归档编号：

项目代号：

**金美康APP IM**

**内层交互协议**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号：V1.39 | | | |
| 编 制： |  | 日 期： | 2015/6/8 |
| 审 核： |  | 日 期： |  |
| 批 准： |  | 日 期： |  |

目录

[目录 2](#_Toc421548991)

[文件修订记录 4](#_Toc421548992)

[1.说明 5](#_Toc421548993)

[1.1数据体 5](#_Toc421548994)

[1.2 文本大小 5](#_Toc421548995)

[1.3 数据体格式 5](#_Toc421548996)

[1.4请求格式 6](#_Toc421548997)

[1.5响应格式 6](#_Toc421548998)

[1.6推送消息数据体Content内容格式 7](#_Toc421548999)

[1.7请求关键字说明 9](#_Toc421549000)

[2. 群组 11](#_Toc421549001)

[2.1获取群组信息 11](#_Toc421549002)

[2.2获取单个组的详细信息 12](#_Toc421549003)

[2.3获取某个组的成员 13](#_Toc421549004)

[2.4创建组 14](#_Toc421549005)

[2.5修改组信息 15](#_Toc421549006)

[2.6组删除 16](#_Toc421549007)

[2.7删除成员 17](#_Toc421549008)

[2.8添加成员 18](#_Toc421549009)

[2.9申请绑定组 19](#_Toc421549010)

[2.10解绑某个组 20](#_Toc421549011)

[2.11申请加组 21](#_Toc421549012)

[2.12退出组 21](#_Toc421549013)

[3用户 22](#_Toc421549014)

[3.1获取用户信息 22](#_Toc421549015)

[3.2个人信息变更 24](#_Toc421549016)

[3.3头像上传 25](#_Toc421549017)

[3.4特服号码 26](#_Toc421549018)

[3.5修改密码 27](#_Toc421549019)

[3.6更改手机号码 28](#_Toc421549020)

[3.7 苹果后台通知 29](#_Toc421549021)

[4.消息 30](#_Toc421549022)

[4.1推送会话消息结构 30](#_Toc421549023)

[4.2获取单人离线消息 31](#_Toc421549024)

[4.3获取组会话离线消息 32](#_Toc421549025)

[4.4发送一条语音 33](#_Toc421549026)

[4.5发送一条文本消息 35](#_Toc421549027)

[4.6推送组信息变化 36](#_Toc421549028)

[5.邮件类消息 37](#_Toc421549029)

[5.1获取邮件类信息 37](#_Toc421549030)

[5.2处理邀请信息 38](#_Toc421549031)

[5.3邮件消息邀请信息已读回执处理 39](#_Toc421549032)

[5.4推送邮件消息 40](#_Toc421549033)

[5.5邮件推送消息回应 41](#_Toc421549034)

[5.6处理加组申请 41](#_Toc421549035)

## 文件修订记录

|  |  |  |
| --- | --- | --- |
| 日期 | 作者 | 主要修订内容 |
| 2014/12/9 |  | 1.文档重写  2.增加新的类型  3.增加“”  4.修改“3.4特服号码”返回结构 |
| 2014/12/10 |  | 1.修正“”的描述文字  2.修正已定义的所有协议，增加标识，指示为服务端或客户端数据格式 |
| 2014/12/26 |  | 1.更改，将修改手机号码独立出来，进行单独的修改  2.增加协议  3.增加协议  4.增加 |
| 2014/12/30 |  | 修正“”返回值说明 |
| 2015/1/23 |  | 修改“”参数值和说明 |
| 2015/1/26 |  | 修改“”密码的位数长度 |
| 2015/6/8 |  | 1.新增  2.新增协议 |

# 1.说明

## 1.1数据体

一次请求，若有多个数据体，则约定

* 只能存在一个文件数据体，可以有多个文本数据体；
* 除文件数据体以外的其他部分大小不超过2k(2048B)；
* 如果一次请求有多个数据体，则文件数据体放在最后。

。

## 1.2 文本大小

文本数据体最大不超过1k;

单个语音文件最大不超过64k;

单个图片文件尽量200k以内(尚未约定)

## 1.3 数据体格式

数据体包含2种，一种是文件数据体，一种是文本数据体

* 文件数据体格式:

文件id(4B) +文件具体内容(字节数组）

发送文件时，分为两步，第一步是鉴权，但目前不必等鉴权回应，第二步才是发送具体的语音，发送语音，两步之间通过同一个uuid进行关联，标识为一条完整的语音

* 文本数据体格式

json字符串。与发送文件不一样，发送一条文本消息不需要执行鉴权的步骤，直接执行发送即可，发送的文本内容都放置在这里面。

## 1.4请求格式

* 请求格式:单个，仅为语音或文字(目前采用此格式)

段数量(2B)+预留字段(4B)+数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体

* 请求格式:多个(段数限定目前最大为16)

段数量(2B)+预留字段(4B)+数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体 +数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体

-------目前环境下：语音和文字不能同时发送

## 1.5响应格式

每个请求以json串的形式放入到数据体中

* 单个:

段数量(2B)+预留长度(4B)+数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体

* 多个：(目前约定响应单个，暂不采用多个，如果一个消息包含多条内容，分拆开成多条返回)

段数量(2B)+预留长度(4B)+数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体+数据类型(1B)+数据子类型(1B)+数据体长度(4B)+数据体

说明:

* 数据类型

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| 0x1 | 文本类型 |
| 0x2 | 文件类型 |

* 文本类型

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| 0x1 | json串 |
| 0x2 | xml |
|  |  |

* 文件类型

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| 0x1 | 语音 |
| 0x2 | 图片 |
|  |  |

## 1.6推送消息数据体Content内容格式

从服务器推送到客户端的消息，是存放在响应格式的数据体中，数据体采用文本类型。目前消息分为两类：一类是邮件类消息，一类是会话类消息

内容见下表

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| email | 推送的是邮件类型消息 |
| meet | 推送的非邮件类型的语音或者文本 |
| group\_change | 推送组变化,客户端收到通知应当重新获取组的成员或信息 |

推送的会话数据结构见推送会话消息结构 ，而邮件的数据结构见推送邮件消息

它们对应的内容格式皆采用 [关键字]内容[/关键字]

* 推送会话消息关键字见下表

|  |  |
| --- | --- |
| 关键字标识 | 说明 |
| url | 语音url地址  示例:[url]www.baidu.com[/url] |
| text | 文本消息  示例:[text]最近怎么样呢?[/text] |
| warning | 警告,关键字之间放置值，不同值之间使用分号作为分隔符  1.警告类型：(用以做扩展);  0:目前填0  2.经纬度：经纬度之间用逗号分隔  3.组id :发往哪个组  完整示例:  [warning]0;30.1234,130.1234;12[/warning] |
| key | 语音或其他文件的解密key，解密文件用  示例:  [key]DDFRWEC[/key] |
| image | 内部图片消息示例:  [image]http://192.168.1.1:8080/abc.bin[/image] |
| picture | 外部图片(外链接到非自身服务器以外的地方)  [picture]http://www.baidu.com/abc.jpg[/picture] |

* 推送邮件关键字见下表

|  |  |
| --- | --- |
| 关键字标识 | 说明 |
| invite | 表示该邮件类型是一个邀请,invite直接承载的是组id,表示是哪个组的邀请.  示例:[invite]1[/invite] |
| apply | 表示该邮件类型是一个申请，applay直接承载的是组id(gid)和用户id(uid)，表示的是谁申请加入到哪个组里.gid和uid顺序不能变。  示例:[apply]11,22[/applay] |

## 1.7请求关键字说明

* 协议中涉及到时间戳的地方，统一时间单位为秒(s)
* 所有的数据类型标识采用小端格式进行传输
* 调用失败时，返回的统一结构为: { error: '错误信息' }
* type关键字参见下表

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| groups | 获取一个用户的所有群组 |
| group\_detail | 获取一个组的详细信息 |
| members | 获取一个组中成员 |
| account | 获取用户的账号信息 |
| messages | 获取其他用户对当前用户的留言消息 |
| messages\_group | 获取某个组的离线消息 |
| authorize | 发送一条数据前，需要向服务器确认，确认之后发送语音数据 |
| send\_msg | 发送一条文本消息 |
| email\_invite | 处理服务端推送过来的邀请类邮件信息，拒绝/接受/忽略 |
| email\_status | 回执一条邮件消息的状态为已读(邮件消息至少具有已读，未读两个状态) |
| push\_meet | 回应服务端，一条会话(语音，文本)消息 |
| push\_email | 推送email回应服务端 |
| push\_group | 推送组变化通知客户端 |
| get\_ email | 获取邮件消息(邮件消息与普通消息不同) |
| group\_add | 登录用户增加组，如果已有组则创建不成功 |
| modify\_account | 修改当前用户信息 |
| modify\_avatar | 修改头像 |
| modify\_group | 修改组信息 |
| del\_group | 删除组 |
| del\_member | 删除组成员 |
| add\_member | 添加组成员 |
| apply\_group | 申请加入组 |
| quit\_group | 离开某个组 |
| get\_ rescue | 获取特服号码列表 |
| bind\_group | 某个用户绑定到某个组 |
| unbind\_group | 某个用户从某个组解绑 |
| email\_apply | 处理服务端推送过来的申请类邮件信息，拒绝/接受/忽略 |
| modify\_psd | 请求修改密码 |
| modify\_tele | 请求修改手机号码第一步 |
| modify\_phone | 请求修改手机号码第二步 |
| match\_phone | 请求匹配通讯录手机号(暂未使用) |
| ios\_bg\_notify | ios上应用退到后台时，需要发送通知给服务器(ios专用) |

* 推送消息类型，对应json串的key为pushtype，其对应的value取值参见下表

|  |  |
| --- | --- |
| 类型标识 | 类型说明 |
| email | 推送的是邮件类型消息 |
| meet | 推送的非邮件类型的语音或者文本,原warning合并到meet类型 |

# 2. 群组

## 2.1获取群组信息

说明：请求用户所在的群组，及其成员

* 请求参数

(长连接，用于标识信息的seq和标识用户的标志都另一层实现，会与服务端维持该链接，因此只需要区别请求类型)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | √ | 请求标识，请求类型 |

* 客户端请求格式:

{

"type":"groups"

}

* 服务端数据返回值格式:

{

"groups": [

{

"gid": 1,

"owner":1, //创建者id

"gname": "group1",

"createtime": 31221233,

"etag":1,

},

{

"gid": 2,

"owner":1, //创建者id

"gname": "group2",

"createtime": 31221233,

"etag":2

}

]

}

说明：

gid int 组id

owner int 创建者id

gname String 组名

etag int 组标识，如果组中内容发生变化，此etag将会改变

createtime long 时间戳(精确到ms)

## 2.2获取单个组的详细信息

说明：获取组信息

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gId | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type"," group\_detail ",

"gid": 1

}

* 服务端响应返回值格式:

{

"grounp": {

"gid": 1,

"gname": "group1",

"createtime": 31221233,

"etag":1

}

}

说明：

gid int 组id

gname String 组名

createtime long 组创建时间戳 时间戳(ms)

etag int 组标识，如果组中内容发生变化，此etag将会改变

## 2.3获取某个组的成员

说明：获取组信息

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gId | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type": "members",

"gid": 1

}

* 服务端返回值格式:

{

"gid": 222,

"members": [

{

"uid": 1,

"nickname": "sss",

"createtime": 31221233

},

{

"uid": 2,

"nickname": "ssss",

"createtime": 31221233

},

{

"uid": 2,

"nickname": "ssss",

"createtime": 31221233

}

]

}

说明：

gid int 组id

members array 组成员列表

uid int 用户id

nickname String 用户昵称

## 2.4创建组

说明:一个手机号可以对应多个账户，一个帐号只能创建一个组

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | √ | 请求类型 |
| gname | String(16B) | √ | 组名称 |

* 客户端请求格式:

{

"type":" group\_add "，

" gname ":"新组",

}

* 服务端响应返回值格式:

{

"gid":1,

"flag":" 1

"createtime": 31221233,

"etag":1

}

gid int 创建好的组的id

flag int 创建是否成功;1创建组成功;0创建失败；-1已经创建1个组

createtime long 组创建时间戳 时间戳(ms)

etag int 组初始tag

## 2.5修改组信息

说明：修改一个组的信息，目前一个组的信息主要有：组id，组名，拥有者，创建时间，组标识etag，能修改的仅为组名

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gId | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |
| gname | String(16B) | 必填 | 组名 |

* 客户端请求格式：

{

"type"," modify\_group",

"gid": 1,

"gname":"新组名"

}

* 服务端返回值格式:

{

"gid": 1,

"flag":1

"owner":1,

"gname": "新组名",

"createtime": 31221233,

"etag":1

}

说明：

gid int 组id

flag int 修改组是否成功；1修改组成功，0修改组失败

owner int 拥有者id

gname String 组名

createtime long 组创建时间戳 时间戳(ms)

etag int 组标识

## 2.6组删除

说明：删除一个组，的信息，必须是当前用户创建的组才能删除

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gId | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type"," del\_group ",

"gid": 1,

}

* 服务端返回值格式:

{

"gid": 1,

"flag":" 1

}

说明：

gid int 组id

flag int 操作标识 //操作返回值，1表示组删除成功，0表示失败;-1无权删除该组;-2组不存在

## 2.7删除成员

说明：删除组中的成员，必须是当前用户创建的组才能删除

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gId | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |
| uid | int | 是 | 要删除的成员id |

* 客户端请求格式：

{

"type"," del\_member ",

"gid": 1,

"uid":12

}

* 服务端返回值格式:

{

"gid": 1,

"flag":" 1

}

说明：

gid int 组id

flag int 操作标识 //操作返回值，1表示删除成员成功，0表示失败;-1不能删除创建者;-2成员已经被删除

## 2.8添加成员

说明:必须是组的创建者才能为改组增加成员，增加成员之前是通过网页web端发送邮件邀请信息，用户接收到邮件并选择是否加入组。现在由客户端发起。接收邀请信息的方式仍然是邮件消息

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gid | Int | √ | 组id |
| type | String(16B) | √ | 请求类型 |
| uid | int | √ | 用户id |

* 客户端请求格式：

{

"type"," add\_member ",

"gid": 1,

"uid":1

}

* 服务端响应返回值格式:

{

"gid": 1,

"flag":" 1

}

说明：

gid int 组id

flag int 操作标识 //发送邀请返回值，1表示添加邀请成功，0表示失败；-1添加失败，成员已经在组中;

## 2.9申请绑定组

用户主动发起请求申请绑定到某个组，但不需要通过组创建者或管理者的审核 。一个用户只能绑定一个组。 绑定组和成为一个组的组成员没有必然关系，绑定的组将作为成员的一个属性。(绑定组和解除绑定都不需要组创建者或管理者审核)

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gid | int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type","bind\_group ",

"gid": 1,

}

* 服务端响应返回值格式:

{

"gid": 1,

"flag":" 1

}

说明：

gid int 绑定的组id

flag int 操作结果id. //发送申请是否成功，1表示成功，0绑定失败; -1绑定异常，已经绑定了该组

## 2.10解绑某个组

用户主动发起请求申请取消到某个组的绑定，解绑一个组和是否脱离一个组的组(不再是组的成员)没有必然关系， (绑定组和解除绑定都不需要组创建者或管理者审核

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gid | int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type","unbind\_group ",

"gid": 1,

}

* 服务端响应返回值格式:

{

"gid": 1,

"flag":" 1 //发送解绑是否成功

}

说明：

gid int 绑定的组id

flag int 操作结果id， 1表示成功， 0绑定组失败; -1解绑定异常，在本次解绑定之前已经解除了绑定; -2解绑定异常，组已经被删除，

## 2.11申请加组

用户主动发起请求申请加入到某个组，当组的创建者或管理者审核通过后，用户才成为该组的成员

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gid | int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式：

{

"type"," apply\_group",

"gid": 1,

}

* 服务端响应返回值格式:

{

"gid": 1,

"flag":0 ,

}

说明:

gid int 组id

flag int 操作结果，1申请成功，等待审核;0申请失败,；-1申请的组不存在 ; -2重复申请,即已经在某个组里

## 2.12退出组

用户主动脱离某个组，其前提是它已是该组的成员。退出某个组不需要组的创建者或管理者审核。一个组的创建者无法退出他自身创建的组

* 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| gid | int | √ | 组id |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式:

{

"type":" quit\_group ",

"gid":1

}

* 服务端响应返回值格式:

{

"gid": 1,

"flag":0

}

说明:

gid int 组id

flag int 操作结果，1 退出成功；0退出失败，已经退出了该组;-1组不存在

# 3用户

## 3.1获取用户信息

说明：请求一个用户的基本信息.其中用户的头像需要通过http请求才能获取到最终的头像文件，这里只返回它的地址

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | √ | 请求类型 |

* 客户端请求格式:

{

"type":"account"

}

* 服务器响应返回值格式:

{

"uid": 1,

"roleid":1,

"phone":"123455",

"createtime": 122234,

"nickname": "ddddd"

"height":175,

"weight":45.5

"old":18,

"sex":1

"gid":11//当前用户所绑定的组

"avatar":"192.168.1.2/avatar/a.jpg"

}

说明：

uid int 用户id

roleid int 角色id

phone String 角色的手机号码

nickname String 用户昵称

createtime long 创建的时间

height int 身高，单位cm

weight float 体重 ，单位kg

old int 年龄

sex int 性别，1男，0女

avatar String 头像地址

gid int 当前用户所绑定的组

## 3.2个人信息变更

说明:修改用户的信息，包括昵称，身高，体重，年龄等信息的修改和调整.。提交信息时，每次提交所有请求参数都必须填写，这样的问题在于重复提交数据

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | √ | 请求类型 |
| uid | int | √ | 用户id |
| nickname | String | √ | 用户昵称 |
| height | int | √ | 身高 |
| weight | float | √ | 体重 |
| old | int | √ | 年龄 |
| sex | int | √ | 性别 |

* 客户端请求格式:

{

"type":" modify\_account",

"uid ":123,

"nickname":"新昵称",

"height":"175",

"weight":"65.5",

"old":19,

"sex":1,

}

* 服务端响应返回值格式:

{

"uid":123,

"flag":" 1 //操作返回值，1表示成功，0表示失败

}

说明:

uid int 用户id

flag int 操作结果，/操作返回值，1表示成功，0表示失败

## 3.3头像上传

头像的提交采用文件格式单独发送，类似于发送消息中发送文件。如果用户修改了头像，则发送分为2步，第一步为上传当前图片的md5值，当收到服务器回应时,若上传的该图片是预定义图片，则不需要上传图片本身，若上传的该图片是自定义图片，且服务器回应不存在时，第二步再发送对应的图片文件。发送文件的方式同发送语音的第二步相同

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| uuid | int | 是 | 唯一标识一次 |
| type | String(16B) | 是 | 标识请求 |
| md5 | string(16B) | 是 | 文件md5 |
| picindex | int | 是 | 文件索引，主要是关联预定义的图片 |

* 客户端请求格式:

{

"type":" modify\_avatar ",

"uuid ":123,

"picindex":123,

"md5":"A3435DF"

}

* 服务端响应返回值格式:

{

"uuid":123, //唯一标识一次请求

"picindex":123,//图片索引

"flag": -1 //操作返回值，-1表示图片已存在,1成功

}

uuid int 唯一标识一次上传头像

picindex int 图片索引，若是预定义的图片，则该值小于1000

flag int 本次请求返回值，若是上传预定义和自定义的图片成功，则返回1;若图片已经存在(自定义的图片才会返回，预)，则返回-1。

* 第二步发送文件

服务端响应客户端发送头像的返回值格式:

{

"uid ":123,

"avatar ":12234,//头像地址

}

## 3.4特服号码

当遇到告警时，若用户在线则通过数据网络发送告警信息，若未在线或者网络不可用此时通过特服号码发送短信。

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |

* 客户端请求格式:

{

"type ": "get\_ rescue",

}

* 服务端响应返回值格式:

{

"phones": [

{

" phonenumber ": "111111"

},

{

" phonenumber ":"2222222"

}

]

}

## 3.5修改密码

用户需要修改当前帐号的密码，需要在有网络的环境下使用。修改一次之后需要隔一小时再改.由于密码是经过加密之后形成了字节流。字节序无法直接放置到json字符串中，因此需要密码经过BCD编码后再放置到其中。

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| oldpsd | String(32B) | 是 | 旧密码 |
| newpsd | String(32B) | 是 | 新密码 |

* 客户端请求格式:

{

"type ": "modify\_psd",

"oldpsd":"1111",

"newpsd":"1111"

}

* 服务端响应返回值格式:

{

"flag": 1

}

说明:

flag int 操作结果，操作返回值，1表示成功，-1表示失败，原密码不正确;-2表示失败，修改过于频繁；0失败，其他错误

## 3.6更改手机号码

用户修改手机号码。需要通过两个步骤实现(1).输入密码和新的手机号码(客户端需要验证一下手机号码的格式)，服务器收到请求后，短信发送验证码到新的手机(2)填写短信中的验证码发送到服务器

步骤1:

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| psd | String(32B) | 是 | 密码 |
| phone | String(16B) | 是 | 新的手机号码 |

* 客户端请求格式:

{

"type ": "modify\_tele",

"psd":"1111"，

"phone":"133322343"

}

* 服务端响应返回值格式:

{

"flag": 1

}

说明:

flag int 操作结果，操作返回值，1表示成功，0表示密码错误引起的失败

步骤2

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| code | String(16B) | 是 | 短信验证码 |

* 客户端请求格式:

{

"type ": "modify\_phone",

"code":"1111"

}

* 服务端响应返回值格式:

{

"flag": 1

}

说明:

flag int 操作结果，操作返回值，1表示成功，0表示验证码错误引起的失败

## 3.7苹果后台通知

主要针对ios处理。在ios上当程序被退到后台时，需要告知服务器，服务器需要将消息通过app store进行推送。

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| device | String(64B) | 是 | ios唯一识别码 |

* 客户端请求格式:

{

"type ": " ios\_bg\_notify ",

" device ":" 0d943976b24c85900c764dd9f75ce054dc5986ff",

}

* 服务端响应返回值格式:

{

"flag": 1

}

说明:

flag int 操作结果，操作返回值，1表示成功，0失败，其他错误

# 4.消息

## 4.1推送会话消息结构

服务器在长连接模式下，可以向服务器主动推送消息，消息的数据体格式如下，如果服务器存在多条消息，将按照之前已经定义好的响应格式(单条)一条一条回应

* 服务端主动推送消息格式:

{

"uuid":1, //消息id

"sendtime": 2323443,

"content": "[url]www.baidu.com[/url]"

"pushtype":"meet"

"fromid":1,//谁发送

"gid":1, // gid为0 ，表示是发给个人的，不为0是发给某个组的

}

说明：

uuid int 消息id

content String 消息的文本内容，需要符合

sendtime long 消息发送时间戳(精确到秒或者ms)

pushtype String 推送消息类型，类型为 meet

fromid int 谁发送此条语音

gid int 语音发向到哪个组，为0时表示发送给个人

客户端回应推送消息。一条推送消息推送到客户端之后，客户端需要处理并给定一个回应，告诉服务端已经收到消息该消息。

* 客户端回应格式:

{

"uuid":1

"type":" push\_meet "

"result":200

}

uuid int 消息的id,用来唯一标识一条消息

type string 推送消息类型，用来区别属于哪一类推送消息push\_meet

result int 推送结果，客户端统一回应200

## 4.2获取单人离线消息

说明：主动向服务器请求离线消息，获取消息的方式有些类似于分页获取，服务器收到请求之后，将先回应服务端的总数据页数，此后通过推送将消息一条条推送下来。服务器在推送消息后，会从数据库中移除数据，因此请求的page页数依然从第一页开始。由于在请求消息时，服务端没有返回消息的id列表，因此本地需要做一个定时，在该时间内记录已获取的条数，当条数达到后，再发起第二次请求；若在超时时间范围内都没有接受到足够数量的数据，则再发起第二次请求。(区分离线消息和非离线消息根据消息id区别，id &0x80000000==0为离线消息)

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 请求类型 |
| number | int | 是 | 获取多少条离线消息 |
| page | int | 是 | 存在多个页时，返回第几页的消息 |

* 客户端请求格式:

{

"type":"messages" ,

"page":1

"number":20

}

* 服务端响应返回值格式:

{

"page":1

"number":20

"total":20

}

说明：

page int 当前的页数

number int 每页包含的消息条数

total int 服务端总共有多少条消息

## 4.3获取组会话离线消息

说明：主动向服务器其他用户对当前用户的离线消息，获取消息的方式有些类似于分页获取，与上述逻辑相同

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 请求类型 |
| number | int | 是 | 获取多少条离线消息 |
| page | int | 是 | 存在多个页时，返回第几页的消息 |
| gid | int | 是 | 获取对应的组的id |

* 客户端请求格式:

{

"type":"messages\_group"

"page":1

"gid":1

"number":20

}

* 服务端响应返回值格式:

{

"gid":1

"page":1

"number":20

"total":20

}

说明：

gid int 组id

page int 当前的页数

number int 每页包含的消息条数

total int 服务端总共有多少条消息

## 4.4发送一条语音

消息时需要两个步骤，第一个步骤是发送消息进行类似于通话请求，姑且称之为授权，目前第二个步骤的执行，暂时不用第一个步骤的回应。但需要注意两个步骤都需要带上同样的标识

说明：发送一条语音消息第一个步骤，鉴权(目前可以不用等待鉴权回应就开始第二个步骤)，第二个步骤不需要添加请求type标识，在请求中包含与第一步相同的uuid即可， 第二个步骤是发送文件，需要等待文件是否发送成功的回应，回应格式见第2个步骤返回值

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| uuid | int | 是 | 唯一标识一条消息 |
| type | String(16B) | 是 | 标识请求 |
| fromtype | int | 是 | fromtype为0，表示是发给个人之间的；fromtype为1表示是发给组的 |
| toid | int | 是 | toid发给谁，当fromtype是0时，此次填用户id，反之则填组id |

* 客户端请求格式:

{

"uuid": 1,

"type":" authorize",

"fromtype":0, //为0表示个人之间的，1 表示为发给某个组的 "toid":2 //用户id或组id

}

* 服务端响应步骤1返回值格式:

{

"uuid": 1,

"flag": 1, //是否授权1授权，0未授权

"sendtime":122314// 服务器记录的时间

}

说明：

uuid int 标识一条消息请求(与seq不同)

flag int 标识是否授权

sendtime long 发送时间

* 服务器响应步骤2返回值格式:

{

"uuid": 1,

"sendtime":122314// 服务器记录的时间

}

uuid int 标识一条消息请求(与seq不同)

sendtime long 服务器记录到的时间

## 4.5发送一条文本消息

说明：发送文本消息不需要鉴权，一步即可完成。发送文本消息，内容content需要按照“”

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| uuid | int | 是 | 唯一标识一条消息 |
| type | String(16B) | 是 | 标识请求 |
| fromtype | int | 是 | fromtype为0，表示是发给个人之间的；fromtype 为1表示是发给组的 |
| toid | int | 是 | toid发给谁，当fromtype是0时，此次填用户id，反之则填组id |
| content | String(1024B) |  | 消息内容 |

* 客户端请求格式:

{

"uuid": 1,

"type":"send\_msg ",

"fromtype":0 //0表示个人，1则表示给组的

"toid":11, //组或者用户id

"content":"[text]内容[/text]"

}

* 服务端响应返回值格式:

{

"uuid": 1,

"flag": 1, //是否发送成功1,成功，0失败

"sendtime":12344 //发送时间

}

说明

uuid int 消息标识

flag int 发送是否成功

sendtime long 发送时间

## 4.6推送组信息变化

服务器在长连接模式下，可以向服务器主动推送消息，当某个组的成员或者信息发生了变化，将推送该消息到客户端，客户端收到推送消息时，获取最新的组的成员和信息。消息的数据体格式如下

* 服务端推送组信息变化的消息格式:

{

"uuid":12445

"sendtime": 2323443,

"pushtype":" group\_change "

"gid":1, // 组id

"etag":111

}

说明：

uuid int 消息id

sendtime long 消息发送时间戳(精确到秒)

pushtype String 推送消息类型，group\_change

gid int 组id，哪个组发生了变化

etag int 新的etag

客户端对推送消息的回应。一条推送消息推送到客户端之后，客户端需要处理并给定一个回应，告诉服务端已经收到消息该消息。

* 客户端回应推送消息格式:

{

"uuid":1

"type":" push\_group "

"result":200

}

uuid int 消息的id,用来唯一标识一条消息

type string 推送消息类型，用来区别属于哪一类推送消息

result int 推送结果，客户端统一回应200

# 5.邮件类消息

## 5.1获取邮件类信息

(用来获取被用户忽略的未处理的消息,邀请，通知之类的

说明： 该消息是被服务器主动推送到客户端的，推送之后，用户若未处理，则还可以请求获取到,请求之后邮件消息通过推送到达客户端,格式和“”基本相同

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 请求类型 |
| number | int | 是 | 获取多少条离线消息 |
| page | int | 是 | 存在多个页时，返回第几页的消息 |

* 客户端请求格式 :

{

"type":" get\_ email " ,

"page":1

"number":20

}

* 服务端响应返回值格式:

{

"page":1

"number":20

"total":20

}

说明：

page int 当前的页数

number int 每页包含的消息条数

total int 服务端总共有多少条消息

## 5.2处理邀请信息

说明：请求处理邀请类别的推送消息

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| gid | int | 是 | 标识哪一个组的邀请 |
| flag | int | 是 | 0表示拒绝，1表示接受 |

* 客户端请求格式:

{

"type":" email\_invite ",

"gid":1,

"flag":1

}

* 服务端响应返回值格式:

{

"flag":" 1 //操作返回值，1表示成功，0表示失败

}

## 5.3邮件消息邀请信息已读回执处理

说明：回执改变消息是否已读的状态.服务端收到此回执，无需回应客户端

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| uuid | int | 是 | 消息id |
| gid | int | 是 | 组id |

* 客户端请求格式:

{

"type":" email\_status ",

"uuid":1,

"gid":1

}

* 服务端响应返回值格式:

无需回应

## 5.4推送邮件消息

邮件类消息也会通过推送到达客户端，到达客户端之后，则本次推送完成，若未到达或其他原因，用户还通过上述邮件类消息“”主动请求。

* 服务端推送邮件消息的结构:

{

"uuid": 1,

"title":"消息标题"

"content": "[invite]1[/invite] ",

"sendtime": 2323443,

fromid:1//谁发送

"pushtype":" email ", //邮件类型消息

}

说明：

uuid int 消息的id

title String 消息的标题

content String 消息的文本内容,格式参见“”

sendtime long 消息发送时间戳(精确到秒或者ms)

pushtype String 推送类型类型消息

fromid int 谁发的邀请

## 5.5邮件推送消息回应

一条推送消息推送到客户端之后，客户端需要处理并给定一个回应，告诉服务端已经收到消息该消息

* 客户端回应推送消息的格式:

{

"uuid": 1,

"type":"push\_email"

"result":200

}

uuid int 消息的id，用来唯一标识消息

type string 推送消息类型，用来区别属于哪一类推送push\_email

result int 推送结果，客户端统一回应200

## 5.6处理加组申请

说明：处理某个人发起的加组申请，组创建者同意或拒绝或忽略申请，其中不做任何处理即为忽略。

* 请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否必填 | 说明 |
| type | String(16B) | 是 | 标识请求 |
| uid | int | 是 | 标识是谁申请加入组 |
| gid | int | 是 | 标识申请到添加到哪一个组 |
| flag | int | 是 | 0表示拒绝，1表示接受 |

* 客户端请求格式:

{

"type":" email\_apply ",

"uid":2,

"gid":1,

"flag":1

}

* 服务端响应返回值格式:

{

"flag":" 1 //操作返回值，1表示成功，0表示失败

}