# 概述

## 编写目的

本文给出了Samchat手机终端APP与服务器交互接口中所有相关数据结构的定义和相关数据处理流程定义,从而为接口软件的开发提供设计依据.

本文的读者主要包括:

* 开发工程师:能够依据相关设计进行软件编码
* 测试工程师:了解软件流程以便选择或设计相关的测试计划与测试用例
* 其它相关人员:了解接口的实际流程，便于同有关人员进行沟通

## 背景

根据市场人员反馈，市场需要有一种更加直接和有效的方式来将普通用户和商家连接来并为双方提供简单易用的CRM。Samchat系统的出现就是利用多媒体通信的手段，通过问与答的方式，让商家寻找到可能存在需求的用户并对用户直接进行线上营销。本文涉及到的是Samchat系统中的终端APP和后台服务端的接口(API)总体设计.

## 范围

本文是Samchat系统中有关接口软件的概要设计和详细设计的文档。其内容主要包括通信协议、相关数据结构定义和交互流程定义。

## 术语与缩略语

以下给出一些系统中的术语与缩略语定义:

### Sam-pros

Samchat系统中的商家，解答问题并提供相关服务的人。

### Public

公众号，商家用于发布广告的账号，在Samchat系统中，Public Account就是Sam-pros Account。

### Auto Sign In

自动登录，当用户曾经在某个手机登录过后，即使结束Samchat APP，当用户再次打开Samchat APP时，App将在后台自动登录而不需要用户再次输入登录名和密码。

### Favourite

置顶，将某个Public Account置顶，在显示列表中将处于列表的顶端。当用户换一台手机登录时，这个被关注的Public Account也仍然处于显示列表的顶端。

### Block

当某个Public Account被block后，这个商家发布的广告将不会被这个用户接收到直到用户取消对这个Public Account的block(unblock)。

### Query nearby public

根据用户当前所在的位置信息，搜索处于用户附近的Public Account。

### Sync advertisement

当商家换了手机后，商家仍然有编辑已经发布的广告需求。所以当商家进入自己发布广告列表页面时，APP将从服务器Sync最近的10条广告，当用户往下划到底后会再Sync更旧的10条广告，直到广告被全部Sync到新的手机上。在Sync过程中，如果发现存在于本地数据的广告已经被用户在别的手机上通过删除操作删除了，那么APP将会从本地数据库清除此条广告。

## 参考资料

## 1.6 跟踪列表

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改人 | 修改内容 | 时间 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 总体设计

## 总体架构



* IM 接口: 使用第三方IM服务商云信接口实现IM功能
* Push接口: 使用第三方Push服务商个推接口实现Push功能
* 图片上传接口: 与Documentation Server通过Web接口实现，Document Server将采用AWS中的 Amazon S3
* 业务接口：App Server 的接口

本文定义的接口就是定义手机与App Server 之间以及手机与Documentation Server之间的接口协议

## User Case

通过对需求分析，与接口设计相关的user case包括以下这些case:

* Seq-Signup
* Seq-SignIn
* Seq-AutoSignIn
* Seq-SignOut
* Seq-Create Sam-pros Account
* Seq-Find Password
* Seq-Update Password
* Seq-Send Request
* Seq-GetPlacesInfo
* Seq-Browse response-follow/unfollow
* Seq-Query User
* Seq-Browse public-block/unblock
* Seq-Browse public-favourite/unfavourite
* Seq-Query public
* Seq-Sync follow list
* Seq-Edit profile
* Seq-Update avatar
* Seq-Write advertisement
* Seq-Delete advertisement

# 详细设计

## Seq-Signup

Seq-Signup 定义用户注册Samchat Account时APP与Server之间的交互流程已及相关Json 数据结构。注册只支持手机号码注册。



### Register code request/response json

{

"header":

{

"action" : "register-code-request",

},

"body":

{

"countrycode" : “”

"cellphone" : “”

"deviceid" : “”

}

}

成功返回：

{

"ret":0

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

此电话号码已经注册过 :{ret : -201}

非法电话号码: {ret:-202}

申请验证码过于频繁:{ret:-206}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_registerCodeRequest.do

### Register code verify json

{

"header":

{

"action" : " signup-code-verify",

},

"body":

{

"countrycode" :

"cellphone" : “”

"verifycode" : “”

"deviceid" :””

}

}

成功返回：

{

"ret":0,

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

非法email/电话号码: {ret:-202}

验证码错误:{ret:-205}

申请验证码过于频繁:{ret:-206}

验证码过期:{ret:-211}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_signupCodeVerify.do

### Sign up request/response json

{

"header":

{

"action" : "register"

},

"body" :

{

“countrycode” :86

“cellphone” :“1381196123”

“verifycode” : 332682

“username” :”Kevin Dong”

“pwd” :”123456”

“deviceid” :”14EF65” //(IMEI/MEID last 6 byte)

}

}

成功返回：

{

“ret”:0,

“token”: "95189457043652618"

“user”:{

“id”: unique\_id\_in\_samchat

“lastupdate”:

},

“sys\_params”:[{

“param\_code”:””,

“param\_value”:””

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

用户名或者手机号已经注册过： {ret: -201}

非法电话号码: {ret: -202}

验证码错误:{ret:-205}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_register.do

## Seq-SignIn

Seq-Signin 定义已经注册的用户登录Samchat时APP与Server之间的交互流程已及相关Json 数据结构。当用户连续输入密码错误达到一定次数后，服务器端将按照设备号与登录用户名为记录，限制登录，过了限制时间后，方可再次尝试登录。

### Sign in request/response json

 {

"header":

{

“action”: “login”

},

"body" :

{

“countrycode” :86

“account” :“1381196123”

“pwd” :”123456”

“deviceid” :”14EF65” //(IMEI/MEID last 6 byte)

}

}

成功返回：

{

“ret”:0,

“token”: "95189457043652618"

“user”: {

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

“sam\_pros\_info”:{

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“address”: // option

}

},

“sys\_params”:[{

“param\_code”:””,

“param\_value”:””

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

密码错误返回： {ret: -203}

此用户不存在:{ret:-207}

密码错误尝试过于频繁:{ret:-208}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_login.do

## Seq-AutoSignIn

Seq-AutoSignIn : 当用户曾经在当前手机登录过了，用户再次启动Samchat APP时用户不再需要输入用户名和密码，APP利用之前登录后保留在手机上的token(deviceid+store token才是真正的token)在后台完成自动登录。Samchat系统利用IM server实现自动登录，当自动登录到IM Server则认为自动登录成功，因为IM server 和 APP server使用同样的token，token在用户输入密码登录后会被更新。所以无需为APP和APP server 之间定义自动登录的json结构。

自动登录的逻辑如下:



## Seq-GetAppKey

Seq-GetAppKey: 当用户登录成功后，去服务器取得登录第三方Push服务的Key。因为把Push Key放在手机上是不安全的，而且当极端情况下Key泄漏后，服务器可以动态更改Key，客户端可以取得新的Key。



### Get app key request/response json

{

"header":

{

"action" : "appkey-get",

"token": ""

},

"body" :

{

}

}

返回结果：

成功返回：

{ "ret ": 0,

"appi" : "",

"appk":"",

"apps":""

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

token 格式不正确： {ret: -4}

token 不合法： {ret: -401}

测试地址:

http://service-test.samchat.com/api\_1.0\_profile\_appkeyGet.do

## Seq-SignOut

Seq-AutoSignOut 定义用户安全退出Samchat系统时APP与Server之间的交互流程已及相关Json 数据结构。



### Sign out request/response json

{

"header":

{

"action" : "logout",

"token": ""

},

"body" :

{

}

}

成功返回： {ret: 0}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

token 格式不正确： {ret: -4}

token 不合法： {ret: -401}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_logout.do

## Seq-Create Sam-pros Account

Seq-Create Sam-pros account 定义用户申请商家账号时APP与App Server 交互过程以及相关Json数据结构。



### Create Sam-pros account request/response json

{

"header":

{

"action" : "create-sam-pros",

"token": "token"

},

"body":

{

"company\_name" : “”

"service\_category" : ””

"service\_description" : “”

“countrycode”: // option

“phone”: // option

“email”: // option

"location" :{

"location\_info":{ //option

“longitude”:

“latitude”

}

"place\_id": " "//option

"address": " "//option

}

}

成功返回：

{

"ret":0,

“user”:{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

}

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确： {ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

用户已经有Sam-pros Account :{ret : -501}

商家等待后台审核 :{ret: -505}

测试地址:

## Seq-Find Password

Seq-Find Password 定义注册用户找回密码过程中APP与Server之间的交互流程已及相关Json 数据结构。Samchat系统提供两种找回密码的方式，一种是通过Phone number，另一种通过Email。

在整个找回密码过程中会做2次对校验码的验证，在验证校验码和修改密码时服务器都要验证此时的校验码合法性。

### Find password by phone request/response json



#### findpwd-code-request

{

"header":

{

"action" : "findpwd-code-request",

},

"body":

{

"countrycode" : “”

"cellphone" : “”

"deviceid" :””

}

}

成功返回：

{

"ret":0

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

非法电话号码: {ret:-202}

电话号码未注册过 :{ret : -204}

申请验证码过于频繁:{ret:-206}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_findpwdCodeRequest.do

#### findpwd-code-verify

{

"header":

{

"action" : " findpwd-code-verify",

},

"body":

{

"countrycode" :

"cellphone" : “”

"verifycode" : “”

"deviceid" :””

}

}

成功返回：

{

"ret":0,

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

非法email/电话号码: {ret:-202}

此email/电话号码未注册过 :{ret : -204}

验证码错误:{ret:-205}

申请验证码过于频繁:{ret:-206}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_findpwdCodeVerify.do

#### findpwd-update

{

"header":

{

"action" : " findpwd-update",

},

"body":

{

"countrycode" :

"cellphone" : “”

" verifycode " :

"pwd" : “”

"deviceid" :””

}

}

成功返回：

{

"ret":0,

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

非法email/电话号码: {ret:-202}

此号码未注册过 :{ret : -204}

验证码错误:{ret:-205}

申请验证码过于频繁:{ret:-206}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_findpwdUpdate.do

## Seq-Update Password

Seq-Update Password 定义已经登录的用户修改密码过程中APP与Server之间的交互流程已及相关Json 数据结构。请注意这里的修改密码与找回密码过程中涉及的修改密码不同，这里只有已经登录成功后的用户才可以调用这个接口。



### Update password request/response json

{

"header":

{

"action" : " pwd-update",

"token" : “”

},

"body":

{

"old\_pwd" : “”

"new\_pwd" : “”

}

}

成功返回：

{

"ret":0,

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确： {ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

原始密码错误 :{ret : -502}

测试地址:

http://service-test.samchat.com/api\_1.0\_user\_pwdUpdate.do

## Seq-Send Request

Seq-Send Request 是Samchat的一个主要业务，它包括以下几个过程:

* A. 将用户提出的问题发送到服务器后台
* B. 服务器分析用户的问题
* C. 通过匹配算法寻找到对应商家
* D. 把问题转发给这些商家

需要设计交互数据结构的有 A,D 两个过程。



### Send question request/response json

{

"header":

{

"action": "question",

"token":"95189486473904140"

},

"body":

{

"opt":[0/1] 0:发送问题 1:TBD,

"question" :"aaa"

"location" :{

"location\_info":{ //option

“longitude”:

“latitude”

}

“scell”:{ // //option

“mcc”:”1”,

“mnc”:”2”,

“lac”:”3”,

“cellid”:”4”

},

"place\_id": " "//option

"address": " "//option

}

}

}

成功返回：

{

"ret":0

"question\_id":1

"datetime": 1454076248624

}

解析失败返回： {ret: -1}

action参数不支持返回： {ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确： {ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

问题发送太频繁: {-501}

测试地址:

http://service-test.samchat.com/api\_1.0\_question\_question.do

### Dispatch question json

{

"header" :

{

"category":"1"

},

"body" :

{

"datetime" : 1452221529441,

" question\_id" : 1,

" question ":"Who am I?",

"opt": [0/1] 0: new Question 1: TBD

“dest\_id”: destination user id

“address”: address //if no address, please assign “”

“user”: {

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“lastupdate”:1454076248624

}

}

}

## Seq-GetPlacesInfo

Seq-GetPlacesInfo 定义用户在输入部分地址信息时，由Server 返回依据输入信息返回相关的地址信息. 默认使用 Google Places API 实现.



### GetPlacesInfo request/response json

{

"header":

{

"action" : "get-places-info-request",

"token" : "token"

},

"body":

{

"key" :

}

}

成功返回:

{

"ret" : 0

"count" :

"places\_info":[

{

description:

place\_id:

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_profile\_getPlacesInfoRequest.do

## Seq- Follow/unFollow

Seq-Follow/unFollow 定义用户关注某个公众号过程中APP 与 Server之间交互流程以及相关json数据结构。



### Follow/unFollow request/response json

{

"header":

{

"action" : " follow",

"token" : "token"

},

"body":

{

"opt" : [0/1]

0 : unfollow

1 : follow

"id" : unique\_id\_in\_samchat of Sam-pros

}

}

成功返回:

{

"ret" : 0,

"user" :

{

“lastupdate”:

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

非商家用户:{ret : -503}

此用户不存在:{ret:-207}

还未关注此商家:{ret:-507}

测试地址：

http://service-test.samchat.com/api\_1.0\_officialAccount\_follow.do

## Seq-Block/unBlock

Seq-Block/unBlock意思是指用户暂时不想接收到某个被关注商家的广告信息，只有通过unBlock解开阻止后才能继续收到商家新的广告。



### Block/unBlock request/response json

{

"header":

{

"action" : " block",

"token" : "token"

},

"body":

{

"opt" : [0/1]

0 : unblock

1 : block

" id " : unique\_id\_in\_samchat of Sam-pros

}

}

成功返回:

{

"ret" : 0,

"user" :

{

“lastupdate”:

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

非商家用户:{ret : -503}

此用户不存在:{ret:-207}

还未关注此商家:{ret:-507}

测试地址：

http://service-test.samchat.com/api\_1.0\_ officialAccount\_block.do

## Seq-Favourite/unFavourite

Seq-Favourite/unFavourite 表示用户将某个公众号置顶，此公众号将一直处于显示列表的顶端。而且如果用户在另外的手机上登录，此公众号也仍然处于置顶状态。所以当用户选择置顶/取消置顶后，APP 需要通知Server 维护这个状态。



### Favourite/unFavourite request/response json

{

"header":

{

"action" : " favourite",

"token" : "token"

},

"body":

{

"opt" : [0/1]

0 : unfavourite

1 : favourite

" id " : unique\_id\_in\_samchat of Sam-pros

}

}

成功返回:

{

"ret" : 0,

"user" :

{

“lastupdate”:

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

非商家用户: {ret : -503}

此用户不存在:{ret:-207}

还未关注此商家:{ret:-507}

测试地址：

http://service-test.samchat.com/api\_1.0\_ officialAccount\_favourite.do

## Seq-Query User

Seq-Query User 定义了用户查询并取得某用户信息过程中APP与Server之间的交互过程以及相关json数据结构。

查询用户信息API有4种查询方式:

* 用户登录后模糊查询用户的信息
* 用户登录后根据精确信息查询用户信息
* 用户登录后查询一组用户的信息
* 用户在未登录情况下查询某个用户信息，这种查询只能在某些特殊场合才可以使用,比如在用户注册时APP用于查询用户输入的电话或用户名是否已经注册过了



### 登录后模糊查询用户的信息

{

"header":

{

"action" : "query-fuzzy",

"token": "token"

},

"body":

{

“count”:500,

"opt":1, 1: Fuzzy User Query

"param":

{

"search\_type":"1", // 0. User 1. Sam-pros 2. all

"search\_key":""

}

}

}

成功返回:

{

"ret" : 0,

"count" : users count

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

“sam\_pros\_info”:{

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“address”: // option

}

}]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_queryFuzzy.do

### 登录后根据精确信息查询用户信息

{

"header":

{

"action" : "query-accurate"

"token": "token",

},

"body":

{

"opt":2, 2: Query by cellphone or username

"param":{

"type":[0/1/2] 0: cellphone 1:unqiue\_id 2: username

"cellphone": //option

"unique\_id ": //option

" username": //option

}

}

}

成功返回:

{

"ret" : 0,

"count" : users count,

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

“sam\_pros\_info”:{

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“address”: // option

}

}]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_queryAccurate.do

### 登录后查询一组用户的信息

{

"header":

{

"action" : "query-group"

"token": "token",

},

"body":

{

"opt":3, 3: Query group users by unique\_id

"param":{

"unique\_id":[1,2,3,4,5,6]

}

}

}

成功返回:

{

"ret" : 0,

"count" : users count,

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

“sam\_pros\_info”:{

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“address”: // option

}

}]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_queryGroup.do

### 未登录查询

{

"header":

{

"action" : "query-without-token"

},

"body":

{

"opt":4, 4: User Query Without Token

"param":

{

"type":[0/1/2] 0: cellphone 2:username

"username":"" //optional

"countrycode":"" //optional

"cellphone ":"" //optional

}

}

}

成功返回:

{

"ret" : 0

"count" : users count

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

查询过于频繁：{ret: -209}

测试地址：

http://service-test.samchat.com/api\_1.0\_user\_queryWithoutToken.do

## Seq-Send Invite Msg

Seq-Send Invite Msg用户可以向自己电话联系人发送加入Samchat的邀请消息.



### Send Invite Msg request/response json

{

"header":

{

"action" : "send-invite-msg",

"token": "token"

},

"body":

{

"phones":[{

“countrycode”: “”//option

“cellphone”:””

}]

"msg":

}

}

成功返回:

{

"ret" : 0

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

内部错误： {ret: -103}

发送邀请过于频繁：{ret: -210}

测试地址：

http://service-test.samchat.com/api\_1.0\_common\_sendInviteMsg.do

## Seq-Edit Profile

Seq-Edit profile定义用户更改用户信息时的交互流程以及相关json数据结构。

考虑把profile 具体信息根据UI 分组出来，然后用type 区分每次更新什么信息。



### Edit profile request/response json

{

"header":

{

"action" : " profile-update",

"token" : "token"

},

"body":

{

"user":{

“countrycode”://option

“cellphone”: //option

“email”: //option

“address”:// option

“location":{ // option

"location\_info":{ // option

"longitude":"", // option

"latitude":"" // option

},

"place\_id":"", // option

"address":"" // option

}

“sam\_pros\_info”:{ // option

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“location":{ // option

"location\_info":{ // option

"longitude":"", // option

"latitude":"" // option

}, // option

"place\_id":"", // option

"address":"" // option

}

}

}

}

}

成功返回:

{

"ret" : 0,

“user”:{

“lastupdate”:

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

此电话号码已经注册过 :{ret : -201}

测试地址：

http://service-test.samchat.com*/api\_1.0\_profile\_profileUpdate.do*

## Seq-Update Avatar

Seq-Update Avatar 表示用户更新头像。这里设计将头像的更新API单独出来。因为考虑到后台的架构设计，头像需要先将图片存放到后台server的文件服务器上，然后通知App Server更新数据库信息。所以头像的更新分为下面2个步骤:

* A.上传头像图片给文件服务器并取得头像存放地址的URL
* B.通知APP server 更新数据库信息

A由Amazon S3提供的API完成，所以只设计B需要的API。



### Update avatar request/response json

{

"header":

{

"action" : " avatar-update",

"token" : "token"

},

"body":

{

"avatar":

{

"origin": <http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

},

}

}

成功返回:

{

"ret" : 0,

“user”:{

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

“lastupdate”:

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com*/api\_1.0\_profile\_avatarUpdate.do*

## Seq-Query Public

Seq-Query public 表示用户根据一些特殊条件搜索相关公众号。Samchat根据以下几种条件搜索:

* 根据公众号名字
* 公众号关键字
* 位置信息



### Query public request/response json

{

"header":

{

"action" : "public-query"

"token": "token",

},

"body":

{

“count”:100, //

key: //option

location:{

"location\_info":{

“longitude”:

“latitude

}//option

"place\_id": " "//option

"address": " "//option

}

}

}

成功返回:

{

"ret" : 0,

"count" : users count,

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“countrycode”:

“cellphone”:

“email”: //option

“address”:// option

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

”origin”:<http://121.42.207.185/avatar/2016/1/18/origin_1453123489091.png>

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_1453123489091.png>

}

“lastupdate”:1454076248624

“sam\_pros\_info”:{

“company\_name”:”KFC”

“service\_category”:“fast food”

“service\_description”:”deliver all kinds of fast food”

“countrycode”: // option

“phone”: // option

“email”: // option

“address”: // option

}

}]

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_officialAccount\_publicQuery.do

## Seq-Add/Remove contact

Seq-Sync add/remove contact

添加: 表示将某个用户添加入自己的通讯录中，Samchat系统中，通讯录是单向的，就像电话中的联系人一样，只是自己单方面添加，而对方并不知道，和社交软件的好友联系人不同。

在Samchat中有一种特殊的联系人，就是商家的客户，对于商家来说，这是整个通讯录的一个子集，表示这些用户为商家的常客户

删除: 将某个用户从自己的通讯录中删除，与添加一样，是单方面删除.

而对于客户列表中的联系人，只是移出和删除两种模式

### Add/Remove contact request/response json

{

"header":

{

"action" : "contact "

"token": "token",

},

"body" :

{

"opt": 0: add 1:remove

"type" [0/1]: 0: add into to contact /remove contact

1:add into customer contact/ move from customer to contact

"id": unique id in samchat

}

}

成功返回:

{

"ret" : 0

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com*/api\_1.0\_contact\_contact.do*

## Seq-Sync contact list

Seq-Sync contact list表示sync通讯录列表。当用户登录后，需要每次检查是否要更新本地数据库的通讯录列表。因为可能用户换手机登录，需要同步最新的通讯录列表信息到本地数据库。



### Sync contact list request/response json

{

"header":

{

"action" : "contact-list-query"

"token": "token",

},

"body" :

{

“type[0,1]”: 0 servicer 1 customer

}

}

成功返回:

{

"ret" : 0,

"count" : users count,

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“lastupdate”: 123

“type”:[0/1] 0:user 1:Sam-pros

”avatar”: // option

{

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_145312348.png>

}

“sam\_pros\_info”:{

“service\_category”:“fast food”

}]

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com*/api\_1.0\_contact\_contactListQuery.do*

## Seq-Sync follow list

Seq-Sync follow list表示sync所有关注的公众号列表。当用户登录后，需要每次检查是否要更新本地数据库的关注列表。因为可能用户换手机登录，需要同步最新的关注列表信息到本地数据库。



### Sync follow list request/response json

{

"header":

{

"action" : "follow-list-query"

"token": "token",

},

"body" :

{

}

}

成功返回:

{

"ret" : 0,

"count" : users count,

"users" :[

{

“id”: unique\_id\_in\_samchat

“username”: ”Kevin Dong”

“lastupdate”:123123

”avatar”: // option

{

“thumb:” <http://121.42.207.185/avatar/2016/1/18/thumb_145312348.png>

}

“sam\_pros\_info”:{

“service\_category”:“fast food”

}

favourite\_tag:[0/1] 0:no tag 1:tag

block\_tag:[0/1] 0: no tag 1:tag

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_ officialAccount\_followListQuery.do

### Query User

请参考3.12 read profile

## Seq-Write advertisement

Seq-Create advertisement表示商家发布新的广告。Samchat中的广告和传统的电子广告不同,Samchat 利用聊天的方式实现广告功能，所以广告本身非常的简洁，只有一段话，一张图或一个视频等。在发布新的广告成功后，系统将给整个广告分配一个广告ID，并通知给所有关注此商家的用户。

广告发布/更新分为如下几个步骤:

* A.上传图片或视频到文件服务器S3并获得图片存放URL
* B.上传文字或S3 URL给APP server
* C.服务器分发广告缩略信息给相关用户’

A使用Amazon S3 API来完成。需要定义B和C过程中的API及json数据结构。



### Write advertisement request/response json

{

"header":

{

"action" : "advertisement-write"

"token": "token",

},

"body" :

{

"type":[0/1] 0: text 1:picture 2:vedio

"content": text or url

"content\_thumb": //option url,

”message\_id”: “DASDADNAB12123”

}

}

成功返回:

{

"ret" : 0,

"adv\_id" : ,

"content\_thumb": url

"publish\_timestamp" : 服务器端记录的发布时间

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com*/api\_1.0\_advertisement\_advertisementWrite.do*

### Dispatch advertisement json

{

"header" :

{

"category":" 2"

},

"body" :

{

"id": sender unique\_id\_in\_samchat

"adv\_id" : ,

"publish\_timestamp " : ,

"type":[0/1/2] 0: text 1:picture 2:vedio

"content": text or url

"content\_thumb": url

“dest\_id”: destination user id

}

}

## Seq-Delete advertisement

Seq-Delete advertisement表示商家删除了留在本地的广告，服务器则对这则广告打上tag，这则广告将不在同步。



### Delete advertisement request/response json

{

"header":

{

"action" : "advertisement-delete"

"token": "token",

},

"body" :

{

" advertisements":[{

“adv\_id”: 1

“publish\_timestamp”:12

}]

}

}

成功返回:

{

"ret" : 0

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

广告不存在： {ret: -504}

测试地址：

[http://service-test.samchat.com/api\_1.0\_advertisement\_advertisementDelete.do](http://ec2-54-222-170-218.cn-north-1.compute.amazonaws.com.cn:8081/sam_svr/api_1.0_advertisement_advertisementDelete.do)

## Seq-Send clientId

Seq-Send clientId, 发送手机当前的个推client\_id

### Send clientId request/response json

{

"header":

{

"action" : "send-clientId"

"token": "token",

},

"body" :

{

“client\_id”:”1231”

}

}

成功返回:

{

"ret" : 0

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_profile\_sendClientId.do

## Seq-QueryStateDate

Seq-QueryStateDate, 发送用户列表、客户列表、关注列表的最近更新时间

### Query state date request/response json

{

"header":

{

"action" : "query-state-date "

"token": "token",

},

"body" :

{

}

}

成功返回:

{

"ret" : 0,

“state\_date\_info”:{

“servicer\_list”: 123123,

“customer\_list”: 123123,

“follow\_list”:12312

}

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_profile\_queryStateDate.do

## Seq-QueryPopularRequest

Seq-QueryPopularRequest, 查询热点问题

### Query popular request request/response json

{

"header":

{

"action" : "query-popular-request"

"token": "token"

},

"body" :

{

“count”:50

}

}

成功返回:

{

"ret":0,

"popular\_request":[

{

"content":"121111111"

}

]

}

解析失败返回： {ret: -1}

action参数不支持返回:{ret: -2}

参数不满足返回： {ret: -3}

token 格式不正确：{ret: -4}

内部错误： {ret: -103}

token 不合法： {ret: -401}

测试地址：

http://service-test.samchat.com/api\_1.0\_question\_queryPopularRequest.do

# 附录

## 附录A- 返回值列表

|  |  |
| --- | --- |
| Ret值 | 解释 |
| -1 | 解析失败返回 |
| -2 | action参数不支持返回 |
| -3 | 参数不满足返回 |
| -4 | token 格式不正确 |
|  |  |
| -103 | 内部错误 |
| -201 | 电话号码已经注册过 |
| -202 | 非法电话号码 |
| -203 | 密码错误返回 |
| -204 | 电话号码未注册过 |
| -205 | 验证码错误 |
| -206 | 申请验证码过于频繁 |
| -207 | 此用户不存在 |
| -208 | 密码错误尝试过于频繁 |
| -209 | 查询过于频繁 |
| -210 | 发送邀请过于频繁 |
| -211 | 验证码过期 |
|  |  |
| -401 | token 不合法 |
|  |  |
| -501 | 用户已经有Sam-pros Account |
| -502 | 原始密码错误 |
| -503 | 非商家用户 |
| -504 | 广告不存在 |
| -505 | 等待商家后台审核 |
| -506 | 关注人数超过最大值 |
| -507 | 还未关注此商家 |
| -508 | 还未添加此联系人 |
| -509 | 发送问题太频繁 |
| -510 | 普通用户无法添加普通用户 |
| -511 | 普通用户无法添加商户联系人列表 |