

消息列表实现概述

本篇主要介绍保利威聊天室消息列表基本实现逻辑。

一、历史聊天消息

消息列表中的消息来源，主要分为历史消息以及实时消息，而历史消息又有场次历史消息和频道历史消息之分。

场次历史消息

每场直播都会对应一个唯一的场次id (sessionId)，该场直播中所有用户的发言都关联了该 id，可以通过 `getSessionHistory` 方法获取该场次直播的历史聊天消息。

可根据返回数据中 `curPage` 跟 `totalPage` 字段的对比，判断当前是否加载到末页。

```
async function getSessionHistory() {
  const res = await chat.getSessionHistory({
    // 场次 id(非必传，默认使用实例的 sessionId)
    sessionId: '',
    // 页数
    page: 1,
    // 条数(非必传，默认10)
    size: 20,
  });
  const { curPage, totalPage, count, size, data = [] } = res;
  const isLastPage = curPage === totalPage;
  console.log('场次消息列表数据: ', data);
}
```

频道历史消息

可通过 `getChannelHistory` 方法获取频道所有场次的聊天消息。注意，每个频道都有允许保存的最大消息数量，因此当消息量超过最大数量时，会优先舍弃旧消息。

```
async function getChannelHistory() {
  let start = 0;
  let end = 9;

  const msgList = await chat.getChannelHistory({
    // 房间号(若不传, 默认为实例中的房间号/频道号)
    roomId: '',
    // 开始索引
    start,
    // 结束索引
    end,
    // 是否获取全部信息。1表示获取全部(默认), 0表示过滤图片和自定义消息
    fullMessage: 1,
    // 是否仅获取特殊角色发言(讲师、助教、嘉宾), 后端默认 0(全部获取)
    getSpecialMessage: 0,
  });

  console.log('频道消息列表数据', msgList);
  // 可根据返回的消息列表长度是否满足索引跨度, 来判断当前是否加载到末尾。
  if (msgList.length < end - start) {
    console.log('已加载到消息列表末尾');
  }
}
```

二、实时收发聊天消息

他人的文字发言、发送图片、消息被删除、自定义消息等, 均会在 SDK 内触发相应事件, 可监听这些事件, 并根据参数去处理实际所需显示的消息。

接收消息

此处列举所有与聊天室消息相关的事件, 详细事件、参数、字段等, 请参考 API 文档。以下摘取部分常用事件:

事件名	说明
chat.events.SPEAK	他人发言
chat.events.CHAT_IMG	他人聊天图片消息
chat.events.SEND_MESSAGE	个人发言。调用发言方法后将触发该事件, 此时可将个人发言插入到本地消息列表。
chat.events.SELF_CHAT_IMG	个人开始发送图片消息。在图片开始上传时, 即触发该事件, 可基于该事件将消息插入本地进行展示, 并展示图片上传状态。

事件名	说明
chat.events.SYS_MSG	系统消息。比如发送消息过快触发限流时，会触发该事件。可在监听到此事件后给予用户反馈。
chat.events.CUSTOMER_MESSAGE	自定义消息
chat.events.REMOVE_HISTORY	管理员清空聊天室消息。收到该事件时，聊天室记录将被清除，相应地前端也需要将展示中的消息进行清除。
chat.events.REMOVE_CONTENT	管理员移除单条消息。可根据id将对应消息删除。
chat.events.CLOSEROOM	关闭聊天室。管理员等特殊角色关闭了聊天室，此时普通用户禁止发言，前端可给予相应提示。
chat.events.OPENROOM	开启聊天室。恢复普通用户发言权限。

消息类型列表

为了便于快速集成，本 SDK 基于保利威目前的消息列表 UI 展示形式对部分聊天室事件进行了分类封装，详细类型常量可通过 `chat.msgTypes` 访问。

部分需要展示的事件数据会含有 `msgType` 字段，只需要将其插入本地消息列表，并根据该字段展示对应形式的组件/样式即可。同样，历史消息也会封装此字段。下面列举各 `msgType` 对应的事件。

msgType	消息类型
NORMAL	SEND_MESSAGE、SPEAK
CHAT_IMG	CHAT_IMG、CHAT_IMG_UPLOAD_PROGRESS、SELF_CHAT_IMG_RESULT(发送图片相关消息，详细可见 发送图片消息)
CUSTOMER_MESSAGE	CUSTOMER_MESSAGE
SYSTEM	SYS_MSG(本地触发，如发言频繁)
RED_PAPER_RESULT	RED_PAPER_RESULT(红包被领取消息)
REWARD	REWARD(打赏消息)

在监听到上述事件时，可以考虑将消息体加入到待显示消息列表中，然后根据具体 `msgType` 采用相应组件进行渲染展示。

发送消息

发送文字消息

可使用 `send` 方法发送普通文字消息。

```
chat.send(  
  '消息内容',  
  // 第二个参数为引用消息，非必传。  
  // 特殊角色使用公屏回复功能时需传入(完整传入，使SDK内部可封装、触发SEND_MESSAGE事件)  
  {  
    id,  
    // ...  
  }  
);
```