楼院议事厅功能采用极光IM的web方案进行开发

web客户端api文档 <https://docs.jiguang.cn//jmessage/client/im_sdk_js_v2/>

服务端api文档<https://docs.jiguang.cn//jmessage/server/rest_api_im/>

sdk下载<https://docs.jiguang.cn//jmessage/resources/>

为结合现有会议功能，将楼院议事厅作为一个会议进行维护，每个提案可生成一个议事厅

聊天功能采用群组形式，提案人作为群主

# 功能设计

## 鉴权信息管理

### 秘钥保存

在极光应用运营平台新建应用WEB\_IM

appkey appscret 保存在 community\_message.m\_app表中

初始化鉴权信息查询

后端根据appkey appscret组装鉴权信息供前端初始化时使用

{  
   "appkey": "7e42e869baa2fbca8ccb823c",  
   "random\_str": "022cd9fd995849b58b3ef0e943421ed9",  
   "signature": "E422A978DE37196588531CD0C42010B5",  
   "timestamp": "1467967210887"  
}  
参数说明:  
​  
appkey : 开发者在极光平台注册的 IM 应用 appkey  
random\_str : 20-36 长度的随机字符串, 作为签名加 salt 使用  
timestamp : 当前时间戳，用于防止重放攻击，精确到毫秒  
signature : 签名，10 分钟后失效（只针对初始化操作，初始化成功则之后的操作跟签名无关）  
签名生成算法如下:  
​  
signature = md5(appkey={appkey}&timestamp={timestamp}&random\_str={random\_str}&key={secret})  
其中 secret 为开发者在极光平台注册的 IM 应用 masterSecret。 签名生成示例:  
​  
signature = md5("appkey=25b693b31d2c2ad5f072ef0c&timestamp=1507791639926&random\_str=022cd9fd995849b&key=bc2efab258f2019727a4f36l")  
\*生产环境签名的生成需要在开发者服务端生成，不然存在 masterSecret 暴露的风险

## 用户管理

### **用户信息查询**

后端提供当前用户信息查询接口，根据当前微信登陆用户查询baseuser信息

### **注册**

用户加入申请审核通过后，后端自动注册

username baseuser.id

password baseuser.id

nickname baseuser.realname

### **登陆**

web端调用极光SDK登陆

注册登陆若要用户无感知，密码只能由我们自己决定，存在安全问题

若需用户自行注册登陆，至少需要先设置密码 并提供密码重置等相关接口

## 楼院议事厅

### **开启议事厅**

app 提案编辑页通过chatMeetingStatus字段控制开启（“1”）/ 关闭（“0”）

创建会议--根据提案人的电话查询基础用户信息--加入会议成员--以其作为群主创建群组--[若报错用户不存在898002，注册用户，再次创建]

### **进入议事厅**

若提案已开启议事厅，公众号我的楼院提案展示页中展示进入议事厅按钮，点击后验证身份（成员、非成员、群主），若是成员直接进入，并根据groupId获取历史信息和监听实时信息。

后端提供申请接口，向提案人发送消息，并记录申请信息，不可重复申请

mongodb存储申请，用完就删

### **审核申请**

app我的提案详情页审核加入申请进入申请列表，处理申请

同意 用户注册，将用户加入议事厅成员

拒绝 删除申请

### **关闭议事厅**

后端提供关闭议事厅接口，web端调用关闭议事厅接口成功后，发送特定内容的群组透传信 息，同时在透传消息监听事件中做登出处理，之后本页面用户登出

### **移出成员**

后端提供移出成员接口，同时监听移出事件，做登出处理并进行相应提示（所有人提示xxx被xxx移出）

### **禁言**

后端提供禁言接口web监听被禁言事件，所有人提示xxx被xxx禁言, 被禁言人发消息时提示自己被 禁言）

禁言特定时间

记录禁言信息保存在redis中设置过期时间，任务每30秒检查所有被禁言成员更新禁言状态

### **会议记录查询**

将楼院议事厅归为新类型的会议，后台管理系统在会议管理可查看相应信息

### **上传会议纪要**

将楼院议事厅归为新类型的会议，后台管理系统在会议管理可上传会议纪要

### **会议纪要展示**

公众号和app在提案详情页面提供会议纪要预览功能

<a :href="filepreview(url)" target="\_blank">{{content}}</a>

## 消息

### **发送消息**

前端调用web SDK JMessage#sendGroupMsg()

同时监听消息

### **复制**

前端提供复制功能

### **@**

记录@用户id（baseuserId）到extras下atUserIdList字段

历史消息返回extras下atUserList字段 @用户信息

~~前端通过 JMessage#sendGroupMsg() at\_list custom\_notification 参数处理消息~~ 消息查询接口msg信息at\_list 返回的是用户的uid,无法使用

### **引用**

记录引用消息id到extras下quoteMsgId字段，监听到此消息时，同时将引用内容展示, 页面效果仿照微信

历史消息中通过extras下quoteMsg字段返回所引用消息，若所引用消息处于非正常状态（已删除、已撤回）其内容仅包含状态status字段，前端需先验证quoteMsg 非空 再验证status 进行处理

### **撤回**

web通过sdk调用撤回方法JMessage#msgRetract ()，监听撤回事件，对消息隐藏，若该消息被引用将引用内容重设为“引用内容已撤回”，调用后台消息撤回接口，若消息已下载将该消息记录为撤回，否则记录撤回消息 id在下载任务中将其标为撤回

### **删除**

调用后台消息删除接口，若消息已下载将该消息记录为删除，否则记录删除消息 id在下载任务中将其标为删除

调用成功后，发送透传消息，同时监听透传消息，将已展示的此消息从页面删除

撤回与删除存在延迟与遗漏问题，立即执行下载消息任务是否能解决

### **文件消息**

实时消息

接收方监听并调用获取资源方法JMessage#getResource ()获取资源路径

历史消息

消息保存时将文件下载到本地，并提供本地路径字段，web展示时先判断本地路径是否存在，若没有则调用获取资源方法JMessage#getResource ()获取资源路径

### **消息保存**

采用mongodb保存消息实体，文件类型下载到文件服务器，并记录地址

### **历史消息查询**

由于websdk未提供本地消息保存，只能采用本地缓存或者后台下载+离线消息漫游（最多 100条保存15天）可能存在遗漏或延迟等问题