天室历史

商家的图片来自个人在聊天室的发图。

GET	/shop/history		
参数	类型	必填	说明
id	整数	必须	商家的ID
skip	整数	必须	跳过多少条消息
pcount	整数	可选	每页的数量，缺省为10

skip的大小应该是聊天室当前最新显示的消息的总数。这样传递的skip获得的历史消息才不会和聊天室当前的消息重复。

例子：

每次进聊天室，总是用下面的Xmpp接口获得最新的10条消息

```
<x xmlns='http://jabber.org/protocol/muc'>
  <history maxstanzas='10'/>
</x>
```

当要加载超过10条的消息时，才调用本接口。历史消息接口的数据从数据库中加在，而通过xmpp的history stanza加载的最新10条消息是从内存中加载的。

```
[["5160f00fc90d8be23000007c", "茉莉", 1367575501, "q9d8G-8"],
["5160f00fc90d8be23000007c", "哦你家", 1367575499, "q9d8G-7"]]
```

返回的是历史消息的数据，其中每个字段的含义是：

[发送者、消息内容、发送时间、消息ID]。

7 签到与足迹

7.1 进入现场签到

POST	/checkins		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
shop_id	整数	必须	现场商家id
user_id	整数	必须	签到用户id
od	整数	必须	实际签到的商家的排序
altitude	浮点数	可选	海拔高度
altacc	整数	可选	海拔高度的精确度
speed	浮点数	可选	速度m/s
bssid	字符串	可选	wifi上网时的bssid，如果不是wifi上网则忽略
ssid	字符串	可选	wifi上网时的ssid，如果不是wifi上网则忽略
baidu	整数	可选	如果是百度坐标，则baidu=1，否则忽略

注意：该请求必须是POST请求。如果是GET请求，获得的是签到列表。

[2012－8－10] 新增od字段，表示用户实际签到的商家在现场商家列表中的顺序位置，从1开始，也就是从/aroundme/shops中获得的商家列表中，那个被用户选中了。

7.2 输入商家的全称创建商家并签到

POST	/checkins/new_shop		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
user_id	整数	必须	签到用户id
sname	字符串	必须	商家的全称
type	字符串	可选	商家的类别
altitude	浮点数	可选	海拔高度
altacc	整数	可选	海拔高度的精确度
speed	浮点数	可选	速度m/s
bssid	字符串	可选	wifi上网时的bssid，如果不是wifi上网则忽略

注意：该请求必须是POST请求。

定位时，当用户当前的商家不存在时，可以通过输入商家的全称进入该商家的现场并签到。

本API接口和签到接口类似，不同之处在于：

1）定位时找到了商家，客户端签到（报告商家id / 商家的排名od），客户端同时进入该商家的聊天室。2）定位时找不到商家，用户输入商家的全称（sname），客户端调用本接口并获得新建立的商家的id，然后进入该商家的聊天室。

例子：

返回值：

{"name":"asdf","lo":[1,2],"lat":1,"lng":2,"address":null,"id":92264}

7.3 签到时输入地点名称查找地点

GET	/shop/add_search		
参数	类型	必填	说明
lat	浮点数	必须	经度
lng	浮点数	必须	纬度
accuracy	整数	必须	经纬度的精确度
sname	字符串	必须	商家的全称

定位时，如果在滚轮中找不到商家，在输入商家名称时查询本接口给用户提供实时匹配的可选择地点。

[2013-05-15] 新增参数visit，表示该地点是否仅仅可以围观。距离在误差范围内的地点，可以进入聊天室聊天。距离超过误差范围的地点，不能聊天，只能围观，也就是可以看照片墙和现场的人。默认visit为0，也就是可以进入聊天。

例子：

curl /shop/add_search?lat=30.28&lng=120.108&accuracy=100&sname=新村

返回值：

[{"id":10443603,"name":"高教新村","visit":0},{"id":10428968,"name":"古荡新村","visit":1}]

7.4 获得当前登录用户的足迹

GET	/user_info/trace		
参数	类型	必填	说明
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
hash	整数	可选	是否返回hash，缺省值为0

例子：

访问: /user_info/trace
返回值:
{"pcount":20,"data":
[{"id":"50b71b73c90d8b3c73000019","time":["09.22","14:24"],
"shop":"(下沙服务区)书报百货","shop_id":1,
"photos":[{"logo":"http://oss.aliyuncs.com/dface_test/5044301dbe4b192a30000002/0.
"logo_thumb2":"http://oss.aliyuncs.com/dface_test/5044301dbe4b192a30000002/t2_0.j
"id":"5044301dbe4b192a30000002","desc":null}]},
{"time":["07.21","00:02"],"shop":"海 洋 二
所","shop_id":60775,"photos":[]}]
}

其中pcount代表数据库中查询到的足迹的数量，如果该pcount等于客户端请求的数量，那么代表还会有更多的足迹。否则代表该用户没有足迹了。而data中保存的实际返回的足迹，其数量可能少于数据库中查询到的足迹。比如一个人在一天内在同一个地方的签到会被合并。

data中的字段说明:
{
id: 签到id
time: [签到日期, 签到时间]
shop: 商家
shop_id:商家id
photos : [{desc: 图片描述, id: 图片id, logo: 原图URL,
logo_thumb2: 缩略图}]
}

7.5 删除当前登录用户的一个签到足迹

POST	/checkins/delete		
参数	类型	必填	说明
id	字符串	必须	签到的id

注意：该请求必须是POST请求。

8 群组与认证

脸脸中的现场有两种：聊天室和群组。聊天室和地点绑定，在这儿的人可以加入，不在这儿的人不能加入。群组则是需要权限控制的。群组也分为两种：1.预先设置固定用户的（一般为商业活动相关的，需要预缴费的），实时确定能否加入，2.申请加入，等待群主审核的（一般的QQ群 / 陌陌群这类）。

8.1 加入群组的标示

在定位时（aroundme/shops）和搜索地点（shop/add_search）时，如果是有权限控制的群，会有“group_hint”字段。当用户进入该地点时，需要弹出一个输入框，提示文字为“group_hint”的内容。当用户输入完成后，调用下面“群组的认证”功能确定能否进入。

8.2 群组的认证

POST	/shop/auth		
参数	类型	必填	说明
shop_id	整数	必须	商家id
txt	字符串	必须	加入时输入的认证信息

返回值：

{ret:0} 代表认证失败
{ret:1} 代表通过认证
{ret:2} 代表等待审核
如果返回值为1，代表可以进入这个地点。进入一个地点时，后台有两个操作：1.加入Xmpp的聊天室，2.发送签到纪录。

当一个用户认证通过后，下次定位时，该地点不带group_hint”字段，也就不需要重复认证了。

9 用户基本信息

9.1 获得用户信息

GET	/user_info/get		
参数	类型	必填	说明
id	整数	必须	用户id

例子：

访问：/user_info/get?id=1
返回值：

{
"name":null,
wb_uid:1884834632
"gender":null,
"logo":"/system/imgs/1/original/clojure.png?1339398140",
"logo_thumb":"/system/imgs/1/thumb/clojure.png?1339398140"
"birthday":null,
"friend":true,
"follower":false,
"signature":"","job":null,"jobtype":null,"hobby":"Weiqi, go",
"last": "1天以前 顺旺基(益乐路店)"
"id":1
}

[2012—8—8] 新增签名、职业、爱好等字段。

[2013—5—21] 最后位置信息和关系信息都单独提供一个接口，以后这个接口将会取消，获得基本信息调用user_info/basic接口。

9.2 获得用户基本信息

GET	/user_info/basic		
参数	类型	必填	说明
id	整数	必须	用户id

例子：

访问：`/user_info/basic?id=50bec2c1c90d8bd12f000086`
返回值：

{"name":"amanda林","signature":"","wb_v":false,"wb_vs":"","gender":2,"birthday":"

9.3 获得用户和当前登录用户的关系

GET	/user_info/relation		
参数	类型	必填	说明
id	整数	必须	用户id

获得信息里有两个关系字段：friend和follower。这里的关系都是针对我的，这里的我就是当前登录用户。如果我关注了该用户，那么该用户就是我的friend; 如果该用户关注了我，那么他就是我的follower。

例子：

访问：`/user_info/friend?id=502e6303421aa918ba000001`
返回值：

{"id":"502e6303421aa918ba000001","friend":true,"follower":true}

9.4 获得用户最后出现的位置信息

GET	/user_info/last_loc		
参数	类型	必填	说明
id	整数	必须	用户id

例子：

访问：/user_info/last_loc?id=1
返回值：

```
{
"time":1天以前,
"last": "1天以前 顺旺基(益乐路店)"
"id":1
}
```

last字段，表示用户最后在脸脸上的时间和地点，一个以空格分割的字符串。

9.5 获得用户的地主地点列表

GET	/user_info/lords		
参数	类型	必填	说明
id	整数	必须	用户id

例子：

访问：/user_info/lords?id=502e6303421aa918ba000001
返回值：

[{"name":"千粥汇华星路店","lo":[30.281387,120.11969],"t":4,"lat":30.281387,"lng":1
"address":"","phone":"","id":21829233,"user":0,"male":0,"female":0},
{"name":"浦东国际人才城","lo":[31.189001,121.586937],"t":"10","lat":31.189001,"lng
"address":"","phone":"","id":21830801,"user":0,"male":0,"female":0}]

9.6 获得当前登录用户基本信息

GET	/user_info/get_self		
参数	类型	必填	说明

例子：

访问：/user_info/get_self
返回值：

{
"name":null,
wb_uid:1884834632
"gender":null,
"birthday":null,
"invisible":0,
"logo":"/system/imgs/1/original/clojure.png?1339398140",
"logo_thumb":"/system/imgs/1/thumb/clojure.png?1339398140"
"password":"c84dad462d5b7282",
"id":1
}

[2013-03-12] 新增
微博登陆的话：增加wb_token，wb_expire
qq登陆的话：增加qq_token，qq_expire

9.7 设置当前登录用户基本信息

POST	/user_info/set		
参数	类型	必填	说明
name	字符串	选填	用户的名字. 长度小于64
gender	数字	选填	用户的性别，未设置0男1女2
birthday	字符串	选填	用户的生日,格式如2012-06-01
signature	字符串	选填	签名档. 长度小于255
job	字符串	选填	职业
jobtype	整数	选填	职业类别
hobby	字符串	选填	爱好. 长度小于255
invisible	数字	选填	0：不隐身,1：对黑名单隐身,2：陌生人隐身, 3：全部隐身

注意：该请求必须是POST请求。

用户初次登录时获取新浪微博的名称 / 性别 / 出生日期等信息并提交到服务端。以后更新时可以只更新其中的某个字段。

隐身的效果是：1、在商家的用户列表中不在出现该用户2、不更新自己的位置信息3、用户进入商家时，不发打招呼的信息。

例子：

访问: curl -b "_session_id=ead9ac4f6291c55bb467ad4138eca2ed"
-F "name=newname" /user_info/set

返回值:

{
"name":newname,
"gender":null,
"birthday":null,
"logo":"/system/imgs/1/original/clojure.png?1339398140",
"logo_thumb":"/system/imgs/1/thumb/clojure.png?1339398140"
"id":1
}

9.8 设置其它用户的备注名

POST	/user_info/set_comment_name		
参数	类型	必填	说明
id	字符串	必填	被设置备注名的用户的id
name	字符串	必填	用户的名字. 长度小于64

当前登录用户给其它用户设置备注名，方便记忆。

返回值: {"success" : 1}

9.9 获得当前登录用户设置的所有备注名

GET	/user_info/get_comment_names		
参数	类型	必填	说明
user_id	字符串	必填	当前登录用户的id

当前登录用户给其它用户设置备注名，方便记忆。

访问：`/user_info/get_comment_names?user_id=502e6303421aa918ba000005`
返回值：

```
[{"id":"502e6303421aa918ba000001","s":"yxy"}]
```

id是被设置备注名的用户的id，s是备注名。

10 用户相册与头像

用户相册中的第一张图片就是头像。

10.1 当前登录用户上传图片

POST	/user_logos/create		
参数	类型	必填	说明
user_logo[img]	file	必须	头像文件
user_logo[t]	整数	可选	图片类型：1拍照；2选自相册

注意：

1. 该请求必须是POST请求，enctype=”multipart/form-data”。

2. 所上传文件的type必须是image/jpeg', 'image/gif' 或者'image/png'。
3. 图片文件要在客户端先压缩到640*640。
4. 图片名为logo；缩略图分为两种大小，logo_thumb为100*100的png，logo_thumb2为200*200的png。

例子：

访问：`curl -F "user_logo[img]=@firefox.png;type=image/png" /user_logos`

```
{"logo_thumb2":null, "logo_thumb1":null
,"id":6,
"logo":"",
"user_id":3,
"img_tmp":"/system/imgs/6/thumb/csky2.png?1340779488"}
```

用户上传图片更改为异步操作。图片上传完成后返回图片的id，但是此时图片还没有实际的阿里云链接。客户端上传完成后，可以用下面的“根据图片id获得用户相册里的图片”接口尝试获得实际的上传图片和缩略图。

10.2 根据图片id获得用户相册里的图片

GET	/user_logos/show		
参数	类型	必填	说明
id	字符串	必须	图片id
size	整数	可选	目前0代表原图；1代表thumb1缩略图，大小为100*100；2代表thumb2缩略图，大小为200*200。默认为1。

10.3 获得用户的头像

GET	/user_info/logo		
参数	类型	必填	说明
id	整数	必须	用户id
size	整数	可选	目前0代表原图；1代表thumb缩略图，大小为100*100；2代表thumb2缩略图，大小为200*200。默认为1。

该接口应该只有一种情况下被使用：在聊天室里有一个陌生的用户发了一条消息。此时根据他在聊天室的resource获得其用户id，然后调用本接口拿头像。

输出的Response Header中包含“Img_url”头，这个头就是查看用户信息时显示的用户头像的路径。

例如：“Img_url:/system/imgs/1/thumb2/1.png?1339649979”

注意：采用阿里云存储后，会输出302重定向，而不是直接给二进制流。

10.4 获得用户的图片列表

GET	/user_info/photos		
参数	类型	必填	说明
id	整数	必须	用户id

例子：

访问：/user_info/photos?id=1
返回值：

[
{"logo_thumb2":"/system/imgs/2/thumb2/2.png.png?1340616951","updated_at":"2012-06
"id":2,"logo":"/system/imgs/2/original/2.png.png?1340616951","img_file_size":8188
{"logo_thumb2":"/system/imgs/4/thumb2/2.png.png?1340617039","updated_at":"2012-06
"id":4,"logo":"/system/imgs/4/original/2.png.png?1340617039","img_file_size":8188
]

图片列表中的第一张图片就是用户的头像。

10.5 当前登录用户批量调整所有图片的位置

POST	/user_logos/change_all_position		
参数	类型	必填	说明
ids	逗号分割的整数	必须	所有图片的id按新的循序排列

说明：

比如原有图片的循序是2468，新的循序是8642，此时需要调用本接口，传递ids=8,6,4,2。此时服务器会更新所有图片的排序，而客户端只需要发送一个请求。

例子：

访问: `curl -F "ids=3,1,4" /user_logos/position`

10.6 当前登录用户删除图片

POST	/user_logos/delete		
参数	类型	必填	说明
id	整数	必须	图片的id

例子:

访问: `curl -F "id=2" /user_logos/delete`

输出:

```
{"deleted":"2"}
```

11 关注和粉丝

11.1 添加我关注的人

POST	/follows/create		
参数	类型	必填	说明
user_id	整数	必须	当前登录用户的id
follow_id	整数	必须	关注用户的id

注意：该请求必须是POST请求。

例子：

访问：
```
curl -b "_session_id=f03bef9371c119b6fcecbeefdaaac1b2"
     -d "user_id=2&follow_id=1" /follows
```

返回值：

```
{
"id":1,
"user_id":2,
"follow_id":1
}
```

11.2 删除我的关注的人

POST	/follows/delete		
参数	类型	必填	说明
user_id	整数	必须	当前登录用户的id
follow_id	整数	必须	关注用户的id

注意：该请求必须是POST请求。

例子：

访问：
```
curl -b "_session_id=f03bef9371c119b6fcecbeefdaaac1b2"
     -d "user_id=2&follow_id=3" /follows/delete
```

返回值：

```
{
"deleted":3
}
```

11.3 粉丝列表

GET	/follow_info/followers		
参数	类型	必填	说明
id	整数	必须	用户的id
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
name	字符串	可选	用户的名字
hash	整数	可选	是否返回hash，缺省值为0

例子：

访问：/follow_info/followers?id=2

返回值：

[
{"count":1}
{"data":
{"id":1,
"name":"name23",
"wb_uid":1884834632,
"gender":0,
"friend":true,
"follower":true,
"birthday":null,
"last":"1 hour 浙江科技产业大厦"}
}
]

其中count是符合条件的粉丝的总数。

获得用户基本信息里有两个关系字段：friend和follower。这里的关系都是针对我的，这里的我就是给定ID的用户。如果我关注了该用户，那么该用户就是我的friend；如果该用户关注了我，那么他就是我的follower。

这里返回的所有用户，其"follower"值一定为true。

[2013-2-20] 新增last，表示用户最后一次出现的时间和地点。

11.4 关注列表

本接口以后将不再支持。

GET	/follow_info/friends		
参数	类型	必填	说明
id	整数	必须	用户的id
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
name	字符串	可选	用户的名字
hash	整数	可选	是否返回hash，缺省值为0

返回的内容格式同粉丝列表一样

在1.5版本以前，本接口返回所有我关注的用户。在1.5版本以后，本接口返回所有我关注的且对方没有关注我的用户。双向关注的则由新增的good_friends接口提供。

11.5 双向好友列表

本接口以后将不再支持。

GET	/follow_info/good_friends		
参数	类型	必填	说明
id	整数	必须	用户的id
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
name	字符串	可选	用户的名字
hash	整数	可选	是否返回hash，缺省值为0

返回的内容格式同粉丝列表一样

这里返回的所有用户，都和当前登录用户互相关注。

11.6 批量获取所有我关注用户的信息

GET	/follow_info/friend_infos		
参数	类型	必填	说明
id	整数	必须	用户的id

返回的用户基本信息的一个数组。

11.7 关注列表ID

GET	/follow_info/friend_ids		
参数	类型	必填	说明
id	整数	必须	用户的id

11.8 双向好友列表ID

GET	/follow_info/good_friend_ids		
参数	类型	必填	说明
id	整数	必须	用户的id

11.9 批量获取关注者的位置信息

GET	/follow_info/friend_locs		
参数	类型	必填	说明
ids	字符串	必填	要获取的用户的id，多个id以英文”,”分割

例子：

访问：/follow_info/friend_ids?ids=502e6303421aa918ba000001,502e6303421aa918ba00000

返回值：

[{"id":"502e6303421aa918ba000001","last":"","time":""},
{"id":"502e6303421aa918ba000003","last":"","time":""}]

11.10 批量获取好友位置信息

GET	/follow_info/good_friend_locs		
参数	类型	必填	说明
ids	字符串	必填	要获取的用户的id，多个id以英文”,”分割

输出同上

11.11 批量获取粉丝位置信息

GET	/follow_info/fan_locs		
参数	类型	必填	说明
ids	字符串	必填	要获取的用户的id，多个id以英文”,”分割

输出同上

11.12 关于关注 / 好友接口的说明

1. 关注信息要在客户端单独建表，本地完整保存。好友信息是关注信息的子集合。
2. 当关注信息为空时，调用friend_infos接口初始化关注信息，然后调用good_friend_ids初始化好友信息。
3. 当用户手动刷新关注 / 好友列表时，调用friend_ids / good_friend_ids获取最新的关注 / 好友列表。对于新增的id，调用user_info/basic接口获得用户信息，对于减少的id，取消关注 / 好友标志。
4. 用户最后出现的地点信息，单独一个接口，每次按需获取。
5. 原有的friends和good_friends接口只是为了兼容性而存在，以后不要再调用。
6. 粉丝接口followers接口不变，每次只缓存最新的20条，手动刷新和加载更多。

12 黑名单

12.1 黑名单列表

GET	/blacklists		
参数	类型	必填	说明
id	整数	必须	用户的id
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20
name	字符串	可选	用户的名字
hash	整数	可选	是否返回hash，缺省值为0

返回的内容格式和好友列表一样。

12.2 添加黑名单

POST	/blacklists/create		
参数	类型	必填	说明
user_id	整数	必须	当前登录用户的id
block_id	整数	必须	加黑阻止的用户的id
report	整数	可选	是否同时举报被加黑的用户，0不举报1举报

返回值：

```
{
"id":1,
"user_id":2,
"report":false,
"block_id":1
}
```

12.3 删除黑名单

POST	/blacklists/delete		
参数	类型	必填	说明
user_id	整数	必须	当前登录用户的id
block_id	整数	必须	加黑用户的id

例子：

访问：`curl -b "_session_id=f03bef9371c119b6fcecbeefdaaac1b2" -d "user_id=2&block_id=3" /blacklists/delete`

返回值：

```
{
"deleted":3
}
```

13 XMPP协议接口

13.1 与Openfire聊天服务器接口

脸脸网的XMPP服务器采用的是Openfire。

1. 脸脸网的用户id加上”@dface.cn”就是openfire的jid，两个系统的密码一致。登录xmpp时的resource使用机器特定的字符串，以区分一个帐号在多台设备上登录。

2. 商家的id加上”@c.dface.cn”就是现场聊天室的jid。

3. 脸脸的用户名就是openfire的用户名，且也就是聊天时的用户名。Xmpp协议允许用户在加入聊天室时取一个名字，脸脸的聊天室应该忽略该名字。脸脸用户加入聊天室的时候，其jid格式为”roomid@c.dface.cn/uid”。

4. 脸脸中的关注和粉丝关系是单向的，和xmpp中的好友roster关系无关。

5. 只要用户登录，其xmpp协议中的presence就是在线，没有其它状态。

6. 聊天室中，消息id以ckn开头的消息代表是用户首次进入聊天室发送的通知类消息，客户端根据这个ckn头确定是否有新的人进入了聊天室。

7. 脸脸中的隐身也和xmpp中的隐身状态无关。

8. xmpp中的任意两个用户可以发送消息，不管对方是否隐身，只要对方未设置黑名单即可。如果对方未登录，那么就是离线消息。

所有的消息（包括单聊和聊天室），要保证消息id的唯一性，建议使用uuid生成消息的id。客户端发送的消息，由客户端生成这个id；服务器端发送的消息，由服务器端生成这个id。

目前，在发图的时候，存在同时收到两张一样的图片的消息的BUG。为了处理这个BUG，客户端在收到图片消息时，如果消息的图片ID一样，不重复显示图片。但是图片id以faq开头的图片消息不需要判断图片id重复的问题。

13.2 脸脸中的聊天用户的三种类型

1. 管理员，其jid固定为”502e6303421aa918ba000001@dface.cn”;
2. 商家，其jid为’s+商家id@dface.cn’;
3. 个人，其jid为’个人id@dface.cn’。

如果一个jid以字母s开头，那么它一定是商家。

13.3 摇一摇及其消息格式

目前，用户在聊天室摇一摇时，客户端发送“用户名摇了摇手机和大家say hello ”给服务器。这导致要分析摇一摇数据很困难，所以要为摇一摇定义特殊的消息格式。

摇一摇的消息格式为：

```
[摇一摇:$name]
```

其它客户端收到摇一摇的消息后，再将其转换为文字描述。

13.4 个人之间聊天的消息发送状态确认

XMPP协议的消息发送状态确认参考规范“XEP-0022: Message Events”。其定义了四种消息事件：Offline、Delivered、Displayed、Composing。Dface只需要其中的两种。

当接收端收到消息时，发送delivered确认消息，例如：

```
<message id="Kk98S-16" to="s6@dface.cn/ylt">
<x xmlns="jabber:x:event"><delivered/><id>purplea5e6669c</id></x>
</message>
```

当接收端展示消息时，发送displayed确认消息，例如：

```
<message id="Kk98S-17" to="s6@dface.cn/ylt">
<x xmlns="jabber:x:event"><displayed/><id>purplea5e6669c</id></x>
</message>
```

发送端根据从接收端获得的状态通知更改单条消息的发送状态。

只有类型是message,且body不为空的消息需要确认状态。照片提醒消息（发送人是"sphoto@dface.cn"）不需要确认状态。

当本地无法发送消息时（比如无网络、或者连接不上xmpp服务器），消息的状态为“发送失败”。发送失败的消息可以再次发送。

13.5 消息提醒

1. 当不在现场的界面收到现场聊天室的消息时，在现场按钮旁边加红点提醒;
2. 当收到发送人是”sphoto@dface.cn”发过来的单聊消息时，表明是有照片提醒消息。这时在我的照片等地方加红点提醒。
3. 状态确认类消息不需要提醒。
4. 其它个人之间聊天的消息以数字计数的方式提醒。

13.6 关于照片的类型

聊天相关的照片本来分为两种：单聊的照片和聊天室的照片。分别通过photos和photo2s接口获取。但是后来增加了问答类的图片消息和单聊时的群聊照片提醒/个人头像推送。所以为了区分这几种照片，规定如下：

1. 在聊天室收到id以“faq”开头的图片消息，说明是问答类的图片消息，查看调用photos/show接口；
2. 在单聊时收到id以“U”开头的图片消息，说明是个人单聊的图片消息，查看调用photo2s/show接口;
3. 在单聊时收到id以“Logo”开头的图片消息，说明是个人头像消息，查看调用photo2s/show接口;
4. 收到的其它图片消息都是聊天室的照片，包括在聊天室发的照片以及将该照片转发给个人。这类消息查看可以调用photos/show接口，也可以调用photo2s/show接口。但是这类照片还可以点评。点评时获取照片详情的接口则必须是photo/detail接口。

不管是在单聊和群聊时，图片消息仅仅根据图片id即可判断出类别。

14 在聊天室发图

聊天发图主要流程是：客户端选择（拍摄）一张照片，通过http上传到服务器。上传完成后在xmpp里发送一个特定格式的消息。其它客户端收到消息后获取图片显示并发送回执消息。

14.1 上传图片

POST	/photos/create		
参数	类型	必填	说明
photo[img]	file	必须	头像文件
photo[room]	字符串	必须	聊天室的id
photo[weibo]	0/1	必须	是否同步发送到微博
photo[qq]	0/1	可选	是否同步发送到QQ空间
photo[wx]	0/1/2/3	可选	分享到微信: 1个人,2朋友圈, 3都分享了
photo[desc]	字符串	可选	图像的说明文字
photo[t]	整数	可选	图片类型：1拍照；2选自相册
wbtoken	字符串	可选	如果分享到微博，该用户的微博token
qqtoken	字符串	可选	如果分享到QQ空间，该用户的QQtoken

注意：

1. 该请求必须是POST请求，enctype=”multipart/form-data”。
2. 所上传文件的type必须是image/jpeg’, ’image/gif’ 或者’image/png’。
3. 图片文件要在客户端先压缩到640*640。
4. 图片名为logo；缩略图分logo_thumb2为200*200的png。
5. 分享到新浪微博和QQ空间由服务器执行，客户端只需要传递标志位即可；而分享到微信则由客户端实现，然后报告服务器分享的结果。

例子：

访问：curl -F "photo[img]=@firefox.png;type=image/png"
/photos

{"_id":"509a1ffcbe4b1982a4000002",
"weibo":false,"room":"1",
"user_id":"502e61bfbe4b1921da000005",
"img_tmp":"20121107-1646-42114-4251/111.jpg",
"logo":"/uploads/tmp/20121107-1646-42114-4251/111.jpg",
"logo_thumb2":null,"id":"509a1ffcbe4b1982a4000002"}

当返回的响应中包含img_tmp字段，且logo_thumb2为null时，说明此时图片上传到了服务器，但是还未启动后台处理。后台异步处理图片，包括生成缩略图以及上传到阿里云。

14.2 发送消息

当图片发送完成后，服务器端发送一个xmpp消息，其中的body内容格式为：

[img:$id]$desc

比如上面的图片发送成功后，发送消息"[img:502fd10abe4b19ed48000002]"。其中$desc是可选的图片说明文字。

14.3 在聊天室收到消息后获取图片

接收端收到消息后，判断body的内容是否符合图片消息的格式。如果符合就调用下面的接口获取图片。

GET	/photos/show		
参数	类型	必填	说明
id	字符串	必须	图片id
size	整数	可选	目前0代表原图；2代表thumb2缩略图，大小为200*200。默认为2。

聊天室里发图不需要发送已读回执消息。在现场收到图片时，如果图片id以“faq”开头，不添加到照片墙上。18:02:24

14.4 在聊天室收点击图片获取图片详细信息

GET	/photos/detail		
参数	类型	必填	说明
id	字符串	必须	图片id

15 照片墙

15.1 删除聊天室图片

POST	/photos/delete		
参数	类型	必填	说明
id	字符串	必须	图片id

个人只能删除自己发布的照片。

15.2 对聊天室图片点赞

POST	/photos/like		
参数	类型	必填	说明
id	字符串	必须	图片id

15.3 对聊天室图片取消赞

POST	/photos/dislike		
参数	类型	必填	说明
id	字符串	必须	图片id

个人只能取消自己给的赞。

15.4 对聊天室图片点评

POST	/photos/comment		
参数	类型	必填	说明
id	字符串	必须	图片id
text	字符串	必须	评论的内容

返回的数据结构是评论的数组。id: 评论人的id name: 评论人的名字txt:评论的内容t:时间rid:被回复者的id rname: 被回复者的名字

15.5 对聊天室图片点评进行回复

POST	/photos/recomment		
参数	类型	必填	说明
id	字符串	必须	图片id
rid	字符串	必须	被回复者的id
text	字符串	必须	评论的内容

这里的rid是被回复者的id，而发布回复者的id是从session中自动获取的。

这里的回复其实是针对人的，如果一个人在一张图片下发布了多个评论，那么这里的回复不针对特定的那条评论。

15.6 对聊天室图片删除点评

POST	/photos/delcomment		
参数	类型	必填	说明
id	字符串	必须	图片id
text	字符串	必须	评论的内容

个人只能删除自己发的点评。

评论的删除是区分两种情况：删除 / 隐藏，评论的发布人可以删除，图片的发布人可以隐藏。对第三方来说都是删除。

如果一个评论是我发的，同时图也是我发的，那么此时对评论操作就是删除，不需要隐藏。

15.7 对聊天室图片隐藏点评

POST	/photos/hidecomment		
参数	类型	必填	说明
id	字符串	必须	图片id
uid	字符串	必须	评论的发布者id
text	字符串	必须	评论的内容

只有图片的发布者才能隐藏他人给自己图片发的点评。调用本接口的必须是图片的发布人。点评隐藏后只有点评的发布者一人能够看到。

15.8 我的照片墙

用户的照片墙由其本人在各个现场照片墙发布的照片组成。

GET	/photos/me		
参数	类型	必填	说明
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20

输出格式参见：6.4 。唯一不同点在于：商家照片墙中包含user_name，指明该照片是谁发的。用户照片墙中包含shop_name，指明该照片是在哪儿拍摄的。

15.9 我的评论

获得当前登录用户的评论的照片。

GET	/photos/my_comments		
参数	类型	必填	说明
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20

输出格式同上。

15.10 查看其它用户的照片墙

其它用户的照片墙由其该在各个现场照片墙发布并分享到微博 / qq的照片组成。没有分享过的图片看成只在该现场才能看到的图片。

GET	/photos/users		
参数	类型	必填	说明
uid	整数	必须	被查看用户的id
page	整数	可选	分页，缺省值为1
pcount	整数	可选	每页的数量，缺省为20

输出格式同上。

16 个人聊天时发图

16.1 个人聊天时上传图片

POST	/photo2s/create		
参数	类型	必填	说明
photo[img]	file	必须	头像文件
photo[to_uid]	字符串	必须	接收人的id
photo[t]	整数	可选	图片类型：1拍照；2选自相册

16.2 发送消息

当图片发送完成后，服务器端发送一个xmpp消息，其中的body内容格式为：

```
[img:$id]
```

16.3 在个人聊天时收到消息后获取图片

GET	/photo2s/show		
参数	类型	必填	说明
id	字符串	必须	图片id
size	整数	可选	目前0代表原图；2代表thumb2缩略图，大小为200*200。默认为2。

注意：聊天时发图，取消的100*100大小的缩略图。[2012-11-7]

16.4 消息状态

和普通的文字消息一样，接收者收到图片后发送状态更新消息（收到 / 已读）给发送者。发送者将状态显示在图片旁边。

17 个人聊天时发语音

17.1 个人聊天时上传语音

POST	/sound2s/create		
参数	类型	必填	说明
sound[img]	file	必须	语音文件
sound[to_uid]	字符串	必须	接收人的id
sound[sec]	整数	可选	语音长度：秒

17.2 发送消息

当语音发送完成后，服务器端发送一个xmpp消息，其中的body内容格式为：

```
[sound:$id]$sec
```

17.3 在个人聊天时收到消息后获取语音

GET	/sound2s/show		
参数	类型	必填	说明
id	字符串	必须	语音id

18 用户推荐

18.1 将个人推荐给好友

POST	/user_info/set_comment_name		
参数	类型	必填	说明
fid	字符串	必填	好友的用户id
uid	字符串	必填	被推荐的用户id
desc	字符串	可选	推荐时的文字

当前登录用户给自己的好友fid推荐一个用户uid。服务器端收到请求后，会发送下一节格式的Xmpp消息给好友。

返回值：`{"success" : 1}`

18.2 接收个人名片消息

在单聊时，可以接收个人名片消息。该消息的格式如下：

`[img:Logo$id:$uid]$desc`
其中`id`是个人头像的`id`，`uid`是个人的`id`.

该消息格式兼容单聊时的图片消息的格式。

19 优惠券

19.1 商家推送优惠券

当用户使用脸脸时，比如签到 / 摇一摇等，可以收到商家推送过来的优惠券。优惠券也是一个xmpp消息，其中的body内容格式为：

[优惠券:$name:$shop:$id:$time:$hint]

其中，name是优惠券的名称、shop是商家的名称，id是该优惠券的id，time是该张优惠券的下发时间。

hint是可选的，表示使用优惠券时的提示语言，比如“请输入此次的消费金额”。有hint的优惠券使用时要求用户输入内容才能使用优惠券。内容默认是数字。

优惠券id不重复。如果有重复的优惠券id时代表消息重发了，此时要忽略这条消息。

所有优惠券都统一显示在会话中的“优惠券”文件夹中。优惠券的消息状态和其他消息同样处理。

19.2 优惠券展示

优惠券以图片的方式展示，由服务器端生成该图片。当用户进入优惠券文件夹查看优惠券时，调用下面的接口获得优惠券的图片。

GET	/coupons/img		
参数	类型	必填	说明
id	字符串	必填	该优惠券的id
size	整数	可选	目前0代表原图(580, 224)；1代表缩略图，大小为(290,112)。默认为1。

19.3 批量获取优惠券详细信息

目前优惠券详细信息里只有经纬度这一个信息，以后可能扩展其它信息。

GET	/coupons/info		
参数	类型	必填	说明
ids	字符串	必填	要获取的优惠券的id，多个id以英文”,”分割

优惠券的经纬度主要用于优惠券的排序，以实现快速找到前所在的商家的优惠券。

优惠券的排序规则为：没有看过的新优惠券在前，看过的优惠在后，用过的在最后。没看过的优惠券，没有获得经纬度信息的距离算作0。同种状态的优惠券，距离当前位置小的在前，大的在后。

每次进入优惠券文件夹排序一次，在操作过程中不改变排序。比如查看 / 使用了优惠券，不改变这个优惠券的位置。只有在下次重新进入的时候，才改变位置。

例子：

访问：`/coupons/info?ids=518de040c90d8be6840000b5,5171d8d7c90d8b3c2900004a`
返回值：

```
[
{"id":"518de040c90d8be6840000b5","loc":[30.2425077887959,120.15320074395899]},
{"id":"5171d8d7c90d8b3c2900004a","loc":[30.282349,120.117012]}]
```

19.4 优惠券使用

目前，所有优惠券都只支持单次使用，使用后即过期。当用户确认使用优惠券时，调用如下的接口：

POST	/coupons/use		
参数	类型	必填	说明
id	字符串	必填	该优惠券的id
data	字符串	可选	优惠券使用时填写的信息

优惠券的使用要保证只能点击一次，使用后就灰掉，不需要等待服务器端的返回结果。如果网络不通或者服务器端失败，要以一定的策略重试。

当优惠券有hint字段时，使用时带上消费者输入的data。

在聊天室收到的id以“coupon”开头的消息时，点击该消息要能跳转到优惠券文件夹。

20 XMPP服务器的Web接口

20.1 获得聊天室的消息历史记录

GET	http://xmpp_ip:5280/api/gchat		
参数	类型	必填	说明
room	字符串	必填	聊天室的房间id

20.2 获得个人聊天的历史记录

GET	http://xmpp_ip:5280/api/chat		
参数	类型	必填	说明
uid	字符串	必填	用户的id

20.3 获得两人之间的聊天历史记录

GET	http://xmpp_ip:5280/api/chat2		
参数	类型	必填	说明
uid1	字符串	必填	用户1的id
uid2	字符串	必填	用户2的id

20.4 其它内部接口

仅用于服务器端的后台内部调用。

20.4.1 以给定用户身份给指定的聊天室发送消息

POST	http://xmpp_ip:5280/api/room		
参数	类型	必填	说明
roomid	字符串	必填	聊天室的房间id
message	字符串	必填	需要发送的消息
uid	字符串	必填	发送此消息的用户id

这条消息会发送给聊天室的所有人，比如用户进入聊天室时的打招呼。如果是给聊天室的某个人发消息，比如优惠券中奖 / 公告消息，则通过rest接口发送groupchat类型的消息。

20.4.2 用户屏蔽通讯

POST	http://xmpp_ip:5280/api/block		
参数	类型	必填	说明
uid	字符串	必填	发起屏蔽请求的用户的id
bid	字符串	必填	被屏蔽的用户的id

20.4.3 用户解除屏蔽

POST	http://xmpp_ip:5280/api/unblock		
参数	类型	必填	说明
uid	字符串	必填	发起解除屏蔽请求的用户的id
bid	字符串	必填	被屏蔽的用户的id

20.4.4 杀掉用户会话

POST	http://xmpp_ip:5280/api/kill		
参数	类型	必填	说明
user	字符串	必填	被封杀的用户的id

20.4.5 禁止用户在聊天室发言

POST	http://xmpp_ip:5280/api/room_ban		
参数	类型	必填	说明
roomid	字符串	必填	聊天室id
uid	字符串	必填	被封杀的用户的id

20.4.6 解除禁止用户在聊天室发言

POST	http://xmpp_ip:5280/api/room_unban		
参数	类型	必填	说明
roomid	字符串	必填	聊天室id
uid	字符串	必填	被封杀的用户的id

20.4.7 发送任意XMPP消息

POST	http://xmpp_ip:5280/rest		
参数	类型	必填	说明

通过POST提交任意类型的XMPP文本消息。

21 关于Push消息

21.1 登录Xmpp服务器时，报告设备的Push消息token

登录Xmpp服务器，发送的密码格式变更为：

“用户密码”+《状态码》+token

状态码为：1代表ios开发设备，2代表实际的发布设备。
token是一个64位的字符串。

比如某用户的密码是“c53b2f16a24c61d9”，
token是“ea6e4f05b48f4a057816956e567c6feacf14ee28cbbbab67993e263c3dfa2c27”，

目前是进行开发测试，那么登录Xmpp服务器时的密码为：
“c53b2f16a24c61d91ea6e4f05b48f4a057816956e567c6feacf14ee28cbbbab67993e263c3dfa2c2

服务器端解析出状态码和token并保存。

21.2 退出时，报告设备的Push消息token

用户退出的API接口新增pushtoken参数，以取消Push消息token绑定。

21.3 Push消息的发送

当有离线消息产生时，xmpp服务器获得用户的状态码以决定给那个Apple Push服务器发消息，以及获得和该用户绑定的token。一个用户只能绑定一个token。

22 帮助链接

GET	/help/index		
参数	类型	必填	说明
ver	字符串	必须	软件版本
os	浮点数	必须	操作系统

本接口需要通过浏览器以URL的方式直接调用。

23 关于程序各页面之间的切换规则

程序的首页是选择现场页面。现场则是一个聊天室页面。
1、程序干净启动时，要判断用户是否处于登录状态。

1.1如果已经登录，进入定位页面，并从服务器端获取新的商家列表。

1.2如果没有登录，进入登录页面。

2、程序从睡眠状态被激活时，要判断距离上次的运行时间。

2.1 如果不足12个小时，上次被任务切换时停留在那个页面就仍然停留在该页面。

2.2 如果超过12个小时，进入定位页面，并从服务器端获取新的商家列表。

4、从其它tab点击现场时，判断用户是否摇进过现场

4.1 如果摇进过某个现场，直接进入该现场聊天室。

4.2 如果没有摇，那么就是定位页面。