

# 腾讯云 IM 开放-WebSDK-群组管理 REST 协议文档

文档编号		
文档名称	腾讯云 IM 开放-WebSDK-群组管理 REST 协议文档	
作者	peakerdong	
首次完成时间	2015-11-3	
当前版本	V1.0	
修改记录		
时间	修改人	修改说明

# 目 录

1 综述 .....	- 3 -
1.1 群组系统简介 .....	- 3 -
1.2 群组数据结构 .....	- 3 -
1.3 群组形态详述 .....	- 4 -
2 群消息收发 .....	- 5 -
2.1 发送群消息 .....	- 5 -
2.2 拉取群漫游消息 .....	- 5 -
3 群组管理 .....	- 7 -
3.1 创建群 .....	- 7 -
3.2 搜索群 .....	- 7 -
3.3 申请加群 .....	- 8 -
3.4 主动退群 .....	- 9 -
3.5 解散群 .....	- 9 -
3.6 获取我的群组列表（高级接口） .....	- 9 -
3.7 获取群组详细资料（高级接口） .....	- 11 -
3.8 修改群基本资料 .....	- 13 -
4 群成员管理 .....	- 14 -
4.1 获取群成员列表 .....	- 14 -
4.2 邀请好友加群 .....	- 15 -
4.3 修改群消息提示 .....	- 16 -
4.4 修改群成员角色 .....	- 16 -
4.5 设置群成员禁言时间 .....	- 17 -
4.6 删除群成员 .....	- 17 -
5 错误码列表 .....	- 18 -

# 1 综述

## 1.1 群组系统简介

所谓群组系统，本质上就是一个支持多人聊天的即时通讯系统。群组系统当前提供三种群组形态：

私有群（Private）：适用于较为私密的聊天场景，群组资料均不公开，类似于微信群。

公开群（Public）：适用于公开群组，具有较为严格的管理机制、准入机制，类似于 QQ 群。

聊天室（ChatRoom）：群成员可以随意进出，组织较为松散。

各群组形态之间的具体差异参见[群组形态详述](#)。

## 1.2 群组数据结构

群基础资料（对该群组的描述）：

字段名称	类型	描述	备注
GroupId	String	群组的唯一标识	只读
Type	String	群组形态（注 1）	只读
Name	String	群组名称（最长 30 字节）	（注 2）
Introduction	String	群组简介（最长 120 字节）	
Notification	String	群组公告（最长 150 字节）	
FaceUrl	String	群组头像 URL（最长 100 字节）	
Owner_Account	String	群主 ID	只读
CreateTime	Integer	群组的创建时间	只读
InfoSeq	Integer	群资料的每次变都会增加该值	只读
LastInfoTime	Integer	群组最后一次信息变更时间	只读
LastMsgTime	Integer	群组内最后发消息的时间	只读
NextMsgSeq	Integer	群内下一条消息的 Seq（注 3）。	只读
MemberNum	Integer	当前成员数量	只读
MaxMemberNum	Integer	最大成员数量	
ApplyJoinOption	String	申请加群选项（注 4）	

群成员资料（对群内每个成员的描述）：

字段名称	类型	描述	备注
Member_Account	String	群成员 ID	只读
Role	String	群内身份（注 5）	
JoinTime	Integer	入群时间	只读
MsgSeq	Integer	该成员当前已读消息 Seq	只读
MsgFlag	String	消息屏蔽类型（注 6）	
LastSendMsgTime	Integer	最后发送消息的时间	只读

注 1：当前支持三种产品形态：Private 表示私有群，Public 表示公开群，ChatRoom 表示聊天室。

注 2：对于公开群、聊天室，这几个数据只有群主或者 APP 管理员才能修改；但是对于私有群，这几个数据任意群成员均可修改，具体可以参见[群组形态详述](#)中的相关描述。

注 3：群内每产生一条消息，NextMsgSeq 自增 1。

注 4：申请加群选项包括如下几种：DisableApply 表示禁止任何人申请加入；NeedPermission 需要群主或管理员审批；FreeAccess 允许自由进出。

注 5：群内身份包括如下几种：Owner 表示群主，Admin 表示群管理员，Member 表示群成员。

注 6：消息屏蔽类型包括如下几种：AcceptAndNotify 表示接收并提示；AcceptNotNotify 表示接收不提示；Discard 表示屏蔽群消息（不会下发）。

## 1.3 群组形态详述

三种群组形态之间的主要差异如下：

	控制字段	备注	讨论组	公开群	聊天室
基础资料	是否允许普通成员修改资料	0: 普通成员（非群主、管理员）不能修改群基础资料； 1: 普通成员可以修改群名称、简介、公告、头像，但不能修改其他字段。	1	0	0
	是否公开	0: 非群成员无法获取该群的公开资料； 1: 非群成员可以通过群 ID 获取该群的公开资料，例如群名称等。	0	1	1
	是否允许需申请加群	0: 任何申请加群的请求都不被允许； 1: 依照申请加群选项（ApplyJoinOption）来决定后续操作。	0	1	1
	是否允许直接拉人入群	0: 群内任意成员可以不经其他用户同意便将其拉入群中； 1: 任何邀请他人入群的操作都是不允许的（注 1）。	1	0	0
	是否允许群主退群	0: 群主不得退群，如果要退群，只能通过解散群的方式； 1: 群主可以退群，退群之后，群组将没有群主。	1	0	0
	是否允许群主解散群	0: 群主不能解散群（但是 APP 管理员可以强制将其解散）； 1: 群主可以解散群。	0	1	1
	是否允许设置管理员	0: 群主不能设置管理员； 1: 群主可以设置管理员。	0	1	1
	是否允许普通成员踢人	0: 普通群成员不具有踢人权限，仅群主、管理员才可以； 1: 群内任意成员可以将任意其他成员踢出该群。	0	0	0
消息	无限制历史消息	0: 群成员只能看到自己入群之后的消息；	0	0	1

		1: 群成员可以看到自己入群之前的历史消息。			
	成员变更通知	0: 群成员变更不会在群内发送通知; 1: 在群成员变更之后推送一条消息, 将这一信息通知所有群成员。	1	1	0
	是否在首条消息之后激活群	0: 新创建的群组立即针对所有群成员可见; 1: 在创建群之后, 不论当前群内有多少成员, 只要创建者没有说话, 其他群成员就不会感知到自己被拉入该群, 只有创建者自己能感知到该群的存在。	1	0	0
自动回收	在不活跃一周之后自动回收	0: 群组永远不会被后台回收, 除非群主解散, 或者所有成员都退出群组; 1: 如果群在给定时间内(一周/一个月/三个月), 既没有人说话, 也没有成员变更, 则静默回收该群。目的是节省 APP 在所占用的资源。	0	0	0
	在不活跃一月之后自动回收				
	在不活跃三月之后自动回收				

## 2 群消息收发

### 2.1 发送群消息

API 名称	sendMsg
协议类型	公开协议
接口说明	Msg-webim.Msg 对象, cbOk-发消息成功回调函数, cbErr-发消息失败回调函数
返回结果示例	{ "ActionStatus": "OK", "ErrorInfo": "", "ErrorCode": 0 }

### 2.2 拉取群漫游消息

API 名称	syncGroupMsgs
协议类型	公开协议
接口说明	
请求参数示例	{ "GroupId": "@TGS#2C5SZEAEF", }

	<pre> "ReqMsgSeq": "100", // 起始消息 seq, 接口返回从请求的 seq 向后取 ReqMsgNumber 条信息 "ReqMsgNumber": "100" // 请求数  } </pre>
返回结果示例	<pre> {   "ActionStatus": "OK",   "ErrorCode": 0,   "GroupId": "@TGS#343GQPAEU",   "IsFinished": 1,   "RspMsgList": [     {       "From_Account": "seven",       "IsPlaceMsg": 0,       "MsgBody": [         {           "MsgContent": {             "Second": 3,             "Size": 59532,             "UUID": "306c0201000465306302010004123134343"           },           "MsgType": "TIMSoundElem"         }       ],       "MsgSeq": 9,       "MsgTimeStamp": 1446542780     },     {       "From_Account": "seven",       "IsPlaceMsg": 0,       "MsgBody": [         {           "MsgContent": {             "Text": "好咯膜"           },           "MsgType": "TIMTextElem"         }       ],       "MsgSeq": 1,       "MsgTimeStamp": 1446515316     }   ] } </pre>

## 3 群组管理

### 3.1 创建群

API 名称	createGroup
接口说明	1. 支持创建私有群、公开群、聊天室三种群组。
请求参数示例	<pre>{   "Owner_Account": "leckie", // 群主的 UserId (选填)   "Type": "Public", // 群组类型: Private/Public/ChatRoom (必填)   "Name": "TestGroup", // 群名称 (必填)   "Introduction": "This is group Introduction", // 群简介 (选填)   "Notification": "This is group Notification", // 群公告 (选填)   "FaceUrl": "http://this.is.face.url", // 群头像 URL (选填)   "MaxMemberCount": 500, // 最大群成员数量 (选填)   "ApplyJoinOption": "FreeAccess", // 申请加群处理方式 (选填)   "AppDefinedData": [ // 群组维度的自定义字段 (选填)     {       "Key": "GroupTestData1", // APP 自定义的字段 Key       "Value": "xxxxxx" // 自定义字段的值     },     {       "Key": "GroupTestData2",       "Value": "abc\u0000\u0001" // 自定义字段支持二进制数据     }   ],   "MemberList": [ // 初始群成员列表, 最多 500 个 (选填)     Bob, peter   ] }</pre>
返回结果示例	<pre>{   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0,   "GroupId": "@TGS#2J4SZEDEL" // 创建成功之后的群 ID, 由 IM 云后台分配 }</pre>

### 3.2 搜索群

API 名称	getGroupPublicInfo
接口说明	1. 接口支持仅获取群组中的指定字段, 通过过滤器 GroupBasePublicInfoFilter 来控制;
请求参数示例	<pre>{   "GroupIdList": [ // 群组列表, 必填</pre>

	<pre>"@TGS#2J4SZEAEEL" ], "GroupBasePublicInfoFilter": [ // 公开信息字段，请在此数组中添加     "Type",     "Name",     "Introduction",     "Notification",     "FaceUrl",     "CreateTime",     "Owner_Account",     "LastInfoTime",     "LastMsgTime",     "NextMsgSeq",     "MemberNum",     "MaxMemberNum",     "ApplyJoinOption" ] }</pre>
返回结果示例	<pre>{   "ActionStatus": "OK",   "ErrorCode": 0,   "GroupInfo": [     {       "ApplyJoinOption": "FreeAccess",       "CreateTime": 1446099817,       "ErrorCode": 0,       "FaceUrl": "",       "GroupId": "@TGS#3VIGQPAEO",       "Introduction": "聊天室 1",       "MaxMemberNum": 10000,       "MemberNum": 3,       "Name": "聊天室 1",       "Owner_Account": "peaker1",       "Type": "ChatRoom"     }   ] }</pre>

### 3.3 申请加群

API 名称	applyJoinGroup
接口说明	
请求参数示例	<pre>{   "GroupId": "@TGS#2J4SZEAEEL",   "ApplyMsg": "想加入贵群", }</pre>



	"UserDefinedField": "" }
返回结果示例	{ "ActionStatus": "OK", "ErrorCode": 0, "JoinedStatus": "JoinedSuccess" }

### 3.4 主动退群

API 名称	quitGroup
接口说明	
请求参数示例	{ "GroupId": "@TGS#2J4SZEAE" }
返回结果示例	{ "ActionStatus": "OK", "ErrorInfo": "", "ErrorCode": 0 }

### 3.5 解散群

API 名称	destroyGroup
接口说明	群组解散之后将无法恢复，请谨慎调用该接口。
请求参数示例	{ "GroupId": "@TGS#2J4SZEAE" }
返回结果示例	{ "ActionStatus": "OK", "ErrorInfo": "", "ErrorCode": 0 }

### 3.6 获取我的群组列表（高级接口）

API 名称	getJoinedGroupListHigh
接口说明	1. 该接口支持分页拉取（通过请求包中的 Limit 和 Offset 控制）， 2. 如果请求包中指定了 GroupType 这个过滤条件，则表示仅获取特定群组形态的群组；

	<p>3. 返回包中的 TotalCount，表示的是符合条件的所有群组，与是否启用分页无关。例如，即使指定 Limit 为 10，GroupType 为 Public，但如果用户加入的公开群组为 100 个，则回包中 TotalCount 的值为 100，但 GroupIdList 数组中只会有 10 个群组的信息。</p>
请求参数示例	<pre>{   "Member_Account": "leckie",   "Limit": 10, // 拉取多少个，不填标识拉取全部   "Offset": 0, // 从第多少个开始拉取   "GroupType": "Public", // 拉取哪种群组形态，不填为拉取所有   "GroupBaseInfoFilter": [ // 需要哪些基础信息字段，含义参见“基本字段介绍”     "Type",     "Name",     "Introduction",     "Notification",     "FaceUrl",     "CreateTime",     "Owner_Account",     "LastInfoTime",     "LastMsgTime",     "NextMsgSeq",     "MemberNum",     "MaxMemberNum",     "ApplyJoinOption"   ],   "SelfInfoFilter": [ // 自身在群内的信息     "Role", // 群内身份     "JoinTime", // 入群时间     "MsgFlag", // 消息屏蔽类型     "UnreadMsgNum" // 未读消息数量（IM 云后台计算得出）   ] }</pre>
返回结果示例	<pre>{   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0,   "TotalCount": 2, // 不论 Limit 和 Offset 如何设置，该值总是满足条件的群组总数   "GroupIdList": [     {       "GroupId": "@TGS#2J4SZEDEL",       "Type": "Public",       "LastMsgTime": 1425976500,       "MemberCount": 48,       "SelfInfo": {         "Role": "Owner",         "MsgFlag": "AcceptAndNotify",         "UnreadMsgNum": 1       }     }   ] }</pre>

	<pre>         }     },     {         "GroupId": "@TGS#2C5SZEAEF",         "Type": "ChatRoom",         "LastMsgTime": 1425976567,         "MemberCount": 120,         "SelfInfo": {             "Role": "Member",             "MsgFlag": "AcceptAndNotify",             "UnreadMsgNum": 0         }     } } </pre>
--	---

### 3.7 获取群组详细资料（高级接口）

API 名称	getGroupInfo
接口说明	<ol style="list-style-type: none"> <li>高级接口支持仅获取群组中的指定字段，通过过滤器 GroupBaseInfoFilter，MemberInfoFilter，AppDefinedDataFilter_Group 来控制，分别针对群基础资料、群成员、群维度的自定义字段，每个请求可以指定一个或者多个子过滤器；</li> <li>群基础资料、群成员子过滤器中，每个字段的含义详见群组系统综述；</li> <li>举例： <ol style="list-style-type: none"> <li>如果只需要获取群组名称，则仅有 GroupBaseInfoFilter 这一个子过滤器，该子过滤器的值为字符串数组，只有一个元素 “Name”；</li> <li>如果需要获取群组名称，以及群组维度的某个自定义字段，则仅有 GroupBaseInfoFilter 和 AppDefinedDataFilter_Group 这两个过滤器。</li> </ol> </li> </ol>
请求参数示例	<pre> {     "GroupIdList": [// 群组列表，必填         "@TGS#2J4SZEAE",         "@TGS#2C5SZEAEF"     ],     "GroupBaseInfoFilter": [ // 如果基础信息字段，请在此数组中添加         "Type",         "Name",         "Introduction",         "Notification",         "FaceUrl",         "CreateTime",         "Owner_Account",         "LastInfoTime",         "LastMsgTime",         "NextMsgSeq", </pre>

	<pre> "MemberNum", "MaxMemberNum", "ApplyJoinOption" ], "MemberInfoFilter": [ // 如果需要成员信息，请添加此数组 "Account", // 成员 ID "Role", "JoinTime", "MsgSeq", "MsgFlag", "LastSendMsgTime", "ShutUpUntil" ], "AppDefinedDataFilter_Group": [ // 群组维度的自定义字段过滤 "GroupTestData1", "GroupTestData2" ] } </pre>
返回结果示例	<pre> {   "ActionStatus": "OK",   "ErrorInfo": "", // 这里的 ErrorInfo 无意义，需要判断每个群组的 ErrorInfo   "ErrorCode": 0, // 这里的 ErrorCode 无意义，需要判断每个群组的 ErrorCode   "GroupInfo": [ // 返回结果为群组信息数组，为简单起见这里仅列出一个群     {       "GroupId": "@TGS#2J4SZEDEL",       "ErrorCode": 0, // 针对该群组的返回结果       "ErrorInfo": "", // 针对该群组的返回结果       "Type": "Public", // 群组类型       "Name": "MyFirstGroup", // 群组名称       "Introduction": "TestGroup", // 群组简介       "Notification": "TestGroup", // 群组通知       "FaceUrl": "http://this.is.face.url", // 群组头像 URL       "Owner_Account": "leckie", // 群主 ID       "CreateTime": 1426976500, // 群组创建时间(UTC 时间)       "LastInfoTime": 1426976500, // 最后群资料变更时间(UTC 时间)       "LastMsgTime": 1426976600, // 群内最后一条消息的时间(UTC 时间)       "NextMsgSeq": 1234, //       "MemberNum": 2, // 当前群成员数量       "MemberNumMax": 50, // 最大群成员数量       "ApplyJoinOption": "FreeAccess", // 申请加群处理方式       "AppDefinedData": [ // 群组维度的自定义字段         {           "Key": "GroupTestData1", // 自定义字段的 key           "Value": "xxxx" // 自定义字段的值         },         { </pre>

	<pre>                 "Key": "GroupTestData2",                 "Value": "abc\u0000\u0001" // 自定义字段支持二进制数据             }         ],         "MemberList": [ // 群成员列表             {                 "Member_Account": "leckie", // 成员 ID                 "Role": "Owner", // 群内角色                 "JoinTime": 1425976500, // 入群时间(UTC 时间)                 "MsgSeq": 1233,                 "MsgFlag": "AcceptAndNotify", // 消息屏蔽选项                 "LastSendMsgTime": 1425976500, // 最后发言时间(UTC 时间)                 "ShutUpUntil": 1431069882 // 禁言截至时间(UTC 时间)             },             {                 "Member_Account": "peter",                 "Role": "Member",                 "JoinTime": 1425976500, // 入群时间                 "MsgSeq": 1233,                 "MsgFlag": "AcceptAndNotify",                 "LastSendMsgTime": 1425976500, // 最后一次发消息的时间                 "ShutUpUntil": 0 // 0 表示未被禁言，否则为禁言的截止时间             }         ]     } </pre>
--	--

### 3.8 修改群基本资料

API 名称	modifyGroupBaseInfo
接口说明	
请求参数示例	<pre> {     "GroupId": "@TGS#2J4SZEAE", // 要修改哪个群的基础资料（必填）     "Name": "NewName", // 群名称（选填）     "Introduction": "NewIntroduction", // 群简介（选填）     "Notification": "NewNotification", // 群公告（选填）     "FaceUrl": "http://this.is.new.face.url", // 群头像（选填）     "MaxMemberNum": 500, // 最大群成员数量（选填）     "ApplyJoinOption": "NeedPermission", // 申请加群方式（选填）     "AppDefinedData": [ // 自定义字段（选填）         { </pre>

	<pre> "Key": "GroupTestData1", // 需要修改的自定义字段 key "Value": "NewData" // 自定义字段的新值     },     {         "Key": "GroupTestData2",         "Value": "" // 设置为空表示删除该项自定义字段     }     ]     } </pre>
返回结果示例	<pre> {     "ActionStatus": "OK",     "ErrorInfo": "",     "ErrorCode": 0 } </pre>

## 4 群成员管理

### 4.1 获取群成员列表

API 名称	getGroupMemberInfo
接口说明	1. 该接口的目的在于分页拉取单个群的成员列表, 同样支持调用者指定需要返回哪些信息;
请求参数示例	<pre> {     "GroupId": "@TGS#2J4SZEAE", // 群组 ID (必填)     "MemberInfoFilter": [ // 需要获取哪些信息, 如果没有该字段则为群成员全部资料         "Account", // 成员 ID         "Role",         "JoinTime",         "MsgSeq",         "MsgFlag",         "LastSendMsgTime",         "ShutUpUntil"     ],     "Limit": 100, // 最多获取多少个成员     "Offset": 200 // 从第多少个开始获取 } </pre>
返回结果示例	<pre> {     "ActionStatus": "OK",     "ErrorInfo": "",     "ErrorCode": 0,     "MemberNum": 2, // 本群组的群成员总数 } </pre>

	<pre>"MemberList": [ // 群成员列表     {         "Member_Account": "bob",         "Role": "Owner",         "JoinTime": 1425976500, // 入群时间         "MsgSeq": 1233,         "MsgFlag": "AcceptAndNotify",         "LastSendMsgTime": 1425976500, // 最后一次发消息的时间         "ShutUpUntil": 1431069882 // 禁言截至时间(秒数)     },     {         "Member_Account": " peter ",         "Role": " Member ",         "JoinTime": 1425976500,         "MsgSeq": 1233,         "MsgFlag": "AcceptAndNotify",         "LastSendMsgTime": 1425976500,         "ShutUpUntil": 0// 0 表示未被禁言，否则为禁言的截止时间     } ]</pre>
--	--

## 4.2 邀请好友加群

API 名称	addGroupMember
接口说明	
请求参数示例	<pre>{     "GroupId": "@TGS#2J4SZEDEL", // 要操作的群组（必填）     "MemberList": [ // 一次最多添加 500 个成员         {             "Member_Account": "tommy" // 要添加的群成员 ID（必填）         },         {             "Member_Account": "jared"         }     ] }</pre>
返回结果示例	<pre>{     "ActionStatus": "OK",     "ErrorInfo": "",     "ErrorCode": 0,     "MemberList": [         {             "Member_Account": "tommy",</pre>

	<pre>         "Result": 1    // 加人结果：0 为失败；1 为成功；2 为已经是群成员       },       {         "Member_Account": "jared",         "Result": 1       }     ]   } </pre>
--	---

### 4.3 修改群消息提示

API 名称	modifyGroupMember
接口说明	消息屏蔽类型包括如下几种：AcceptAndNotify 表示接收并提示；AcceptNotNotify 表示接收不提示；Discard 表示屏蔽群消息（不会下发）。
请求参数示例	<pre> {   "GroupId": "@TGS#2CLUZEAEJ", //要操作的群组（必填）   "Member_Account": "bob", // 要操作的群成员（必填）   "MsgFlag": "AcceptAndNotify"// 消息屏蔽类型（必填） } </pre>
返回结果示例	<pre> {   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0 } </pre>

### 4.4 修改群成员角色

API 名称	modifyGroupMember
接口说明	
请求参数示例	<pre> {   "GroupId": "@TGS#2CLUZEAEJ", //要操作的群组（必填）   "Member_Account": "bob", // 要操作的群成员（必填）   "Role": "Admin", // Admin 或者 Member，分别为设置/取消管理员（必填） } </pre>
返回结果示例	<pre> {   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0 } </pre>



## 4.5 设置群成员禁言时间

API 名称	forbidSendMsg
接口说明	1. 所谓禁言，即禁止某些用户在一段时间内发言。
请求参数示例	<pre>{   "GroupId": "@TGS#2C5SZEAEF",   "Members_Account": [ // 最多支持 500 个     "peter",     "leckie"   ],   "ShutUpTime": 60 // 禁言多少秒，为 0 表示取消禁言 }</pre>
返回结果示例	<pre>{   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0 }</pre>

## 4.6 删除群成员

API 名称	deleteGroupMember
接口说明	1. 只要操作者的权限足够，即使被删除的用户原本就不在群组中，该接口依然返回成功
请求参数示例	<pre>{   "GroupId": "@TGS#2J4SZEAE", //要操作的群组（必填）   "Silence": 1, // 是否静默加入（选填，目前不支持）   "MemberToDel_Account": [ // 要删除的群成员列表，最多 500 个     "tommy",     "jared"   ] }</pre>
返回结果示例	<pre>{   "ActionStatus": "OK",   "ErrorInfo": "",   "ErrorCode": 0 }</pre>

## 5 错误码列表

错误码	错误说明
10001	调用时所携带的签名校验失败。
10002	系统错误。
10003	请求命令非法。
10004	参数非法。
10005	请求参数示例中携带的用户数量过多（超过了 500 个）。
10006	被频率限制。
10007	操作权限不足。
10008	非法请求。
10009	该群不允许群主主动退出。
10010	群组不存在，或者曾经存在过，但是目前已经被解散。
10011	解析 Json 包体失败。
10012	发起操作的用户 ID 非法。
10013	被操作的用户已经是群成员。
10014	群满员，无法再加入新成员。
10015	群组 ID 非法。
10016	IM 云发起第三方回调之后，APP 后台返回失败。
10017	因被禁言而不能发送消息。
10018	响应包大小超限，导致无法正确应答，请尝试减少需要获取的数据量。
10019	被操作的用户 ID 非法。