

脸脸网服务器与手机端的开发接口API

YXY

2014 年 4 月 22 日

目录

| | | |
|----------|--------------------------|-----------|
| 1 | 初始化与升级 | 10 |
| 1.1 | 初始化 | 10 |
| 1.2 | 获得服务器的时间 | 11 |
| 1.3 | 获得最新的升级信息 | 12 |
| 2 | 调试时发送Xmpp断言信息 | 13 |
| 3 | 用户帐号系统 | 14 |
| 3.1 | 新浪微博相关接口 | 14 |
| 3.1.1 | 网页登录接口: oauth2 | 14 |
| 3.1.2 | 客户端登录接口: xauth | 16 |
| 3.1.3 | 客户端SSO接口 | 17 |
| 3.1.4 | 解除新浪微博绑定 | 17 |
| 3.2 | QQ相关接口 | 18 |
| 3.2.1 | QQ登录接口 | 18 |
| 3.2.2 | 解除QQ绑定 | 18 |
| 3.3 | 手机注册接口 | 19 |
| 3.3.1 | 手机号码获取验证码 | 19 |

| | |
|---------------------------------|-----------|
| 目录 | 2 |
| 3.3.2 手机号码注册 | 19 |
| 3.3.3 忘记密码 | 20 |
| 3.3.4 修改密码 | 20 |
| 3.3.5 手机号码登录 | 20 |
| 3.3.6 绑定手机号码 | 21 |
| 3.3.7 绑定手机后设置密码 | 21 |
| 3.3.8 解除手机号码绑定 | 21 |
| 3.3.9 上传手机通讯录 | 22 |
| 3.4 短信上传注册接口 | 22 |
| 3.4.1 获取短信发送号码和内容 | 22 |
| 3.4.2 用户是否短信注册通过 | 23 |
| 3.5 退出 | 23 |
| 3.6 登录与绑定的状态判断 | 23 |
| 3.7 关于首次登录的判断 | 25 |
| 4 QQ登录用户看他人微博 | 25 |
| 5 首次使用脸脸时发送分享信息 | 25 |
| 6 现场与附近 | 26 |
| 6.1 首页：我最近去过的四个地点 | 26 |
| 6.2 定位：根据经纬度获得可能的现场商家 | 26 |
| 6.3 搜索：根据经纬度获得附近的商家 | 29 |
| 6.4 搜索：热点商家 | 30 |
| 6.5 搜索：我的地点 | 30 |
| 6.6 根据经纬度获得附近的热点用户 | 31 |
| 6.7 获得附近的用户 | 31 |
| 6.8 地点报错 | 32 |
| 7 根据商家ID获得商家的信息 | 32 |
| 7.1 根据商家ID获得商家的基本信息 | 32 |

| | |
|-------------------------------------|-----------|
| 目录 | 3 |
| 7.2 在聊天室获得商家的信息 | 33 |
| 7.3 根据商家ID获得商家的图片墙 | 35 |
| 7.4 根据商家ID获得当前在该商家的所有用户 | 36 |
| 7.5 根据商家ID获得最新的6个用户 | 38 |
| 7.6 根据商家ID获得商家的聊天室历史 | 38 |
| 7.7 根据商家ID获得商家的顶部5张banner | 39 |
| 7.8 根据商家ID获得商家的子地点信息 | 39 |
| 8 签到与足迹 | 40 |
| 8.1 进入现场签到 | 40 |
| 8.2 输入商家的全称创建商家并签到 | 41 |
| 8.3 签到时输入地点名称查找地点 | 42 |
| 8.4 获得当前登录用户的足迹 | 43 |
| 8.5 删除当前登录用户的一个签到足迹 | 45 |
| 9 群组与认证 | 45 |
| 9.1 加入群组的标示 | 45 |
| 9.2 群组的认证 | 46 |
| 10 用户基本信息 | 46 |
| 10.1 获得用户基本信息 | 46 |
| 10.2 获得用户和当前登录用户的关系 | 47 |
| 10.3 获得用户最后出现的位置信息 | 48 |
| 10.4 获得用户的地主地点列表 | 48 |
| 10.5 获得当前登录用户基本信息 | 49 |
| 10.6 设置当前登录用户基本信息 | 51 |
| 10.7 设置其它用户的备注名 | 52 |
| 10.8 获得当前登录用户设置的所有备注名 | 53 |
| 10.9 根据用户名查找脸脸用户 | 53 |

| | |
|----------------------------------|-----------|
| 目录 | 4 |
| 11 用户相册与头像 | 54 |
| 11.1 当前登录用户上传图片 | 54 |
| 11.2 获得用户的图片列表 | 55 |
| 11.3 当前登录用户批量调整所有图片的位置 | 56 |
| 11.4 当前登录用户删除图片 | 57 |
| 12 关注和粉丝 | 58 |
| 12.1 添加我关注的人 | 58 |
| 12.2 删除我的关注的人 | 59 |
| 12.3 粉丝列表 | 60 |
| 12.4 关注列表 | 62 |
| 12.5 双向好友列表 | 62 |
| 12.6 批量获取所有我关注用户的信息 | 63 |
| 12.7 关注列表ID | 63 |
| 12.8 双向好友列表ID | 63 |
| 12.9 批量获取关注者的位置信息 | 64 |
| 12.10 批量获取好友位置信息 | 64 |
| 12.11 批量获取粉丝位置信息 | 65 |
| 12.12 关于关注 / 好友接口的说明 | 65 |
| 13 黑名单 | 66 |
| 13.1 黑名单列表 | 66 |
| 13.2 添加黑名单 | 66 |
| 13.3 删除黑名单 | 67 |
| 14 我的地点 | 68 |
| 14.1 获得我的地点 | 68 |
| 14.2 设置我的地点 | 68 |
| 14.3 设置我的地点时选择地点 | 69 |

| | |
|--------------------------|-----------|
| 目录 | 5 |
| 15 XMPP协议接口 | 69 |
| 15.1 与Xmpp协议的不同 | 69 |
| 15.2 脸脸Xmpp的JID | 69 |
| 15.3 脸脸中的聊天用户的三种类型 | 70 |
| 15.4 消息ID | 70 |
| 15.5 聊天室的特定格式消息 | 70 |
| 15.6 摇一摇及其消息格式 | 71 |
| 15.7 个人之间聊天的消息发送状态确认 | 71 |
| 15.8 消息提醒 | 72 |
| 15.9 关于照片的类型 | 73 |
| 16 好友动态 | 73 |
| 16.1 好友动态消息 | 73 |
| 16.2 好友动态中的评论消息 | 74 |
| 16.3 对方加我时的单聊提醒消息 | 74 |
| 16.4 谁访问过我的主页的消息 | 74 |
| 17 在聊天室发图 | 74 |
| 17.1 上传图片 | 75 |
| 17.2 多图上传时，非第一张图的上传 | 76 |
| 17.2.1 获取七牛云存储的图片上传token | 76 |
| 17.2.2 上传图片 | 76 |
| 17.3 发送消息 | 77 |
| 17.4 在聊天室收到消息后获取图片 | 78 |
| 17.5 在聊天室收点击图片获取图片详细信息 | 78 |
| 18 照片墙 | 79 |
| 18.1 删除聊天室图片 | 79 |
| 18.2 删除聊天室消息 | 79 |
| 18.3 对聊天室图片点赞 | 80 |

| | |
|------------------------|-----------|
| 目录 | 6 |
| 18.4 对聊天室图片取消赞 | 80 |
| 18.5 对聊天室图片点评 | 80 |
| 18.6 对聊天室图片点评进行回复 | 81 |
| 18.7 对聊天室图片删除点评 | 82 |
| 18.8 对聊天室图片隐藏点评 | 82 |
| 18.9 我的照片墙 | 82 |
| 18.10 我的评论 | 83 |
| 18.11 查看其它用户的照片墙 | 83 |
| 18.12 其它用户分享到微信 | 84 |
| 19 个人聊天时发图 | 85 |
| 19.1 个人聊天时上传图片 | 85 |
| 19.2 发送消息 | 85 |
| 19.3 在个人聊天时收到消息后获取图片 | 85 |
| 19.4 消息状态 | 86 |
| 19.5 收到个人聊天图片后转发给好友 | 86 |
| 20 个人聊天时发语音 | 86 |
| 20.1 获取七牛云存储的文件上传token | 86 |
| 20.2 上传文件 | 87 |
| 20.3 上传成功后的回调 | 87 |
| 20.4 发送消息 | 88 |
| 20.5 在个人聊天时收到消息后获取语音 | 89 |
| 21 用户推荐 | 89 |
| 21.1 将个人推荐给好友 | 89 |
| 21.2 接收个人名片消息 | 89 |
| 22 优惠券 | 90 |
| 22.1 商家推送优惠券 | 90 |
| 22.2 获取优惠券列表 | 90 |

| | |
|--|-----------|
| 目录 | 7 |
| 22.3 优惠券展示 | 91 |
| 22.4 优惠券使用 | 92 |
| 22.5 优惠券删除 | 93 |
| 22.6 优惠券批量使用 | 93 |
| 22.7 优惠券批量删除 | 93 |
| 23 XMPP服务器的Web接口 | 94 |
| 23.1 获得聊天室的消息历史记录 | 94 |
| 23.2 获得聊天室的消息历史记录2 | 94 |
| 23.3 获得个人聊天的历史记录 | 94 |
| 23.4 获得两人之间的聊天历史记录 | 95 |
| 23.5 其它内部接口 | 95 |
| 23.5.1 以给定用户身份给指定的聊天室发送消息 | 95 |
| 23.5.2 用户屏蔽通讯 | 96 |
| 23.5.3 用户解除屏蔽 | 96 |
| 23.5.4 获取用户屏蔽列表 | 96 |
| 23.5.5 杀掉用户会话 | 96 |
| 23.5.6 禁止用户在聊天室发言 | 97 |
| 23.5.7 解除禁止用户在聊天室发言 | 97 |
| 23.5.8 发送任意XMPP消息 | 97 |
| 24 关于Push消息 | 97 |
| 24.1 登录Xmpp服务器时，报告设备的Push消息token | 97 |
| 24.2 退出时，报告设备的Push消息token | 98 |
| 24.3 Push消息的发送 | 98 |
| 25 帮助链接 | 99 |
| 26 自定义菜单 | 99 |
| 26.1 获取菜单 | 99 |
| 26.2 点击菜单 | 101 |

| | |
|-------------------------------------|------------|
| 27 脸脸WEB应用嵌入接口 | 101 |
| 27.1 加载和退出 | 101 |
| 27.2 脸脸客户端Js API | 102 |
| 27.2.1 隐藏脸脸右上角的选择项按钮 | 102 |
| 27.2.2 设置脸脸的标题 | 102 |
| 27.2.3 获得当前用户名 | 102 |
| 27.2.4 获得当前用户名的session id | 102 |
| 27.2.5 获得当前用户id和商家id | 102 |
| 27.2.6 获得当前经纬度 | 102 |
| 27.2.7 分享到微信好友 | 102 |
| 27.2.8 分享到微信朋友圈 | 103 |
| 28 脸脸应用内跳转规范 | 103 |
| 29 脸脸聊天图片上传时的压缩规则 | 104 |
| 30 坐标转换接口 | 105 |
| 30.1 实际经纬度坐标转GCJ坐标 | 105 |
| 30.2 实际经纬度坐标转Baidu坐标 | 105 |
| 30.3 GCJ坐标转实际坐标 | 105 |
| 30.4 百度坐标转实际经纬度坐标 | 106 |
| 30.5 百度坐标转GCJ坐标 | 106 |
| 31 关于程序各页面之间的切换规则 | 106 |

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 |
| bssid | 字符串 | 可选 | 所连接的wifi的bssid |
| ver | 字符串 | 可选 | 软件版本,如"2.1.3" |
| city | 数字 | 可选 | 如果为1代表要在响应的header中的location字段中包含根据ip获得的城市信息 |
| channel | 字符串 | 可选 | 代表渠道号 |

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。当以ip调用失败时，也就是42.121.79.210服务器当机了的时候，改用域名的方式调用init接口，也就是“https://www.dface.cn/init/init”。如果使用域名的方式也失败，且上次正确的配置也失败，那么最后使用”www.dface.cn”和”xmpp.dface.cn”来访问ip和xmpp。

调用init接口不能阻塞应用的启动和后续网络请求。当init调用未返回时，直接使用上次正确的配置连接web和xmpp服务器。

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 [3.3.9] 类似。

返回值：

```
{  
  "ip": "42.121.79.210",  
  "xmpp": "42.121.79.210",  
  "ver": 2.2  
  "version": "2.2.1"  
}
```

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，是一个浮点数。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。

[2013-8-1] 新增**version**，表示脸脸当前线上的最新的具体版本，是一个字符串。当用户手动检查升级时，可以使用这个字段。

[2014-4-21] 新增**time**，表示脸脸服务器的当前时间，从1970年开始到现在的秒数。

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。

1.2 获得服务器的时间

用于安卓手机校准时间。

| | | | |
|------|------------|----|--------|
| GET | /init/time | | |
| 参数 | 类型 | 必填 | 说明 |
| time | 数字 | 可选 | 手机本地时间 |

返回值：time: long, long表示从1970年开始到现在的秒数。

1.3 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 | /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 调试时发送Xmpp断言信息

客户端在一些关键逻辑部分，可以加入Xmpp信息发送。

| | | | |
|------|-------------|----|--------|
| GET | /init/debug | | |
| 参数 | 类型 | 必填 | 说明 |
| uid1 | 字符串 | 必须 | 发送者id |
| uid2 | 字符串 | 必须 | 接受者id |
| str | 字符串 | 必须 | 要发送的内容 |

其中，接受者id必须是开发者在脸脸的uid。

本接口的调用场景如：

```
1. if(必须成立的条件) {  
    ...  
    }else{  
        发送Xmpp信息  
    }  
  
2. try {  
    ...  
    }catch{  
        发送Xmpp信息  
    }  
  
3. 函数入口处的参数检查  
    if(arg==null) 发送Xmpp信息
```

3 用户帐号系统

3.1 新浪微博相关接口

3.1.1 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。

3.1.2 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。

3.1.3 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 |

当手机客户端安装有新浪官方微博客户端，使用微博SSO登录，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。

3.1.4 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。

3.2 QQ相关接口

3.2.1 QQ登录接口

| | | | |
|--------------|-------------------|----|--|
| 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。

3.2.2 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。

3.3 手机注册接口

3.3.1 手机号码获取验证码

| | | | |
|-------|-------------|----|---------------------|
| POST | /phone/init | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| type | 整数 | 可选 | 1代表注册，2代表绑定，3代表忘记密码 |

调用成功返回”code”:HASH后的验证码，如果有错误，错误信息保存在error中。同时验证码通过短信的方式发送到使用者的手机。

手机号码注册 / 忘记密码 / 绑定手机号码都要先调用本接口获取验证码。

[3013-08-11] 增加type参数，表示请求手机验证码的类型。同时取消客户端对验证码的HASH是否正确的判断

3.3.2 手机号码注册

| | | | |
|----------|-----------------|----|---------|
| POST | /phone/register | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| code | 字符串 | 必须 | 验证码 |
| password | 字符串 | 选填 | 用户设置的密码 |

调用成功返回生成的用户id，如果有错误，错误信息保存在error中。

[3014-4-2] 手机号码注册时，密码变为选填。

3.3.3 忘记密码

| | | | |
|----------|------------------------|----|---------|
| POST | /phone/forgot_password | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| code | 字符串 | 必须 | 验证码 |
| password | 字符串 | 必须 | 用户设置的密码 |

本接口和手机号码注册接口参数一样，要求相反。手机号码注册要求该手机号码不存在，忘记密码要求该手机号码已注册。

3.3.4 修改密码

| | | | |
|----------|------------------------|----|---------|
| POST | /phone/change_password | | |
| 参数 | 类型 | 必填 | 说明 |
| oldpass | 字符串 | 必须 | 原来的密码 |
| password | 字符串 | 必须 | 用户设置的密码 |

3.3.5 手机号码登录

| | | | |
|----------|--------------|----|---------|
| POST | /phone/login | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| password | 字符串 | 必须 | 用户设置的密码 |

调用成功返回用户的信息，格式和user_infoget_self相同。如果有错误，错误信息保存在error中。

3.3.6 绑定手机号码

| | | | |
|-------|-------------|----|---------|
| POST | /phone/bind | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| code | 字符串 | 必须 | 验证码 |

绑定手机号码要先调用phoneinit获取验证码。调用成功返回bind:true，如果有错误，错误信息保存在error中。

[2013-8-15] 取消参数password, 绑定手机号码时可以不用输入手机号码。

3.3.7 绑定手机后设置密码

| | | | |
|----------|---------------------|----|---------|
| POST | /phone/set_password | | |
| 参数 | 类型 | 必填 | 说明 |
| password | 字符串 | 必须 | 用户设置的密码 |
| phone | 字符串 | 必须 | 用户的手机号码 |

如果已经设置了密码后要修改密码，不能调用本接口。

3.3.8 解除手机号码绑定

| | | | |
|-------|---------------|----|---------|
| POST | /phone/unbind | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |

至少有一种以上的绑定关系才能解绑。

3.3.9 上传手机通讯录

| | | | |
|-------|----------------------------|----|---------|
| POST | /phone/upload_address_list | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |
| list | json字符串 | 必须 | 通讯录 |

`list`是一个json字符串表示的数组，其中每一个条目是一个hash，格式如下：

```
[{"id":323,"number":"(0571)28996083","name":"***"},...]
```

返回值为 “{imported: 导入的通讯录的条数}”。

3.4 短信上传注册接口

3.4.1 获取短信发送号码和内容

| | | | |
|-------|---------------|----|---------|
| GET | /phone/sms_up | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |

注意：本接口只支持自己手机号码的注册，也就是发送短信的号码必须是接口中填写的号码。

返回值为 “{phone: 发送到的手机号码, txt:要发送的内容}”。

然后客户端调用本机器的短信发送程序，把目标号码和短信内容设置好。

3.4.2 用户是否短信注册通过

当用户点击本机的短信发送按钮后，调用本接口。

| | | | |
|-------|--------------------|----|---------|
| GET | /phone/do_register | | |
| 参数 | 类型 | 必填 | 说明 |
| phone | 字符串 | 必须 | 用户的手机号码 |

注册成功时，本接口的返回值和register接口一致。短信未收到则返回error信息。

3.5 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。

3.6 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或者手机号码中的一个。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。

3.7 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）都会用户在首次登录时返回字段：newuser:1。当有newuser时，走注册流程。

4 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

5 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。

6 现场与附近

6.1 首页：我最近去过的四个地点

| | | | |
|---------|---------------|----|------|
| GET | /aroundme/my4 | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必须 | 用户ID |

提供最后签到时间，{time: 字符串时间，timei: 整数时间}

6.2 定位：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30条数据，最少返回3-5条。

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。

[2013-6-20] 新增city字段，商家所在的城市名称。只有排名第一的商家才返回city字段。city字段的用途是用于显示热点的城市名称。

6.3 搜索：根据经纬度获得附近的商家

| | | | |
|----------|--------------|----|-----------------------|
| 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. 餐饮
- 2. 酒店
- 3. 酒吧●咖啡●茶馆
- 4. 休闲●娱乐●KTV
- 5. 楼宇●社区●学校
- 6. 购物●广场●其它

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。

6.4 搜索：热点商家

| | | | |
|----------|---------------|----|-----------------------|
| GET | /aroundme/hot | | |
| 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 |
| baidu | 整数 | 可选 | 如果是百度坐标，则baidu=1，否则忽略 |

6.5 搜索：我的地点

| | | | |
|---------|--------------------|----|------|
| GET | /aroundme/my_shops | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必须 | 用户ID |

6.6 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 |
| gender | 整数 | 可选 | gender=1代表查看男性，gender=2代表查看女性。不传默认查看异性 |

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。

6.7 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。

6.8 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。

7 根据商家ID获得商家的信息

7.1 根据商家ID获得商家的基本信息

| | | | |
|-----|-------------|----|-------|
| GET | /shop/basic | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 整数 | 必须 | 商家的ID |

本接口用于不在现场聊天室获得商家基本信息。

[2014-04-17] 增加五个参数: 是否签约sign, 地址addr, 电话tel, 商
标logo, 子地点sub)

7.2 在聊天室获得商家的信息

| | | | |
|-----|------------|----|-------|
| 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。

[2013-12-05] 新增字段**filters**，表示图片合成滤镜。其中**filters**是图片URL的数组，**filter_name**是图片合成的名称，**filter_size**是图片合成后图片的大小。

[2014-01-03] **photos**中排名第一且od值大于0的图片，显示为（包含赞/3个评论）的卡片。

7.3 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.jpg",  
  "id": "509a061fbe4b197297000002"}]
```

7.4 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。

7.5 根据商家ID获得最新的6个用户

| | | | |
|-------|--------------|----|---------------------------|
| GET | /shop/user6s | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 整数 | 必须 | 商家的ID |
| total | 整数 | 可选 | 是否返回商家总人数，新客户端都要设置total=1 |

只有当商家用户数大于3时，这个接口才返回数据；否则返回空数组。
客户端自行控制是显示5个还是6个用户头像。

当total=1时，返回值的格式为：

```
{total:总人数, users:[用户信息,...]}
```

7.6 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5 |

skip的大小应该是聊天室当前最新显示的消息的总数(但是不包括评论通知消息、问答消息)。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：

以前每次进聊天室，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>
```

现在全部用本接口获得历史消息。

[[{"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]。

7.7 根据商家ID获得商家的顶部5张banner

| | | | |
|-----|---------------|----|-------|
| GET | /shop/banners | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 整数 | 必须 | 商家的ID |

7.8 根据商家ID获得商家的子地点信息

| | | | |
|-----|------------|----|-------|
| GET | /shop/subs | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 整数 | 必须 | 商家的ID |

8 签到与足迹

8.1 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。

8.2 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 |
| t | 整数 | 可选 | 商家的内部类别 |
| addr | 字符串 | 可选 | 商家的地址 |
| tel | 字符串 | 可选 | 商家的联系电话 |
| 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 |
| 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。

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。

[2014-04-17] 增加三个参数：商家内部的整数类型t，地址addr，电话tel，取消一个参数type。

商家内部的整数类型t的值如下：0 活动1 酒吧2 咖啡3 茶馆4 餐饮5 酒店6 休闲娱乐7 运动8 购物9 广场10 写字楼11 住宅12 学校13 交通14 大型医院15 景点51 美容美发52 公司53 其他

例子：

返回值：

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

8.3 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 | 商家的名称 |

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。

[2013-09-27] 新增参数distance，表示该地点的距离。

[2013-12-21] 在响应的头部增加属性“GCJOFFSET”，内容格式“纬度偏移,经度偏移”。用于在国内的谷歌地图、soso地图上显示地点时，进行坐标偏移。这些地图都使用GCJ坐标系；而脸脸中所有的坐标都是真实坐标。要获得GCJ坐标，将真实坐标的经纬度加上GCJOFFSET的经纬度即可。如果没有GCJOFFSET，则忽略。

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}]
```

8.4 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 } ]  
}
```

8.5 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。

9 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）。

9.1 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。

9.2 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。

10 用户基本信息

10.1 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": ""}
```

如果是微博认证用户，有如下信息

 :wb_v, #是否是微博认证用户

 :wb_vs # 微博认证说明

10.2 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}
```

10.3 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, 表示用户最后在脸上的时间和地点, 一个以空格分割的字符串。

10.4 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": 120.11969, "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": 121.586937, "address": "", "phone": "", "id": 21830801, "user": 0, "male": 0, "female": 0}]
```

10.5 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

[2013-06-28] 新增wb_hidden, 1代表对他人隐藏自己的微博, 2代表解除微博绑定.

[2013-08-26] 新增psd字段, 代表设置过密码; 新增pmatch字段代表启用了通讯录匹配.

10.6 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 |
| wb_hidden | 数字 | 选填 | 0: 不隐藏微博, 1代表对他人隐藏自己的微博 |
| no_push | 数字 | 选填 | 0: 默认可以push, 1代表不接收push消息提醒。 只有android客户端需要设置这个参数 |

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  
}
```

10.7 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}`

10.8 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。

10.9 根据用户名查找脸脸用户

| | | | |
|--------|-------------------|----|-------------------------|
| GET | /user_info/search | | |
| 参数 | 类型 | 必填 | 说明 |
| name | 字符串 | 必须 | 用户的脸脸名字、微博 / qq昵称、手机号码等 |
| 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 |

根据用户的脸脸名字、微博 / qq昵称、手机号码等查找脸脸用户。其中手机号码是精确查询，其它是模糊查询。

返回用户的基本信息，该用户和当前用户的关系信息，距离等。如果是黑名单用户，则有“black:1”。

```
[{"name":"袁乐天","signature":"","gender":1,"birthday":"","jobtype":null,"pcount"
"friend":false,"follower":false,"distance":99999999}]
```

11 用户相册与头像

用户相册中的第一张图片就是头像。

11.1 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。

11.2 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":"2012-06-06 11:11:11",
"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":"2012-06-06 11:11:11",
"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。

11.3 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`

11.4 当前登录用户删除图片

| POST | <code>/user_logos/delete</code> | | |
|------|---------------------------------|----|-------|
| 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"}
```

12 关注和粉丝

12.1 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
}
```

12.2 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  
}
```

12.3 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。

12.4 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。

12.5 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。

12.6 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。

12.7 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 |

12.8 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 |

12.9 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": ""}]
```

12.10 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

12.11 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

12.12 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。

13 黑名单

13.1 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。

13.2 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
}
```

13.3 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
}
```

14 我的地点

14.1 获得我的地点

| | | | |
|---------|------------------|----|------|
| GET | /my_loc/get_locs | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 整数 | 必须 | 用户id |

返回值:

```
"{"home_id":10434569,"home_name":"某小区"}"
{"school_id":10434569,"school_name":"某学校"}"
{"building_id":10434569,"building_name":"某楼宇"}"
```

14.2 设置我的地点

| | | | |
|---------|-----------------|----|----------------------------|
| POST | /my_loc/set_loc | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 整数 | 必须 | 用户id |
| type | 字符串 | 必须 | “home/school/building” 三选一 |
| sid | 整数 | 必须 | 地点id |

执行成功返回ok:1。

14.3 设置我的地点时选择地点

| POST | /my_loc/select_loc | | |
|---------|--------------------|----|----------------------------|
| 参数 | 类型 | 必填 | 说明 |
| user_id | 整数 | 必须 | 用户id |
| type | 字符串 | 必须 | “home/school/building” 三选一 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |
| baidu | 整数 | 可选 | 如果是百度坐标，则baidu=1，否则忽略 |
| name | 字符串 | 可选 | 名字 |

15 XMPP协议接口

15.1 与Xmpp协议的不同

1. 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。
2. 只要用户登录，其xmpp协议中的presence就是在线，没有其它状态。
3. 脸脸中的隐身也和xmpp中的隐身状态无关。
4. 脸脸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。

15.2 脸脸Xmpp的JID

1. 脸脸网的用户id加上”@dface.cn”就是Xmpp的jid。登录时的resource格式为: '操作系统脸脸版本-月日'，其中操作系统为i或者a。

2. 商家的id加上"@c.dface.cn"就是现场聊天室的jid。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"。

例如：个人登录Xmpp服务器后的完整JID如“502e6303421aa918ba000005@c.dface.cn/i2.3.4-0927-09:34”，进入聊天室的完整JID如“1234@c.dface.cn/502e6303421aa918ba000005”。

15.3 脸脸中的聊天用户的三种类型

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

如果一个jid以字母s开头，那么它一定是商家。

15.4 消息ID

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。

15.5 聊天室的特定格式消息

1. 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。

2. 聊天室中，消息id以FAQ开头的是问答消息。
3. 聊天室中，消息id以COMMENT开头的是图片的评论消息。聊天室评论消息带楼数的格式：[img:id]楼数:评论内容。
4. 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。

15.6 摇一摇及其消息格式

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：“\$name摇了摇手机和大家say hello~”

15.7 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/y!t">  
<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/y!t">  
<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”。发送失败的消息让用户手动重发。没有收到回执的消息,则自动重发。

对于服务器端发送的消息,比如图片/语音/名片等,客户端先保存消息到数据库,而服务器端发消息的ID客户端不知道。所以当客户端发送回执时。

15.8 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。[2.3以上版本,此功能取消]。
3. 已发/已读类状态确认类消息不需要提醒。
4. 好友动态消息以红点的方式提醒。
5. 好友评论消息以数字计数的方式提醒。
6. 其它个人之间聊天的消息以数字计数的方式提醒。

15.9 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。

16 好友动态

16.1 好友动态消息

所有消息id以“FEED”开头的，都属于好友动态消息。好友动态消息以红点的方式提醒，不计数。好友动态消息增加地点，IOS通过属性实现，Android通过扩展实现，如下：

```
"<x xmlns='dface.shop' SID='#{shop.id}' SNAME='#{shop.name}' ></x>"
```

如果是多图的动力，那么在图片的描述信息开始是数字加“:”开始，在扩展属性中有“dface.thumb2s”字段，内容是第二张图到最后一张图的缩略图的url，以“,”号分割。

16.2 好友动态中的评论消息

所有消息id以"COMMENT"开头的，都属于好友评论消息。评论消息以数字计数。

评论消息提醒的消息id格式："COMMENT#{图片id},#{时间}"。

当同时有动态消息和评论消息时，优先显示评论消息的数字。

好友动态在会话列表中按最新的动态的时间自然排序。

16.3 对方加我时的单聊提醒消息

所有消息id以"FOLLOW"开头的单聊消息，都属于加关注的提醒消息。

该消息的id格式为“FOLLOW#{uid1},#{uid2}”，其中uid1是加我的人的id，uid2是我的id。

16.4 谁访问过我的主页的消息

所有消息id以"VISIT"开头的单聊消息，都属于访问我的主页的提醒消息。

该消息的id格式为“VISIT#{uid1},#{uid2}”，其中uid1是访问我的人的id，uid2是我的id。

17 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。

17.1 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 |
| filter1 | 整数 | 可选 | 使用的滤镜编号, 从1开始。0代表未使用滤镜, 缺省0 |
| filter2 | 整数 | 可选 | 使用的图像合成编号, 从1开始。0代表未使用合成, 缺省0 |
| 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 |
| total | 整数 | 可选 | 如果多张图分享, 图片的总数 |
| time | 整数 | 可选 | 多张图上传时客户端时间, unix时间1970到秒 |

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",
  "mid": "ckn509a1ffcbe4b1982a4000002",
  "weibo": false, "room": "1",
  "user_id": "502e61bfbe4b1921da000005",
  "logo": "/uploads/tmp/20121107-1646-42114-4251/111.jpg",
  "logo_thumb2": null, "id": "509a1ffcbe4b1982a4000002" }
```

[2014-01-03] 图片上传成功返回值中增加mid字段，表示该图片发送到聊天室的消息id。

17.2 多图上传时，非第一张图的上传

17.2.1 获取七牛云存储的图片上传token

| | | | |
|---------|-----------------|----|-----------|
| POST | /photos/uptoken | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必须 | 当前登录用户的id |

17.2.2 上传图片

调用七牛的API上传文件，其中包含的参数有：

| POST | 七牛的上传模式2：高级上传（带回调） | | |
|-----------------|--------------------|----|-----------------|
| 参数 | 类型 | 必填 | 说明 |
| uptoken | 字符串 | 必须 | 上一步获得的token |
| file | 文件 | 必须 | 要上传的文件 |
| bucket | 字符串 | 必须 | 必须是”dphoto” |
| key | 字符串 | 必须 | 上传时的主键，格式见下面的说明 |
| callback_params | 字符串 | 必须 | 回调参数 |

key的格式为：“uid/time-seq”，其中hid就是发图者的id，time就是第一张图里带的time参数，seq表示第几张图。

回调参数的格式：“from=&room=&id=&key=\$(etag)&size=\$(fsize)”

其中from是发送者的uid，

room是聊天室编号，

id是上面的主键key，

“key=\$(etag)&size=\$(fsize)”这一段则不做改变。

17.3 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。

如果一次上传了多张图片,那么xmpp消息格式为:

[img:\$id]\$total:\$desc

其中total是上传图片的总数。

17.4 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

如果是多图上传, 其它的图直接通过url拿, 不需要调用本接口。

17.5 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 |

如果是多图上传, 新增两个参数photos和thumb2s, 分别表示多图的原图和缩略图的url, 类型是数组, 不包含第一张图。

18 照片墙

18.1 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。商家和商家的员工（也就是在聊天室加V的用户）可以删除本商家的图片。

18.2 删除聊天室消息

| | | | |
|---------|-----------------|----|----------|
| POST | /shop/del_gchat | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 字符串 | 必须 | 聊天室消息的ID |
| sid | 字符串 | 必须 | 商家ID |
| user_id | 字符串 | 必须 | 当前登录用户id |

只有商家和商家的员工（也就是在聊天室加V的用户）删除聊天室消息时才调用本接口、以及个人删除自己发的消息时调用。其它用户删除聊天室消息只是删除自己本地的消息。

18.3 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 |

18.4 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。

18.5 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 |
| sid | 字符串 | 可选 | 当前用户所在现场的id |

[2013-9-24] 新增sid，表示发表评论的用户所在现场的id。同时，如果评论者的sid和图片的商家id一致，那么会在聊天室中发布一条动态消息，消息id以COMMENT开头。评论消息的格式和普通图片消息一致。

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

18.6 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 |
| sid | 字符串 | 可选 | 当前用户所在现场的id |

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。

[2013-9-24] 新增sid，表示发表评论的用户所在现场的id。

18.7 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。

18.8 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。

18.9 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：7.6。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。

18.10 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。

18.11 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。

18.12 其它用户分享到微信

看到别人发的照片/问答, 分享到自己的微信朋友圈或者微信好友, 在客户端完成分享后, 调用本接口。

| | | | |
|----------|------------------|----|-------------------------|
| GET | /photos/re_share | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必须 | 分享者的用户id |
| photo_id | 字符串 | 必须 | 被分享的图片id |
| weibo | 0/1 | 可选 | 是否同步发送到微博 |
| qq | 0/1 | 可选 | 是否同步发送到QQ空间 |
| wx | 0/1/2/3 | 可选 | 分享到微信: 1个人, 2朋友圈, 3都分享了 |

本接口主要用于统计用户和照片的分享数量。分享到微信时, 采用url类型, url地址为“http://www.dface.cn/web_photo/show?id=图片id&uid=用户id”。

19 个人聊天时发图

19.1 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 |

[2014-01-03] 图片上传成功返回值中增加mid字段，表示该图片发送到会话的消息id。

19.2 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]

19.3 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]

19.4 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。

19.5 收到个人聊天图片后转发给好友

| | | | |
|---------|------------------|----|--------|
| GET | /photo2s/forward | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 字符串 | 必须 | 图片id |
| user_id | 字符串 | 必须 | 转发者的id |
| to_uid | 字符串 | 必须 | 接收者的id |

图片转发成功返回值中包含mid字段，表示该图片转发到会话的消息id。

20 个人聊天时发语音

20.1 获取七牛云存储的文件上传token

| | | | |
|---------|------------------|----|-----------|
| POST | /sound2s/uptoken | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必须 | 当前登录用户的id |

20.2 上传文件

调用七牛的API上传文件，其中包含的参数有：

| | | | |
|-----------------|--------------------|----|-------------|
| POST | 七牛的上传模式2：高级上传（带回调） | | |
| 参数 | 类型 | 必填 | 说明 |
| uptoken | 字符串 | 必须 | 上一步获得的token |
| file | 文件 | 必须 | 要上传的文件 |
| bucket | 字符串 | 必须 | 必须是”sound” |
| key | 字符串 | 必须 | UUID的主键 |
| callback_params | 字符串 | 必须 | 回调参数 |

回调参数的格式：`"sec=&from=&to=&id=&key=$(etag)&size=$(fsize)"`
其中sec是声音的长度（秒），
from是发送者的uid，
to是接受者的uid，
id是客户端生成的UUID主键key，
"key=\$(etag)&size=\$(fsize)"这一段则不做改变。

20.3 上传成功后的回调

文件上传成功后，七牛服务器会回调脸脸服务器的接口，然后把响应转发给客户端。

| | | | |
|------|-------------------|----|----------------------|
| POST | /sound2s/callback | | |
| 参数 | 类型 | 必填 | 说明 |
| sec | 整数 | 可选 | 语音长度：秒 |
| from | 字符串 | 必须 | 发送者uid |
| to | 字符串 | 必须 | 接受者uid |
| id | 字符串 | 必须 | 语音文件的id，也就是客户端生成的key |

注意：本接口不需要客户端调用。

比如上一步中callback_params 的输入：
"sec=3&from=fid&to=tid&id=1376548998&key=\$(etag)&size=\$(fsize)"

本接口的输出如下：
{ "from"=>"fid", "sec"=>"3", "size"=>"691", "to"=>"tid"
"id"=>"1376548710", "key"=>"FgaFYWcGt7w4f60v4NG2DB8zKSNV" }

20.4 发送消息

语音上传完成后，服务器端发送一个xmpp消息，其中的body内容格式为：

[sound:\$id]\$sec

20.5 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 |

21 用户推荐

21.1 将个人推荐给好友

| POST | /follows/recommend | | |
|------|--------------------|----|-------------------|
| 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 |
| mid | 字符串 | 必填 | 客户端生成的UUID格式的消息ID |

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" : mid}

21.2 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]

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。

22 优惠券

22.1 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。

[2013-11-13] 优惠券消息增加一个**seq**属性，代表优惠券的编号。

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。

22.2 获取优惠券列表

当用户收到优惠券以后，先根据优惠券的Xmpp消息显示“优惠券生成中...”的效果，然后起http请求拿到优惠券列表。优惠券Xmpp消息给的是优惠券的基本信息，本接口给的是优惠券的详细信息。

关于优惠券的排序，默认按时间排序，最新的在最前面。相同时间同一批收到的优惠券，按距离排序。当前所在商家的优惠券自动置顶。

| | | | |
|---------|---------------|----|---|
| GET | /coupons/list | | |
| 参数 | 类型 | 必填 | 说明 |
| user_id | 字符串 | 必填 | 用户的id |
| status | 数字 | 必填 | 优惠券的状态，1代表未使用的，2代表已使用的，4代表已过期的，8代表未激活的，0代表未使用和未激活的，默认为0 |
| lat | 浮点数 | 可选 | 经度 |
| lng | 浮点数 | 可选 | 纬度 |
| sid | 整数 | 可选 | 当前现场的商家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10 |

例子：

访问： /coupons/list?

返回值：

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

id是该优惠券的id，name是优惠券的名称、shop是商家的名称，dat是该张优惠券的下发时间，status是状态，hint是优惠券使用提示，seq是优惠券的流水号。

22.3 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。 |
| seq | 整数 | 可选 | 0代码客户端没有seq, 1代表客户端支持显示seq, 默认为0 |

2.4以后的版本, 都要传递seq=1, 这样服务器端才不会重复生成seq.

22.4 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 |
| sid | 整数 | 可选 | 当前现场的商家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。

优惠券使用时, 增加一个可选的参数sid, 代表当前现场的商家id。如果用户没有摇入任何商家, sid不传。

22.5 优惠券删除

| | | | |
|---------|-----------------|----|---------|
| GET | /coupons/delete | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 字符串 | 必填 | 该优惠券的id |
| user_id | 字符串 | 必填 | 用户的id |

22.6 优惠券批量使用

本接口用于数据同步。

| | | | |
|-------|--------------------|----|--------|
| GET | /coupons/batch_use | | |
| 参数 | 类型 | 必填 | 说明 |
| infos | 字符串 | 必填 | 批量使用信息 |

不同的优惠券以,分割，同一张优惠券内发送三个参数id-sid-data。

22.7 优惠券批量删除

| | | | |
|-----|-----------------------|----|------------------|
| GET | /coupons/batch_delete | | |
| 参数 | 类型 | 必填 | 说明 |
| ids | 字符串 | 必填 | 该优惠券的id,以 “,” 分割 |

23 XMPP服务器的Web接口

23.1 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 |

返回值的数据结构: [发送者ID,消息内容,以秒表示的时间,消息ID]

23.2 获得聊天室的消息历史记录2

| GET | http://xmpp_ip:5280/api/gchat2 | | |
|-------|--------------------------------|----|----------|
| 参数 | 类型 | 必填 | 说明 |
| room | 字符串 | 必填 | 聊天室的房间id |
| skip | 数字 | 必填 | 跳过多少条 |
| count | 数字 | 必填 | 取多少条 |

注意: skip=0时, 消息历史用ejabberd的内存中获取; skip大于0时, 从数据库取。

返回值的数据结构: [发送者ID,消息内容,以秒表示的时间,消息ID]

23.3 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 |

返回值的数据结构: [接受者ID,消息内容,以秒表示的时间,消息ID]

23.4 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 |

返回值的数据结构: [发送者ID, 接受者ID,消息内容,以秒表示的时间,消息ID]

23.5 其它内部接口

仅用于服务器端的后台内部调用。

23.5.1 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。

23.5.2 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 |

23.5.3 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 |

23.5.4 获取用户屏蔽列表

| | | | |
|------|-----------------------------------|----|----------------|
| POST | http://xmpp_ip:5280/api/blocklist | | |
| 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 |

23.5.5 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 |

23.5.6 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id |

23.5.7 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id |

23.5.8 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。

24 关于Push消息

24.1 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ios发布设备，3代表android个推的clientid，4代表百度云推送。

对于iOS设备，token是一个64位的字符串。对于android设备，如果token小于64，以0填充尾部到64位。百度云推送的token格式为：channel_id,user_id,补齐0到64位)

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：

“c53b2f16a24c61d91ea6e4f05b48f4a057816956e567c6feacf14ee28cbbbab67993e263c3dfa2c27”

服务器端解析出状态码和token并保存。

24.2 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。如果是个推，退出时要接触和个推的绑定。

24.3 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。

25 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。

26 自定义菜单

脸脸的自定义菜单兼容微信的自定义菜单。如果获得商家信息时有”has_menu”字段，并且值大于0，代表该商家有自定义菜单。”has_menu”值等于1时默认显示聊天工具栏，2默认是菜单栏。

26.1 获取菜单

| | | | |
|------|-----------|----|------|
| POST | /menu/get | | |
| 参数 | 类型 | 必填 | 说明 |
| id | 整数 | 必须 | 商家id |

返回值：

```
{
  "menu": {
    "button": [
      {
        "type": "click",
        "name": "今日歌曲",
        "key": "V1001_TODAY_MUSIC",
        "sub_button": []
      },
      {
        "type": "click",
        "name": "歌手简介",
        "key": "V1001_TODAY_SINGER",
        "sub_button": []
      },
      {
        "name": "菜单",
        "sub_button": [
          {
            "type": "view",
            "name": "搜索",
            "url": "http://www.soso.com/",
            "sub_button": []
          },
          {
            "type": "view",
            "name": "视频",
            "url": "http://v.qq.com/",
            "sub_button": []
          },
          {
            "type": "click",
            "name": "赞一下我们",
            "key": "V1001_GOOD",
            "sub_button": []
          }
        ]
      }
    ]
  }
}
```

26.2 点击菜单

自定义菜单有两种类型：view和click。

当点击view类型时，判断url的类型进行处理。如果url是http链接，那么打开一个新的webview显示这个url。如果url是dface链接，那么按照脸脸应用内跳转规范处理。

当点击click类型时，访问下面的接口并显示提示“消息收取中...”。

| | | | |
|---------|-------------|----|----------|
| GET | /menu/click | | |
| 参数 | 类型 | 必填 | 说明 |
| sid | 整数 | 必须 | 商家id |
| user_id | 字符串 | 必须 | 用户id |
| key | 字符串 | 必须 | 点击菜单的KEY |

执行成功后，脸脸会发送回一条消息。

27 脸脸WEB应用嵌入接口

本章所有的功能只对域名以dface.cn结尾的网址有效。

27.1 加载和退出

启动时：

- 第一步：WEB应用中嵌入链接“dface://init?mode=”，
- 第二步：脸脸客户端监测到链接“dface://init”，回调WEB应用中的js方法：dface_init(sid,uid)，其中sid是摇入的现场的id，uid是当前用户的id，都是字符串。mode定义浏览器的展现，0全屏，1标准，2只有头部，3只有底部。默认为0。

关闭时:

第一步: WEB应用退出时, 触发链接 “dface://close”,

第二步: 脸脸客户端监测到链接 “dface://close”, 关闭WebView浏览器。

27.2 脸脸客户端Js API

27.2.1 隐藏脸脸右上角的选择项按钮

```
DfaceJSBridge.call('hideOptionsMenu');
```

27.2.2 设置脸脸的标题

```
setTitle
```

27.2.3 获得当前用户名

```
getUsername
```

27.2.4 获得当前用户名的session id

```
getSessionID
```

27.2.5 获得当前用户id和商家id

```
getUidSid
```

27.2.6 获得当前经纬度

```
getLoc
```

27.2.7 分享到微信好友

```
wxShare
```

27.2.8 分享到微信朋友圈

wxmShare

28 脸脸应用内跳转规范

现场

dface://scheme/scene

查找现场

dface://scheme/seek

dface://scheme/seek/relocate

热点

dface://scheme/near/user

dface://scheme/near/shop

对话

dface://scheme/record

dface://scheme/record/coupon

dface://scheme/record/user/XXidXXX?name=张三

好友

dface://scheme/friend/good

dface://scheme/friend/follow

dface://scheme/friend/fan

设置

dface://scheme/setting

dface://scheme/setting/help

dface://scheme/setting/blacklist

我的

dface://scheme/atme

dface://scheme/atme/gallery

dface://scheme/atme/footmark

获取图片

dface://scheme/getphoto 用户自选

dface://scheme/getphoto/camera 打开相机

dface://scheme/getphoto/gallery 打开本地相册

图片

dface://scheme/photo?id=XXXXXX 查看图片详情页面

用户信息

dface://scheme/user/info?id=XXXXX 查看用户信息

29 脸脸聊天图片上传时的压缩规则

1. 图片压缩时的质量控制：图片压缩时的质量控制在0.95-0.8之间。当图片宽乘以高小于等于40万时，质量为0.95；当图片大于400万时，质量为0.8；当图片的大小（设为x）在40-400万之间时，质量为 $0.95 - (x - 40) * 0.15 / 360$ 。

2. 图片尺寸控制：当图片的最短边小于640时，不处理；否则，设短边为x，长边为y，压缩前尺寸(x, y)，压缩后的尺寸为(640, $y * 640 / x$)

3. 上述规则只使用于聊天图片，个人头像还是采用原来的压缩到640*640的规则。

30 坐标转换接口

30.1 实际经纬度坐标转GCJ坐标

| | | | |
|-----|---------------|----|----|
| GET | /loc/real2gcj | | |
| 参数 | 类型 | 必填 | 说明 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |

返回转换后的坐标 {lat:, lng:}

30.2 实际经纬度坐标转Baidu坐标

| | | | |
|-----|----------------|----|----|
| GET | /loc/real2bd09 | | |
| 参数 | 类型 | 必填 | 说明 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |

30.3 GCJ坐标转实际坐标

| | | | |
|-----|---------------|----|----|
| GET | /loc/gcj2real | | |
| 参数 | 类型 | 必填 | 说明 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |

30.4 百度坐标转实际经纬度坐标

| | | | |
|-----|----------------|----|----|
| GET | /loc/bd092real | | |
| 参数 | 类型 | 必填 | 说明 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |

30.5 百度坐标转GCJ坐标

| | | | |
|-----|---------------|----|----|
| GET | /loc/bd092gcj | | |
| 参数 | 类型 | 必填 | 说明 |
| lat | 浮点数 | 必须 | 经度 |
| lng | 浮点数 | 必须 | 纬度 |

31 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。