

聊天重放

一、功能概述

- 聊天室模块(chat) 提供聊天重放功能 Api, 用于开发者集成聊天重放功能。
- [聊天重放-基础使用示例](#)

二、使用方式 - 方法

2.1 初始化聊天消息重放

Api 方法: `initChatMsgReplay(): Promise<void>`

从 v0.3.0 该方法需要在 `setChatMsgReplayPlaybackTarget` 之后去执行, 聊天室模块才能正常地去初始化聊天重放记录, 使用过程中, 会回调 `ChatEvents.ChatMsgReplayStatusChange` 事件 版本开始支持

示例:

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatMsgReplayStatusChange, ({ status }) => {
  console.info(status)
})

// 需要先执行 setChatMsgReplayPlaybackTarget 方法
await watchCore.chat.initChatMsgReplay()
```

2.3 获取聊天消息重放历史数据

通过 `getChatMsgReplayHistoryList` 来获取聊天消息重放的历史数据

Api 方法: `getChatMsgReplayHistoryList(options: Object): Promise<ChatReplayMsgType[]>`

从 v0.3.0 版本开始支持

参数说明:

- options: `Object` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
mode	聊天重放请求历史数据模式	ChatReplayRequestHistoryMode	是	-
replayMsg	根据指定的模式传入对应的重放消息，比如需要加载旧数据，则传入当前聊天数据中的第一条消息	ChatReplayMsgType	否	-
count	聊天数量，限制最大为 60	number	是	-

返回值说明：聊天消息重放历史数据，`Promise<ChatReplayMsgType[]>` 类型

示例：

```
import { ChatReplayRequestHistoryMode } from "@polyv/live-watch-sdk"
const data = await watchCore.chat.getChatMsgReplayHistoryList({
  mode: ChatReplayRequestHistoryMode.Init,
  count: 20
})
```

2.4 处理聊天消息重放

外部使用时，需要结合观看页 SDK 的 `PlayerEvents.TimeUpdate` 事件回调来一起使用，使用过程中，可以通过 `ChatEvents.ChatMessage`` 事件来获取重放的消息 由于回放时可以通过播放器进度条快速跳转，此时可以通过 `ChatEvents.ChatMsgReplayReload` 事件来监听是否需要重新加载聊天室数据

Api 方法：`processChatMsgReplay(params: Object): void`

从 v0.3.0 版本开始支持

参数说明：

- params: `Object` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
currentTime	当前播放时间，单位：秒	number	是	-
durationTime	播放总时长，单位：秒	number	是	-

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatMessage, ({ chatMsg })=> {
  console.info(chatMsg);
})

watchCore.chat.eventEmitter.on(ChatEvents.ChatMsgReplayReload, ()=> {
  console.info("重新加载聊天室数据");
})

watchCore.player.eventEmitter.on(PlayerEvents.TimeUpdate, params => {
  watchCore.chat.processChatMsgReplay(params);
});
```

三、使用方式 - 事件

3.1 聊天消息重放状态

Event 事件： ChatEvents.ChatMsgReplayStatusChange

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
status	重放状态	ChatMsgReplayStatus

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatMsgReplayStatusChange, (params) => {
  console.log('聊天重放状态变更： ', params.status);
});
```

3.2 聊天消息重放-重新加载

说明： 通知外部重新加载聊天重放的数据

Event 事件： ChatEvents.ChatMsgReplayReload

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
seconds	-	number

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatMsgReplayReload, () => {
  // initChatMsgRender
  console.log('需要重新加载聊天重放数据渲染表格');
});
```

3.3 聊天消息事件

说明： 当收到聊天消息后触发该事件

Event 事件： ChatEvents.ChatMessage

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
chatMsg	聊天消息对象	ChatMsgType
isChatReplayMsg	是否为聊天重放消息	boolean

示例：

```
// 聊天消息列表
const chatMsgList: ChatMsgType[] = [];

// 聊天消息事件
watchCore.chat.eventEmitter.on(ChatEvents.ChatMessage, (data) => {
  // 插入到聊天消息列表
  chatMsgList.push(data.chatMsg);
  // 渲染聊天消息...
});
```