

- [1、简介](#)
  - [1.1、说明](#)
  - [1.2、UI展示](#)
  - [1.3、开源源码](#)
  - [1.4、术语表](#)
- [2、API概述](#)
  - [2.1、Initialization](#)
    - [initEasemob\(param\) 初始化](#)
    - [login\(param\) 登录](#)
    - [cbLogin\(param\) 登陆回调](#)
    - [logout\(\) 退出登录](#)
    - [registerUser\(param\) 注册](#)
    - [cbRegisterUser\(param\) 注册回调](#)
    - [updateCurrentUserNickname\(param\) 更新当前用户的昵称](#)
    - [getLoginInfo\(\) 获取当前登陆信息\(仅iOS可用\)](#)
    - [cbGetLoginInfo\(param\) 获取当前登陆信息的回调\(仅iOS\)](#)
    - [onConnected\(\); 已连接上](#)
    - [onDisconnected\(param\) 链接断开](#)
  - [2.2、Message](#)
    - [onNewMessage\(param\) 收到新消息监听](#)
    - [onCmdMessageReceive\(param\) 透传消息监听](#)
    - [onAckMessage\(param\) 消息已读监听](#)
    - [onDeliveryMessage\(param\) 消息送达监听](#)
    - [sendText\(param\) 发送文本消息及表情](#)
    - [sendFile\(param\) 发送文件](#)
    - [sendPicture\(param\) //发送图片](#)
    - [sendLocationMsg\(param\) //发送地理位置信息](#)
    - [sendVoice\(param\) //发送语音](#)
    - [sendVideo\(param\) 发送视频](#)
    - [sendCmdMessage\(param\) 发送透传消息](#)
    - [setNotifyBySoundAndVibrate\(param\) 消息提醒相关配置](#)
    - [getMessageById\(param\) 根据id获取消息记录](#)
    - [cbGetMessageById\(param\) 得到一条消息记录](#)
    - [sendHasReadResponseForMessage\(param\) 发送消息已读回执](#)
    - [onMessageSent\(param\) 消息已发送监听](#)
  - [2.3、Conversation](#)
    - [getConversationByName\(param\) 根据用户名获取conversation对象](#)
    - [cbGetConversationByName\(param\) 回调](#)
    - [getMessageHistory\(param\) 获取聊天记录](#)
    - [cbGetMessageHistory\(param\) 获取聊天记录回调](#)

- [getUnreadMsgCount \(param\)](#) 获取未读消息数量
- [cbGetUnreadMsgCount \(param\)](#) 获取未读消息数量回调
- [resetUnreadMsgCount \(param\)](#) 指定会话未读消息数清零
- [resetAllUnreadMsgCount \(\)](#); 所有未读消息数清零(仅Android可用)
- [getMsgCount \(param\)](#) 获取消息总数(仅Android可用)
- [cbGetMsgCount \(param\)](#) 获取消息总数回调(仅Android可用)
- [clearConversation \(param\)](#) 清空会话聊天记录(仅Android可用)
- [deleteConversation \(param\)](#) 删除和某个user的整个的聊天记录(包括本地)
- [removeMessage \(param\)](#) 删除当前会话的某条聊天记录
- [deleteAllConversation \(\)](#); 删除所有会话记录(包括本地)
- [getChatterInfo \(\)](#); 获取聊天对象信息
- [cbGetChatterInfo \(param\)](#); 获取聊天对象信息回调
- [getTotalUnreadMsgCount \(\)](#); 获取总计未读消息数
- [cbGetTotalUnreadMsgCount \(param\)](#); 获取总计未读消息数回调
- [getRecentChatters \(\)](#); 获取近期聊天对象信息
- [cbGetRecentChatters \(param\)](#);

#### ◦ 2.4、Friend

- [onContactAdded \(param\)](#)//新增联系人监听
- [onContactDeleted \(param\)](#)//删除联系人监听
- [onContactInvited \(param\)](#)//接到好友申请
- [onContactAgreed \(param\)](#)//好友请求被同意
- [onContactRefused \(param\)](#)//好友请求被拒绝
- [getContactUserNames \(\)](#);//获取好友列表
- [cbGetContactUserNames \(param\)](#)//获取好友列表回调
- [addContact \(param\)](#)//添加好友
- [deleteContact \(param\)](#)//删除好友
- [acceptInvitation \(param\)](#)//同意username的好友请求
- [refuseInvitation \(param\)](#)//拒绝username的好友请求
- [getBlackListUsernames \(\)](#);//获取黑名单列表
- [cbGetBlackListUsernames \(param\)](#)//获取黑名单列表回调
- [addUserToBlackList \(param\)](#)//把用户加入到黑名单
- [deleteUserFromBlackList \(param\)](#)//把用户从黑名单中移除

#### ◦ 2.5、Group

- [onInvitationDeclined \(param\)](#)//群聊邀请被拒绝
- [onInvitationAccepted \(param\)](#)//群聊邀请被同意
- [onUserRemoved \(param\)](#)//当前用户被管理员移除出群聊
- [onGroupDestroy \(param\)](#)//群聊被创建者解散
- [onApplicationReceived \(param\)](#)//用户申请加入群聊,收到加群申请
- [onApplicationAccept \(param\)](#)// // 加群申请被同意
- [onApplicationDeclined \(param\)](#)//加群申请被拒绝
- [createPrivateGroup \(param\)](#)//创建私有群
- [createPublicGroup \(param\)](#)//创建公开群
- [onGroupCreated \(param\)](#)//群组建立监听

- `addUsersToGroup(param)` // 群聊加入
- `removeUserFromGroup(param)` // 群聊减人
- `joinGroup(param)` // 加入某个群聊, 只能用于加入公开群
- `exitFromGroup(param)` // 退出群聊
- `exitAndDeleteGroup(param)` // 解散群聊
- `getGroupsFromServer(param)` // 从服务器获取自己加入的和创建的群聊列表
- `cbGetGroupsFromServer(param)` // 从服务器获取自己加入的和创建的群聊列表回调
- `getAllPublicGroupsFromServer(param)` // 获取所有公开群列表
- `cbGetAllPublicGroupsFromServer(param)` // 获取所有公开群列表回调
- `getGroup(param)` // 获取群详情
- `cbGetGroup(param)` // 获取群详情回调
- `blockGroupMessage(param)` // 屏蔽群消息
- `unlockGroupMessage(param)` // 解除屏蔽群
- `changeGroupName(param)` // 修改群组名称
- `setReceiveNotNoifyGroup(param)` // 群聊不提醒只显示数目(仅Android可用)
- `blockUser(param)` // 将群成员拉入群组的黑名单
- `unlockUser(param)` // 将拉入黑名单的群成员移除
- `getBlockedUsers(param)` // 获取群组的黑名单用户列表
- `cbGetBlockedUsers(param)` // 获取群组的黑名单用户列表回调
- `onGroupUpdateInfo(param)` // 群组信息更新的监听(仅iOS)
- `onDidJoinedGroup(param)` // 自动加入群组监听
- `onReceiveGroupInvitation(param)` // 收到群组邀请监听
- `acceptJoinApplication(param)` // 批准入群申请, 需要Owner权限
- `declineJoinApplication(param)` // 拒绝入群申请, 需要Owner权限
- `acceptInvitationFromGroup(param)` // 接受入群邀请
- `declineInvitationFromGroup(param)` // 拒绝入群邀请

#### ◦ 2.6、Call

- `onCallReceive(param)` // 实时语音监听
- `onCallStateChanged(param)` // 通话状态监听
- `makeVoiceCall(param)` // 拨打语音通话
- `answerCall()` // 接听通话
- `rejectCall()` // 拒绝接听
- `endCall()` // 挂断通话

#### ◦ 2.7、Apns(以下方法全部仅限iOS)

- `registerRemoteNotification()` // 注册Apns推送
- `cbRegisterRemoteNotification(param)` // 回调
- `onApnsLaunch(param)` ;
- `updatePushOptions(param)` // 设置apns全局属性
- `cbUpdatePushOptions(param)` // 设置apns全局属性回调
- `ignoreGroupPushNotification(param)` // 设置指定群组是否接收
- `cbIgnoreGroupPushNotification(param)` // 回调

#### • 3、附录

- [EMMessage\\_json字符串返回值结构](#)
- [普通文本消息](#)
- [透传消息](#)
- [位置消息](#)
- [视频/语音/图片/文件消息](#)
- [EMConversation\\_json字符串返回值结构](#)
- [EMGroup\\_json字符串返回值结构](#)
- [EMChatterInfo\\_json字符串返回值结构](#)
- [“isGroup”参数废弃 改用“chatType”的相关说明](#)
- [4、更新历史](#)
  - [iOS](#)
  - [Android](#)
- [5 文档更新记录](#)

## 1、简介 官

环信插件

### 1.1、说明

本插件是基于环信API封装的AppCan平台的插件模块, 用户可以使用本插件实现基本的即时通讯功能, 包括——

单聊功能: 支持发送语音, 图片, 表情, 文字, 位置, 附件;

群聊功能: 支持500人到2000人大群, 拥有完善的群组权限管理;

实时语音 : 基于IP网络的点对点实时语音, 适应低带宽要求;

#### 使用前说明:

本插件为单例插件 ——

- 在任何网页调用本插件, 调用的是同一个插件实例;
- 所有的API都是异步方法, 不会直接返回值;
- 所有的回调都会传到root页面 (config.xml中配置的App起始页面)

以上内容非常重要

root页面收到回调后, 可以通过uexWindow的相关方法传递到各个网页去,

以下方法是您可能要用到的——

传递给某个窗口：

- `uexWindow.evaluateScript`;
- `uexWindow.evaluatePopoverScript`
  
- `uexWindow.evaluateMultiPopoverScript`

传递给某些窗口：

- `uexWindow.publishChannelNotification`
  
- `uexWindow.subscribeChannelNotification`

这些方法具体用法在[uexWindow文档](#) 内有描述

当然, 也可[下载Demo](#) 参考Demo内的调用。

## 1.2、 UI展示

暂无

## 1.3、 开源源码

插件测试用例与自定义插件下载：[点击此处](#)（插件测试用例与插件源码已经提供）

## 1.4、 术语表

Path Types

协议头	Android对应路径（其中” /sdcard/ “等同于” /storage/emulated/0/ “）	i
<code>res://</code>	<code>widget/wgtRes/</code>	<code>widget/wgtRe</code>
<code>wgts://</code>	<code>/storage/emulated/0/widgetone/apps/ xxx(widgetAppId)/</code>	<code>/Documents/a</code>
<code>wgts://</code>	<code>/storage/emulated/0/widgetone/widgets/</code>	<code>/Documents/w</code>
<code>file:///sdcard/</code>	<code>/storage/emulated/0/</code>	无

## 2、API概述

### 2.1、Initialization

#### initEasemob(param) 初始化

param为json字符串

```
1. var param{
2.   appKey:,//区别app的标识
3.   apnsCertName://iOS中推送证书名称(仅iOS)
4.   isAutoLoginEnabled://可选参数 是否开启自动登录功能 1-开启 2-关闭
5.   isAutoAcceptGroupInvitation://可选参数 是否开启用户自动同意群邀请,
6. };
```

1. 注:自动登录功能Android SDK 默认开启,iOS SDK默认关闭。

#### login(param) 登录

param为json字符串

```
1. var param = {
2.   username://用户名
3.   password://密码
4. };
```

#### cbLogin(param) 登陆回调

param为json字符串

```
1. var param = {
2.   result://1-成功,2-失败
3.   msg://提示信息
4. };
```

logout() 退出登录

registerUser(param) 注册

param为json字符串

```
1. var param = {  
2.   username:,//用户名  
3.   password:,//密码  
4. };
```

cbRegisterUser(param) 注册回调

param为json字符串

```
1. var param = {  
2.   result:,//1-成功,2-失败  
3.   msg:,//提示信息  
4. };
```

updateCurrentUserNickname(param) 更新当前用户的昵称

param为json字符串

```
1. var param = {  
2.   nickname:  
3. };
```

1. 注:此方法主要为了在苹果推送时能够推送昵称(nickname)而不是userid,一般可以在

getLoginInfo() 获取当前登陆信息(仅iOS可用)

cbGetLoginInfo(param) 获取当前登陆信息的回调(仅iOS)

param为json字符串

```
1. var param={  
2.
```

```
3.     userInfo://当前登陆用户信息
4.     isLoggedIn://当前是否已有登录用户  1-是 2-否
5.     isConnected://是否连上聊天服务器  1-是 2-否
6.     isAutoLoginEnabled://是否自动登录  1-是 2-否
7. };
```

onConnected(); 已连接上

onDisconnected(param) 链接断开

param为json字符串

```
1. var param = {
2.     error://1-账号被移除,2-账号其他设备登陆,3-连接不到聊天服务器,4-当前P
3. };
```

## 2.2、Message

onNewMessage(param) 收到新消息监听

1. 注:param为EMMessage的json格式对象
2. EMMessage具体结构见文末附录
3. 所有离线和在线时接受到的的非透传消息,都通过此回调传递

onCmdMessageReceive(param) 透传消息监听

param为json字符串

```
1. var param = {
2.     msgId:,
3.     message://EMMessage 对象json格式
4.     action:,
5. };
```

onAckMessage(param) 消息已读监听

param为json字符串



```
1. var param = {  
2.     msgId: ,//消息ID  
3.     username: ,//来源  
4. };
```

#### onDeliveryMessage(param) 消息送达监听

param为json字符串

```
1. var param = {  
2.     msgId: ,//消息ID  
3.     username: ,//来源  
4. };
```

#### sendText(param) 发送文本消息及表情

说明：环信移动客服功能，详情见[环信移动客服文档](#)

param为json字符串

```
1. var param = {  
2.     username: ,//单聊时聊天人的userid或者群聊时groupid  
3.     chatType: ,//0-单聊,1-群聊  
4.     content: ,//文本内容  
5.     ext: ,//扩展属性(可选参数,String)  
6.     extObj: ,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr  
7. };
```

#### sendFile(param) 发送文件

说明：环信移动客服功能，详情见[环信移动客服文档](#)

param为json字符串

```
1. var param = {  
2.     username: ,//单聊时聊天人的userid或者群聊时groupid  
3.     chatType: ,//0-单聊,1-群聊  
4.     filePath: ,//文件路径  
5.     displayName: ,//对方接收时显示的文件名(仅iOS需要)  
6.     ext: ,//扩展属性(可选参数,String)  
7.     extObj: ,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr  
8. };
```

### sendPicture(param)//发送图片

说明：环信移动客服功能，详情见[环信移动客服文档](#)

param为json字符串

```
1. var param = {
2.     username:,//单聊时聊天人的userid或者群聊时groupid
3.     chatType:,//0-单聊,1-群聊
4.     filePath:,//图片文件路径
5.     displayName:,//对方接收时显示的文件名(仅iOS需要)
6.     ext:,//扩展属性(可选参数,String)
7.     extObj:,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr
8. };
```

### sendLocationMsg(param)//发送地理位置信息

说明：环信移动客服功能，详情见[环信移动客服文档](#)

param为json字符串

```
1. var param = {
2.     username:,//单聊时聊天人的userid或者群聊时groupid
3.     chatType:,//0-单聊,1-群聊
4.     locationAddress:,//图片文件路径
5.     latitude:,
6.     longitude:,
7.     ext:,//扩展属性(可选参数,String)
8.     extObj:,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr
9. };
```

### sendVoice(param)//发送语音

说明：环信移动客服功能，详情见[环信移动客服文档](#)

param为json字符串

```
1. var param = {
2.
3.     username:,//单聊时聊天人的userid或者群聊时groupid
4.     chatType:,//0-单聊,1-群聊
5.     filePath:,//语音文件路径
6.     length:,//长度(Android必选,iOS可选)
7.     displayName://对方接收时显示的文件名(仅iOS需要)
```

```
8.     ext:,//扩展属性(可选参数,String)
9.     extObj:,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr
10.  };
```

### sendVideo(param) 发送视频

说明: 环信移动客服功能, 详情见[环信移动客服文档](#)

param为json字符串

```
1.  var param = {
2.
3.     username:,//单聊时聊天人的userid或者群聊时groupid
4.     chatType:,//0-单聊,1-群聊
5.     filePath:,//视频文件路径
6.     length:,//长度(Android必选,iOS可选)
7.     displayName://对方接收时显示的文件名(仅iOS需要)
8.     ext:,//扩展属性(可选参数,String)
9.     extObj:,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr
10.  };
```

### sendCmdMessage(param) 发送透传消息

说明: 环信移动客服功能, 详情见[环信移动客服文档](#)

param为json字符串

```
1.  var param = {
2.     chatType:,//0-单聊,1-群聊
3.     action://
4.     toUsername://
5.     ext:,//扩展属性(可选参数,String)
6.     extObj:,//扩展参数 (iOS 3.0.22, Android 3.0.23新增可选参数, JSONStr
7.  };
```

### setNotifyBySoundAndVibrate(param) 消息提醒相关配置

param为json字符串

```
1.  var param = {
2.     enable:,//0-关闭,1-开启。默认为1 开启新消息提醒
3.     soundEnable:,// 0-关闭,1-开启。默认为1 开启声音提醒
4.     vibrateEnable:,// 0-关闭,1-开启。默认为1 开启震动提醒
5.     userSpeaker:,// 0-关闭,1-开启。默认为1 开启扬声器播放(仅Android可用)
```

```
6. showNotificationInBackgroud:// 0-关闭,1-开启。默认为1。设置后台接收
7. acceptInvitationAlways:// 0-关闭,1-开启。默认添加好友时为1,是不需要
8. deliveryNotification:// 0-关闭 1-开启 默认为1 开启消息送达通知
9. };
```

#### getMessageById(param) 根据id获取消息记录

param为json字符串

```
1. var param = {
2.   msgId://消息ID
3. };
```

#### cbGetMessageById(param) 得到一条消息记录

param为json字符串

```
1. var param = {
2.   msg:// EMessage的json格式对象
3. };
```

#### sendHasReadResponseForMessage (param) 发送消息已读回执

param为json字符串

```
1. var param = {
2.
3.   msgId://消息Id
4. };
```

#### onMessageSent (param) 消息已发送监听

param为json字符串

```
1. var param = {
2.
3.   isSuccess://消息是否发生成功 true/false
4.   errorStr://发送失败的原因 当isSuccess为false时才有此项
5.   message://被发送的消息 EMessage结构的json
6. };
```

- 所有send开头的发送消息的API均会触发此监听
- 此监听仅表示消息是否成功发送至环信服务器,不能以此判断接收人是否收到消息

## 2.3、Conversation

**getConversationByName(param)** 根据用户名获取conversation对象

param为json字符串

```
1. var param = {  
2.   username: ,  
3.   chatType: , //聊天类别 0 - 个人 1 - 群组(仅iOS需要,默认0)  
4. }
```

**cbGetConversationByName(param)** 回调

param为json字符串

```
1. var param = {  
2.   conversation: , // EMConversation的json格式对象,格式见附录  
3. };
```

**getMessageHistory(param)** 获取聊天记录

param为json字符串

```
1. var param = {  
2.   username: , //单聊时聊天人的userName或者群聊时groupid  
3.   chatType: , //0-单聊,1-群聊  
4.   startMsgId: , //获取startMsgId之前的pagesize条消息  
5.   pageSize: , //分页大小,为0时获取所有消息,startMsgId可不传  
6. };
```

**cbGetMessageHistory(param)** 获取聊天记录回调

param为json字符串

```
1. var param = {
```

```
2.   messages:,//List<EMessage>的json格式对象
3.   };
```

`getUnreadMsgCount (param)` 获取未读消息数量

param为json字符串

```
1. var param = {
2.   username://username | groupid
3.   chatType://聊天类别 0-单聊 1-群聊(仅iOS需要,默认0)
4. };
```

`cbGetUnreadMsgCount (param)` 获取未读消息数量回调

param为json字符串

```
1. var param = {
2.   count://未读消息数
3. };
```

`resetUnreadMsgCount (param)` 指定会话未读消息数清零

param为json字符串

```
1. var param = {
2.   username://username | groupid
3.   chatType://聊天类别 0-单聊 1-群聊(仅iOS需要,默认0)
4. };
```

`resetAllUnreadMsgCount ()`; 所有未读消息数清零(仅Android可用)

`getMsgCount (param)` 获取消息总数(仅Android可用)

param为json字符串

```
1. var param = {
2.   username://username | groupid
3. };
```

`cbGetMsgCount (param)` 获取消息总数回调(仅Android可用)

param为json字符串

```
1. var param = {  
2.   msgCount:,//消息总数  
3. };
```

`clearConversation(param)` 清空会话聊天记录(仅Android可用)

param为json字符串

```
1. var param = {  
2.   username://username | groupid  
3. };
```

`deleteConversation(param)` 删除和某个user的整个的聊天记录(包括本地)

param为json字符串

```
1. var param = {  
2.   username://username | groupid  
3.   chatType://0-个人 1-群组(默认0,此参数仅iOS需要)  
4. };
```

`removeMessage(param)` 删除当前会话的某条聊天记录

param为json字符串

```
1. var param = {  
2.   username://username | groupid  
3.   msgId:,  
4.   chatType://0-个人 1-群组(默认0,此参数仅iOS需要)  
5. };
```

`deleteAllConversation()`; 删除所有会话记录(包括本地)

`getChatterInfo()`; 获取聊天对象信息

`cbGetChatterInfo(param)`; 获取聊天对象信息回调

1. `param`为`list<EMChatterInfo>`,一个由`EMChatterInfo`结构组成的数组。

- 包含所有好友和群组的聊天对象信息
- `EMChatterInfo`结构说明见附录

`getTotalUnreadMsgCount()`; 获取总计未读消息数

`cbGetTotalUnreadMsgCount(param)`; 获取总计未读消息数回调

`param`为json字符串

```
1. var param = {  
2.     count: , //总计未读消息数  
3. };
```

`getRecentChatters()`; 获取近期聊天对象信息

`cbGetRecentChatters(param)`;

1. `param`为`list<EMChatterInfo>`,一个由`EMChatterInfo`结构组成的数组。

- 仅包含有聊天记录的聊天对象(包括非好友)
- `param`已按`lastMsg`的时间戳排序,最后接收消息的时间越新,排序越靠前
- `EMChatterInfo`结构说明见附录

## 2.4、Friend

`onContactAdded(param)`//新增联系人监听

说明

(iOS 3.0.22后支持, android已支持)

`param`为json字符串

```
1. var param = {  
2.     userNameList: , //json格式的List<String>  
3. };
```



### onContactDeleted(param)//删除联系人监听

#### 说明

(iOS 3.0.22后支持, android已支持)

param为json字符串

```
1. var param = {  
2.     userNameList://json格式的List<String>  
3. };
```

### onContactInvited(param)//接到好友申请

param为json字符串

```
1. var param = {  
2.     username://  
3.     reason://  
4. };
```

### onContactAgreed(param)//好友请求被同意

param为json字符串

```
1. var param = {  
2.     username://  
3. };
```

### onContactRefused(param)//好友请求被拒绝

param为json字符串

```
1. var param = {  
2.     username://  
3. };
```

### getContactUserNames();//获取好友列表

### cbGetContactUserNames(param)//获取好友列表回调

param为json字符串

```
1. var param = {  
2.   usernames: ,//用户姓名字符串构成的数组  
3.  
4. };
```

**addContact (param)**//添加好友

param为json字符串

```
1. var param = {  
2.   toAddUsername: ,//要添加的好友  
3.   reason:  
4. };
```

**deleteContact (param)**//删除好友

param为json字符串

```
1. var param = {  
2.   username: ,//  
3. };
```

**acceptInvitation (param)**//同意username的好友请求

param为json字符串

```
1. var param = {  
2.   username: ,//  
3. };
```

**refuseInvitation (param)**//拒绝username的好友请求

param为json字符串

```
1. var param = {  
2.   username: ,//  
3.   reason: ,//拒绝好友请求原因(仅iOS需要)  
4. };
```

`getBlackListUsernames()` //获取黑名单列表

`cbGetBlackListUsernames(param)` //获取黑名单列表回调

param为json字符串

```
1. var param = {  
2.   usernames:,//List<String> json格式  
3. };
```

`addUserToBlackList(param)` //把用户加入到黑名单

param为json字符串

```
1. var param = {  
2.   username://  
3. };
```

`deleteUserFromBlackList(param)` //把用户从黑名单中移除

param为json字符串

```
1. var param = {  
2.   username://  
3. };
```

## 2.5、Group

`onInvitationDeclined(param)` //群聊邀请被拒绝

param为json字符串

```
1. var param = {  
2.   groupId:,  
3.   invitee:,  
4.   reason:,  
5. };
```

### onInvitationAccepted(param)//群聊邀请被同意

param为json字符串

```
1. var param = {  
2.   groupId:,  
3.   inviter:,  
4.   reason:,  
5. };
```

### onUserRemoved(param)//当前用户被管理员移除出群聊

param为json字符串

```
1. var param = {  
2.   groupId:,  
3.   groupName:,  
4. };
```

### onGroupDestroy(param)//群聊被创建者解散

param为json字符串

```
1. var param = {  
2.   groupId:,  
3.   groupName:,  
4. };
```

### onApplicationReceived(param)//用户申请入群聊,收到入群申请

param为json字符串

```
1. var param = {  
2.   groupId:,  
3.   groupName:,  
4.   applyer:,  
5.   reason:,  
6. };
```

### onApplicationAccept(param)// // 入群申请被同意

param为json字符串

```
1. var param = {
2.     groupId:,
3.     groupName:,
4.     accepter:,
5. };
```

`onApplicationDeclined(param)` //加群申请被拒绝

param为json字符串

```
1. var param = {
2.     groupId:,//(仅Android)
3.     groupName:,
4.     decliner:,
5.     reason:,
6. };
```

`createPrivateGroup(param)` //创建私有群

param为json字符串

```
1. var param = {
2.     groupName:,//要创建的群聊的名称
3.     desc://群聊简介
4.     members://群聊成员,为空时这个创建的群组只包含自己
5.     allowInvite://是否允许群成员邀请人进群
6.     maxUsers://最大群聊用户数,可选参数,默认为200,最大为2000
7.     initialWelcomeMessage://群组创建时发送给每个初始成员的欢迎信息(仅iOS)
8. };
```

`createPublicGroup(param)` //创建公开群

param为json字符串

```
1. var param = {
2.     groupName:,//要创建的群聊的名称
3.     desc://群聊简介
4.     members://群聊成员,为空时这个创建的群组只包含自己
5.     needApprovalRequired://如果创建的公开群用需要户自由加入,就传false。否
6.     maxUsers://最大群聊用户数,可选参数,默认为200,最大为2000
```

```
7.     initialWelcomeMessage://群组创建时发送给每个初始成员的欢迎信息(仅i  
8. };
```

**onGroupCreated(param)**//群组建立监听

param为json字符串

```
1. var param = {  
2.     isSuccess://群组是否创建成功 true/false  
3.     errorStr://创建失败的原因 仅isSuccess传false时会有此参数  
4.     group://EMGroup的json字符串,被建立的群组信息  
5. };
```

**addUsersToGroup(param)**//群聊加入

param为json字符串

```
1. var param = {  
2.     isGroupOwner://是否群主(仅Android需要)  
3.     groupId://  
4.     newmembers://群聊新成员,List<String> Json格式  
5.     inviteMessage:// 新增参数 邀请信息  
6. };
```

**removeUserFromGroup(param)**//群聊减人

param为json字符串

```
1. var param = {  
2.     groupId://  
3.     username://  
4. };
```

1. 只有owner才有权限进行此操作

**joinGroup(param)**//加入某个群聊,只能用于加入公开群

param为json字符串

```
1. var param = {
```

```
2.     groupId://
3.     reason:// //如果群开群是自由加入的,即group.isMembersOnly()为false,
4.     groupName://群组名称
5. };
```

`exitFromGroup(param)` //退出群聊

param为json字符串

```
1. var param = {
2.     groupId://
3. };
```

`exitAndDeleteGroup(param)` //解散群聊

param为json字符串

```
1. var param = {
2.     groupId://
3. };
```

`getGroupsFromServer(param)` //从服务器获取自己加入的和创建的群聊列表

param为json字符串

```
1. var param = {
2.     loadCache://是否从本地加载缓存,(默认为false,从网络获取)
3. };
```

`cbGetGroupsFromServer(param)` //从服务器获取自己加入的和创建的群聊列表回调

param为json字符串

```
1. var param = {
2.     result://0-成功,1-失败
3.     grouplist://List<EMGroup> json格式
4.     errorMsg:
5. };
```

`getAllPublicGroupsFromServer(param)` ;//获取所有公开群列表

param为json字符串

```
1. var param = {
2.   pageSize://期望结果的数量, 如果 < 0 则一次返回所有结果
3.   cursor://获取公开群的cursor,首次调用传空即可
4. };
```

cbGetAllPublicGroupsFromServer (param)//获取所有公开群列表回调

param为json字符串

```
1. var param = {
2.   result://0-成功,1-失败
3.   grouplist:List< EMGroup> json格式 见附录
4.   errorMsg:
5.   cursor://
6. };
```

getGroup (param)//获取群详情

param为json字符串

```
1. var param = {
2.   groupId://
3.   loadCache://是否从本地加载缓存,(默认为false,从网络获取)
4. };
```

cbGetGroup (param)//获取群详情回调

param为json字符串

```
1. var param = {
2.   //EMGroup 对象json格式
3. };
```

blockGroupMessage (param)//屏蔽群消息

param为json字符串

```
1. var param = {
```



```
2.     groupId://
3. };
```

**unlockGroupMessage(param)**//解除屏蔽群

param为json字符串

```
1. var param = {
2.     groupId://
3. };
```

**changeGroupName(param)**//修改群组名称

param为json字符串

```
1. var param = {
2.     groupId://
3.     changedGroupName://改变后的群组名称
4. };
```

**setReceiveNotNoifyGroup(param)**//群聊不提醒只显示数目(仅Android可用)

param为json字符串

```
1. var param = {
2.     groupIds:// List<String>
3. };
```

**blockUser(param)**//将群成员拉入群组的黑名单

说明

(iOS 3.0.22后支持, android已支持)

param为json字符串

```
1. var param = {
2.     groupId://
3.     username://待屏蔽的用户名
4. };
```

**unlockUser(param)**//将拉入黑名单的群成员移除

## 说明

(iOS 3.0.22后支持, android已支持)

param为json字符串

```
1. var param = {  
2.   groupId:,//  
3.   username://待解除屏蔽的 用户名  
4. };
```

`getBlockedUsers(param)`//获取群组的黑名单用户列表

## 说明

(iOS 3.0.22后支持, android已支持)

param为json字符串

```
1. var param = {  
2.   groupId:,//  
3. };
```

`cbGetBlockedUsers(param)`//获取群组的黑名单用户列表回调

## 说明

(iOS 3.0.22后支持, android已支持)

param为json字符串

```
1.  
2.   var param = {  
3.     usernames:// List<String> json格式  
4.   };
```

`onGroupUpdateInfo(param)`//群组信息更新的监听(仅iOS)

param为json字符串

```
1. var param={  
2.   group://EMGroup对象的json格式字符串  
3. };
```

- 每当添加/移除/更改角色/更改主题/更改群组信息之后, 都会触发此回调

### `onDidJoinedGroup(param)` //自动加入群组监听

#### 说明

(iOS 3.0.22新增接口)

SDK自动同意了用户A的加B入群邀请后,用户B接收到该回调,需要initEasemob时设置isAutoAcceptGroupInvitation为TRUE

param为json字符串

```
var param = {  
  groupId:,  
  groupName:,  
  username:, //邀请者  
  message:, //邀请消息  
};
```

### `onReceiveGroupInvitation(param)` //收到群组邀请监听

#### 说明

(iOS 3.0.22, Android 3.0.23新增接口)

用户A邀请用户B入群,用户B接收到该回调

param为json字符串

```
var param = {  
  groupId:,  
  username:, //邀请者  
  message:, //邀请消息  
};
```

### `acceptJoinApplication(param)` //批准入群申请, 需要Owner权限

#### 说明

(iOS 3.0.22, Android 3.0.23新增接口)

param为json字符串

```
var param={  
  groupId:, //所申请的群组ID
```

```
username://申请人  
};
```

#### **declineJoinApplication(param)//拒绝入群申请, 需要Owner权限**

##### 说明

(iOS 3.0.22, Android 3.0.23新增接口)

param为json字符串

```
var param={  
groupId://所申请的群组ID  
username://申请人  
reason://拒绝理由  
};
```

#### **acceptInvitationFromGroup(param)//接受入群邀请**

##### 说明

(iOS 3.0.22, Android 3.0.23新增接口)

param为json字符串

```
var param={  
groupId://所申请的群组ID  
username://邀请者  
};
```

#### **declineInvitationFromGroup(param)//拒绝入群邀请**

##### 说明

(iOS 3.0.22, Android 3.0.23新增接口)

param为json字符串

```
var param={  
groupId://所申请的群组ID  
username://邀请者  
reason://拒绝理由  
};
```

## 2.6、Call

`onCallReceive(param)` // 实时语音监听

param为json字符串

```
1. var param = {  
2.     
3.   from; // 拨打方username  
4.   callType; // 0-语音电话 1-视频电话  
5.   callId; // 本次通话的EMSessionId  
6. };
```

`onCallStateChanged(param)` // 通话状态监听

param为json字符串

```
1. var param = {  
2.   state; // 1-正在连接对方, 2-双方已经建立连接, 3-同意语音申请, 建立语音通话  
3. };
```

1. eg. 一个成功的语音通话流程为 :A发送通话请求给B ==> AB建立语音通话连接 ==>

`makeVoiceCall(param)` // 拨打语音通话

param为json字符串

```
1. var param = {  
2.   username; //  
3. };
```

`answerCall()` // 接听通话

`rejectCall()` // 拒绝接听

`endCall()` // 挂断通话

## 2.7、Apns(以下方法全部仅限iOS)

```
registerRemoteNotification();//注册Apns推送
```

```
cbRegisterRemoteNotification(param);//回调
```

param为json字符串

```
1. var param{
2.
3.   result;//1-成功 2-失败
4.   errorInfo;//注册失败时的错误信息
5. };
```

```
onApnsLaunch(param);
```

1. 若APP是通过点击apns推送调起的,当插件初始化时会触发此回调。
2. param为此条推送的内容,json格式。

```
updatePushOptions(param);//设置apns全局属性
```

param为json字符串

```
1. var param{
2.   nickname;//昵称
3.   displayStyle;//推送显示类型 0-提示"您有一条新消息" 1- 显示详细消息内容
4.   noDisturbingStyle;//是否开启免打扰模式 0-全天免打扰 1-自定义时段免打
5.   noDisturbingStartH;//免打扰模式开始时间 小时(int)
6.   noDisturbingEndH;//免打扰模式结束时间 小时(int)
7. };
```

```
cbUpdatePushOptions(param);//设置apns全局属性回调
```

param为json字符串

```
1. var param{
2.   nickname;//昵称
3.   displayStyle;//推送显示类型 0-提示"您有一条新消息" 1- 显示详细消息内容
```

```
4. noDisturbingStyle; //是否开启免打扰模式 0-全天免打扰 1-自定义时段免打  
5. noDisturbingStartH; //免打扰模式开始时间 小时(int)  
6. noDisturbingEndH; //免打扰模式结束时间 小时(int)  
7. };
```

1. 说明: updatePushOptions全为可选参数, 当传入空值时, 即可通过回调获得当前apns

ignoreGroupPushNotification(param): //设置指定群组是否接收

param为json字符串

```
1. var param{  
2.   groupId; //指定的群组Id  
3.   isIgnore; //1-屏蔽 2-取消屏蔽  
4. };
```

cbIgnoreGroupPushNotification(param): //回调

param为json字符串

```
1. var param{  
2.   groupIds; //已屏蔽接收推送消息的群列表  
3. };
```

### 3、附录

EMMessage json字符串返回值结构

key	说明
from	发送者
to	接受者
messageId	消息id
messageTime	消息发送或接收的时间
isAcked	是否接收到了接收方的阅读回执, 或是否已发送了阅读回执给对方

isDelivered	对于发送方来说, 该值表示:接收方是否已收到了消息, 对于接收方来说, 表示:接收方是否已发送了”已接收回执”给对方
isRead	是否已读
chatType:	聊天类别 0-个人 1-群组
messageType	消息类型 text/video/audio/image/location/file/cmd
ext	扩展属性 String格式
extObj	扩展属性 (iOS 3.0.22, Android 3.0.23新增, JSONObject用于环信移动客服功能, 详情见 <a href="#">环信移动客服文档</a> )
messageBody	消息主体 json

messageBody的结构为

#### 普通文本消息

key	说明
text	文本内容

#### 透传消息

key	说明
action	具体命令

#### 位置消息

key	说明
longitude	经度
latitude	纬度
address	地理位置信息

#### 视频/语音/图片/文件消息

key	说明
displayName	显示名
remotePath	服务器远程文件路径
secretKey	远端文件的密钥



length	长度 (单位:秒 仅语音/视频消息)
thumbnailRemotePath	预览图文件的服务器远程路径(仅视频/图片消息)
thumbnailSecretKey	预览图文件的密钥(仅视频/图片消息)

- 返回的json数据中会包含除上述属性之外的一些其他信息, 均可以忽略
- 由于 **Android SDK** 不能获取 **已发送消息** 的 **remotePath** 和 **thumbnailRemotePath** , 因此改用 **本地文件路径** (file://开头)代替

#### EMConversation json字符串返回值结构

key	说明
chatter	conversation识别名
chatType	聊天类别 0-个人 1-群组
messages	“conversation所包含的message列表, 表内元素为EMMessage的json字符串”

- 返回的json数据中会包含除上述属性之外的一些其他信息, 均可以忽略

#### EMGroup json字符串返回值结构

key	说明
groupName	群组名称
groupDescription	群组描述
members	包含的成员
owner	群主
isPushNotificationEnable	是否允许推送提醒
isBlock	是否被用户屏蔽
groupMaxUserCount	群组最大人数
groupId	群组Id
isPublic	群组类型
allowInvites	是否允许群成员邀请人进群
membersOnly	需要申请和验证才能加入

- 返回的json数据中会包含除上述属性之外的一些其他信息, 均可以忽略
- 受SDK所限, 部分接口的members属性会有误
  - 群组的members属性以cbGetGroup回调为准
  - **Android的members包含群组的owner, 而iOS并不包含** , 此问题已反馈给环信官

方,待解决中

#### EMChatterInfo json字符串返回值结构

key	说明
chatter	联系人的username或群组的groupId
groupName	群组名(仅群组有此值)
chatType	联系人类型 0-个人 1-群组
unreadMsgCount	未读消息数
lastMsg	最后一条消息 EMMessage格式的json字符串

#### “isGroup”参数废弃 改用“chatType”的相关说明

由于环信插件即将添加 **聊天室功能**, isGroup参数即将不能满足需求,因此做如下修改:

- 所有的调用API中,入参里的isGroup改为chatType
- 所有的回调API中, isGroup属性改为chatType

## 4、更新历史

iOS

API版本: **uexEasemob-3.0.22**

最近更新时间: **2016-5-7**

历史发布版本	更新内容
3.0.22	initEasemob接口新增isAutoAcceptGroupInvitation参数;消息接口新增extObj参数以适配环信客服功能; group内新增群组加人的相关接口
3.0.21	更新SDK到3.1.1
3.0.20	修复加入公开群会失败的问题
3.0.19	修复onMessageSent中群组信息错误的问题;setNotifyBySoundAndVibrate支持后台通知
3.0.18	修复透传消息也会提示用户的问题

3.0.17	修复getMessageHistory接口传入startMsgId但pagesize为0时取不到所有messages的问题
3.0.16	修复cbGetGroup回调结构和Android不一致的问题
3.0.15	SDK版本更新至V2.2.0r1 修复iOS和Android之间互发消息时ext属性丢失的bug
3.0.14	cbLogin 返回信息key修改为” msg”
3.0.13	修复发送位置经度为0的bug
3.0.12	新增API:onGroupCreated
3.0.11	新增API:getRecentChatters, onMessageSent
3.0.10	SDK更新; EMGroup回调结构优化; 支持IDE打包
3.0.9	SDK更新, 代码优化等, 详见文档
3.0.8	新增方法 发送视频消息等, 详见文档
3.0.7	优化了本地数据库的相关设置
3.0.6	自动登录功能改在 初始化 中进行设置
3.0.5	针对新版本做了一些代码优化
3.0.4	支持自动登录, 更新环信iOS SDK版本至V2.1.6
3.0.3	新增了一个获取聊天对象信息的方法
3.0.2	修复了回调信息不全引起程序崩溃的bug
3.0.1	修正了多次初始化会导致程序卡死的bug
3.0.0	环信(Easemob)插件 for iOS

## Android

API版本: `uexEasemob-3.0.23`

最近更新时间: `2016-5-7`

历史发布版本	更新内容
3.0.23	升级环信官方SDK到3.1.2, 同时添加客服功能
3.0.22	init只执行一次
3.0.21	修改getRecentChatters
3.0.20	修复setNotifyBySoundAndVibrate无效的问题
3.0.19	纠正拼写错误

3.0.18	修改cbGetMessageHistory与文档不一致的问题
3.0.17	支持initEasemob配置key
3.0.16	修复退出再登陆,聊天记录只有一条的问题
3.0.15	去掉特殊字符
3.0.14	修复CMD消息.json不能解析的问题
3.0.13	修复重新登陆后消息记录remoteUrl为空的问题
3.0.12	发送附件支持wgt, wgts协议
3.0.11	添加创建群回调onGroupCreated
3.0.10	添加getRecentChatters接口
3.0.9	支持离线消息
3.0.8	getGroup添加groupName, groupDescription 字段
3.0.7	getChatterInfo增加返回新加好友的聊天记录
3.0.6	添加getTotalUnreadMsgCount接口
3.0.5	升级环信SDK
3.0.4	升级环信SDK
3.0.3	发送语音添加length参数
3.0.2	添加sendHasReadResponseForMessage 等接口
3.0.1	添加getChatterInfo接口
3.0.0	环信插件

## 5 文档更新记录

2016-5-7

- initEasemob接口新增isAutoAcceptGroupInvitation参数;
- 消息接口新增extObj参数以适配环信客服功能;
- group内新增群组加人的相关接口

2015-12-3

- Android支持接口配置appKey, 无需自定义插件

2015-8-18

- 新增API: onGroupCreated 群组已建立的回调
- 修复若干bugs

2015-8-3

- 新增了如下API:
  - onMessageSent 消息已发送的监听
  - getRecentChatters 获取最近联系人
  - cbGetRecentChatters 获取最近联系人回调

2015-7-7

- iOS SDK更新至2.1.8;
- EMGroup中 groupSubject属性废弃,改用groupName和GroupDescription来获取相应信息;

2015-6-18

- iOS插件版本更新至3.0.9,用以支持新版SDK (iOS V2.1.7)
- Android插件版本更新至3.0.6 用以支持新版SDK(AndroidV2.1.9)
- 方法getAllPublicGroupsFromServer添加的参数,变得更加实用了(详情见接口说明)
- 新增方法getTotalUnreadMsgCount() 获取总计未读消息数 及其回调
- 开放Apns离线推送相关接口(仅限iOS)
- 所有参数中的" isGroup" 即将废弃,改用" chatType" (详见附录)

2015-5-25

- iOS插件版本更新至3.0.8
- 新增方法sendVedio 发送视频消息;
- EMMessage回调中,添加length 长度(单位:秒,仅语音、视频消息);
- EMMessage回调中,添加ext 扩展属性; 发送消息的各个API也添加此项作为可选参数;(扩展属性为一个自定义的字符串,用以携带开发者可能需要的其他参数)
- 新增方法sendHasReadResponseForMessage 发送已读回执;(对方会触发回调 onAckMessage, 插件不再自动发送此回执)
- 方法getMessageById根据消息id获取消息记录及其回调也支持iOS了

2015-05-06

- 删去方法 设置是否自动登录 ,改为在 init初始化 中添加相关参数

2015-05-05

- 更新环信iOS SDK版本至V2.1.6(2015-04-30版),部分代码做了优化以支持此新版本;
- 现在所有的回调函数都会返回给进行 init初始化 操作的那个网页了。
- cbGetLoginInfo现在也会返回 是否开启自动登录 的信息了。

2015-05-04

- 新增方法 设置是否自动登录;
- 现在iOS也支持回调 [1.10]onConnected 了。

2015-04-28

- 新增方法getChatterInfo获取聊天对象信息及其回调;

2015-04-20

- 新增 getMessageById 方法及回调;
- 修改了onNewMessage的回调值的结构;
- 统一了Android和iOS返回的json对象的结构, 对附录做了大量修订。

2015-04-17

- 初稿