

# 脸脸网服务器与手机端的开发接口API

YXY

2012 年 8 月 9 日

## 目录

|                       |    |
|-----------------------|----|
| 1 根据IP地址获得可能的现场商家     | 4  |
| 2 根据经纬度获得可能的现场商家      | 4  |
| 3 根据经纬度获得附近的商家        | 6  |
| 4 根据经纬度获得附近的用户        | 7  |
| 5 根据商家ID获得当前在该商家的所有用户 | 8  |
| 6 根据商家ID获得商家的公告       | 9  |
| 7 新浪微博登录后的回调页         | 9  |
| 8 进入现场签到              | 12 |
| 9 获得用户基本信息            | 12 |
| 10 获得用户的头像            | 14 |
| 11 获得用户的图片列表          | 14 |

|                                 |    |
|---------------------------------|----|
| 目录                              | 2  |
| 12 获得当前登录用户基本信息                 | 15 |
| 13 设置当前登录用户基本信息                 | 17 |
| 14 当前登录用户上传图片                   | 18 |
| 15 当前登录用户调整图片位置                 | 19 |
| 16 当前登录用户批量调整所有图片的位置            | 20 |
| 17 当前登录用户删除图片                   | 21 |
| 18 添加我关注的人                      | 21 |
| 19 删除我的关注的人                     | 22 |
| 20 粉丝列表                         | 23 |
| 21 关注(好友)列表                     | 25 |
| 22 黑名单列表                        | 25 |
| 23 添加黑名单                        | 26 |
| 24 删除黑名单                        | 26 |
| 25 XMPP协议接口                     | 27 |
| 25.1 与Openfire聊天服务器接口 . . . . . | 27 |
| 25.2 脸脸中的聊天用户的三种类型 . . . . .    | 28 |
| 25.3 摇一摇及其消息格式 . . . . .        | 28 |
| 25.4 个人之间聊天的消息发送状态确认 . . . . .  | 29 |
| 26 关于程序各页面之间的切换规则               | 29 |

**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. 获得附近的商家分解为三个API，对应三种情况。
3. 新浪微博登录回调以后的API调用时都需要登录。目前登录信息通过cookie中的“\_session\_id”获得。
4. 当调用出错时，会在返回的json中包含“error”信息。

## 1 根据IP地址获得可能的现场商家

| http://www.dface.cn/aroundme/shops_by_ip |     |    |             |
|--|-----|----|-------------|
| 参数                                       | 类型  | 必填 | 说明          |
| ip                                       | 字符串 | 可选 | 默认为访问者的IP地址 |

最多返回10条数据，同一个IP地址商家应该不多。

定位时不再需要本接口，直接调用根据经纬度获得可能的现场商家的接口。

## 2 根据经纬度获得可能的现场商家

| http://www.dface.cn/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 |

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。

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。

3 根据经纬度获得附近的商家

| http://www.dface.cn/mshop/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        |

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。

4 根据经纬度获得附近的用户

| http://www.dface.cn/aroundme/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          |
| 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 |

例子：

访问: /aroundme/users.json?lat=30.2829754&lng=120.1337336&accuracy=100  
返回值:

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

如果用户登录后访问该API, 那么返回的属性中包括"friend"和"follower"。

"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。

5 根据商家ID获得当前在该商家的所有用户

|                                 |    |    |       |
|---------------------------------|----|----|-------|
| http://www.dface.cn/mshop/users |    |    |       |
| 参数                              | 类型 | 必填 | 说明    |
| id                              | 整数 | 必须 | 商家的ID |



返回值格式和“根据经纬度获得附近的用户”基本一致，加上time字段。

[2012-8-8] 新增time字段，表示用户在该商家的最后活跃时间。

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。当用户进入下一个商家时，自动判定为离开上一个商家。如果用户没有退出，那么默认12小时后用户离开商家。

在商家的用户的集合要保证大于在该商家聊天室的用户的集合。

当前的商家用户数据则完全是基于方便测试的目的构造的。

## 6 根据商家ID获得商家的公告

| http://www.dface.cn/shop_notices |    |    |       |
|----------------------------------|----|----|-------|
| 参数                               | 类型 | 必填 | 说明    |
| id                               | 整数 | 必须 | 商家的ID |

返回值:

```
[  
{"id":1,"title":"asdf"},  
{"id":2,"title":"a title"}  
]
```

## 7 新浪微博登录后的回调页

| http://www.dface.cn/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: [http://www.dface.cn/oauth2/sina\\_callback?code=...](http://www.dface.cn/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。

8 进入现场签到

| http://www.dface.cn/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 |
| mshop_id                     | 整数  | 必须 | 现场商家id  |
| user_id                      | 整数  | 必须 | 签到用户id  |

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。

9 获得用户基本信息

| http://www.dface.cn/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/avatars/1/original/clojure.png?1339398140",
  "logo_thumb":"/system/avatars/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
}
```

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**。

[2012-8-8] 新增签名、职业、爱好等字段。

[2012-8-8] 新增**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 获得用户的头像

| http://www.dface.cn/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75*75；2代表thumb2缩略图，大小为150*150。默认为1。 |

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/avatars/1/thumb2/1.png?1339649979”

Response Body中包含头像的二进制信息。

# 11 获得用户的图片列表

| http://www.dface.cn/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/avatars/2/thumb2/2.png.png?1340616951","updated_at":"2012-01-01 12:00:00",
"id":2,"logo":"/system/avatars/2/original/2.png.png?1340616951","avatar_file_size":10240},
{"logo_thumb2":"/system/avatars/4/thumb2/2.png.png?1340617039","updated_at":"2012-01-01 12:00:00",
"id":4,"logo":"/system/avatars/4/original/2.png.png?1340617039","avatar_file_size":10240}
]
```

图片列表中的第一张图片就是用户的头像。

## 12 获得当前登录用户基本信息

|  |    |    |    |
|--|----|----|----|
| http://www.dface.cn/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/avatars/1/original/clojure.png?1339398140",
  "logo_thumb":"/system/avatars/1/thumb/clojure.png?1339398140"
  "password":"c84dad462d5b7282",
  "id":1
}
```



### 13 设置当前登录用户基本信息

| http://www.dface.cn/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                       |
| job type                          | 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: 对陌生人隐身,2: 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。

例子：

访问: curl -b "\_session\_id=ead9ac4f6291c55bb467ad4138eca2ed"  
-F "name=newname" http://www.dface.cn/user\_info/set

返回值:

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

## 14 当前登录用户上传图片

| http://www.dface.cn/user_logos |      |    |      |
|--------------------------------|------|----|------|
| 参数                             | 类型   | 必填 | 说明   |
| user_logo[avatar]              | file | 必须 | 头像文件 |

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. 图片文件必须小于5M。

- 4. 图片名为logo；缩略图分为两种大小，logo\_thumb为75\*75的png，logo\_thumb2为150\*150的png。

例子：

访问： curl -F "user\_logo[avatar]=@firefox.png;type=image/png"  
http://www.dface.cn/user\_logos

```
{"logo_thumb2":"/system/avatars/6/thumb2/csky2.png?1340779488",  
  "updated_at":"2012-06-27T06:44:48Z"  
  , "id":6,  
  "logo":"/system/avatars/6/original/csky2.png?1340779488",  
  "avatar_file_size":156604,  
  "user_id":3,  
  "logo_thumb":"/system/avatars/6/thumb/csky2.png?1340779488"}
```

## 15 当前登录用户调整图片位置

| http://www.dface.cn/user_logos/position |    |    |           |
|---|----|----|-----------|
| 参数                                      | 类型 | 必填 | 说明        |
| id                                      | 整数 | 必须 | 图片的id     |
| order                                   | 整数 | 必须 | 新的位置，从0开始 |

例子：

访问: curl -F "id=2;order=0"  
http://www.dface.cn/user\_logos/position

16 当前登录用户批量调整所有图片的位置

| http://www.dface.cn/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：当只有一到两张图片更改位置时，不能调用本接口，需要调用单独修改图片位置的接口。

比如原有图片的循序是2468，新的循序是2846，此时需要调用前一个接口，传递id=8&order=1。此时服务器只需要更新一条记录。如果调用本接口，会导致服务器端所有图片都被更新。

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"  
http://www.dface.cn/user\_logos/position

## 17 当前登录用户删除图片

| http://www.dface.cn/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"`

`http://www.dface.cn/user_logos/delete`

输出:

`{"deleted": "2"}`

## 18 添加我关注的人

| http://www.dface.cn/follows |    |    |           |
|-----------------------------|----|----|-----------|
| 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" http://www.dface.cn/follows`

返回值:

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

## 19 删除我的关注的人

| http://www.dface.cn/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" http://www.dface.cn/follows/delete
```

返回值:

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

## 20 粉丝列表

| http://www.dface.cn/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}
}
]
```

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**。



## 21 关注(好友)列表

| http://www.dface.cn/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

这里返回的所有用户, 其”friend”值一定为true。

## 22 黑名单列表

| http://www.dface.cn/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。

## 23 添加黑名单

| http://www.dface.cn/blacklists |    |    |                      |
|--------------------------------|----|----|----------------------|
| 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 |

注意：该请求必须是POST请求。

返回值：

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

## 24 删除黑名单

| http://www.dface.cn/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   |

注意：该请求必须是POST请求。

例子：

```
访问： curl -b "_session_id=f03bef9371c119b6fcecbeefdaaac1b2"  
        -d "user_id=2&block_id=3" http://www.dface.cn/blacklists/delete
```

返回值：

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

## 25 XMPP协议接口

### 25.1 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。
2. 商家的id加上"@conference.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。
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. 脸脸中的隐身也和xmpp中的隐身状态无关。
7. 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。

## 25.2 脸脸中的聊天用户的三种类型

1. 管理员，其jid固定为"1@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。

## 25.3 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：

```
<handshake name="用户名" xmlns="http://dface.cn" />
```

如果是用户第一次进店的时候摇一摇，那么增加first属性，如：

```
<handshake name="用户名" first="true" xmlns="http://dface.cn" />
```

用户第一次进店摇一摇的时候可以参与抽奖，其后的摇一摇不行。用户在12小时后只能在一个商家参与抽奖一次，也就是在首页的进店摇一摇只有12小时内的第一次发送first属性。

其它客户端收到摇一摇的XML格式消息后，再将其转换为文字描述。

## 25.4 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/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。

## 26 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，进入选择现场页面，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，进入选择现场页面，并从服务器端获取新的商家列表。

3、从现场返回到选择现场时，判断当前的经纬度和上次进入现场时的经纬度。

3.1 如果超过500米，提示用户是否刷新现场位置。

3.2 如果不足500米，不处理。用户要选择其它现场的话，从滚轮中选择。

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。进入时要特殊处理下部的tab，提供动画。只有这种情况需要这么特殊处理。

4.2 如果没有摇，那么就是选择现场页面。

5、从附近里点击商家的时候。不进入现场。显示商家的聊天室。但是该聊天室不是现场，没有聊天工具栏。在附近里进入商家聊天室时不能聊天。