

模块事件

一、聊天室信息修改

Event 事件： `ChatEvents.ChatInfoChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>chatInfo</code>	聊天室信息	<code>ChatModuleInfo</code>

二、聊天室连接失败事件

说明： 当断网或其他因素导致聊天室链接失败时触发该事件

Event 事件： `ChatEvents.ChatConnectFail`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>reason</code>	连接失败原因	<code>ChatConnectFailReason</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatConnectFail, (data) => {
  confirm({
    message: '聊天室连接失败，无法与其他人互动，立即刷新重试？',
    onConfirm: () => location.reload(),
  });
});
```

三、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
<code>chatMsg</code>	聊天消息对象	<code>ChatMsgType</code>

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...
});
```

四、替换聊天消息事件

当调用 `sendSpeakMsg` 等发送消息方法中，发送到服务端前就会回调 `ChatEvents.ChatMessage` 事件，此时消息 id 为本地 id，发送到服务端并回调了消息 id 后触发该事件，收到该消息后根据 id 更新成新的消息对象和渲染信息。

另外发送图片消息如果服务端检测到违规图后也通过该事件更新。

Event 事件： `ChatEvents.ReplaceChatMessage`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>id</code>	替换的消息 id	<code>string</code>
<code>chatMsg</code>	新的消息对象	<code>ChatMsgType</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ReplaceChatMessage, (data) => {
  // 需要被替换的消息 id
  const replaceId = data.id;
  // 新的消息对象
  const chatMsg = data.chatMsg;

  const index = chatMsgList.findIndex((item) => item.id === replaceId);
  if (index !== -1) {
    chatMsgList[index] = chatMsg;
  }
  // 将视图的消息节点替换...
});
```

五、点赞事件

说明：通过该事件监听用户的点赞事件

Event 事件： ChatEvents.ChatLike

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
count	点赞数	number
realtimeLikes	实时点赞数	number
userId	用户 id	string
nick	用户昵称	string
isSelf	是否自己点赞	boolean

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatLike, (data) => {
  console.log('实时点赞数: ', data.realtimeLikes);
  console.log('该观众的点赞次数', data.count);
  console.log('点赞的观众 id', data.userId);
  console.log('点赞的观众昵称', data.nick);
});
```

六、点赞数修改事件

说明： 通过该事件监听点赞数改变事件，回调后更新页面的点赞数显示。

Event 事件： `ChatEvents.ChatLikeCountChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>realtimeLikes</code>	实时点赞数	<code>number</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatLikeCountChange, (data) => {
  console.log('实时点赞数: ', data.realtimeLikes);
});
```

七、聊天室用户登录事件

说明： 有观众进入聊天室后触发该事件

Event 事件： `ChatEvents.ChatUserLogin`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>user</code>	用户信息	<code>ChatMessageUser</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatUserLogin, (data) => {
  const user = data.user;
  toast.info(`欢迎 ${user.nick} 进入`);
});
```

八、聊天室用户登出事件

说明： 当观众退出聊天室后触发该事件

Event 事件： ChatEvents.ChatUserLogout

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
userId	用户 userId	string

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatUserLogout, (data) => {
  const userId = data.userId;
  console.log('退出的用户 id', userId);
});
```

九、当前用户重复登录事件

说明： 当用户重复登录聊天室时触发该事件，触发后当前页面将无法接收到任何聊天室消息

Event 事件： ChatEvents.CurrentUserReLogin

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.CurrentUserReLogin, (data) => {
  toast.error('您已在其他地方登录，3 秒后将推出该页面');
  setTimeout(() => {
    location.replace('跳出到页面地址');
  }, 3000);
});
```

十、清空聊天室事件

说明： 管理员清空聊天历史记录后触发该事件

Event 事件： ChatEvents.ClearMsgHistory

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ClearMsgHistory, () => {
  console.log('管理员清空历史记录，todo 清空聊天记录列表');
});
```

十一、删除某条消息事件

说明： 管理员删除某条历史消息后触发该事件

Event 事件： `ChatEvents.RemoveChatMsg`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>id</code>	删除的消息 id	<code>string</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.RemoveChatMsg, (data) => {
  console.log('管理员删除历史消息， 删除的消息 id: ', data.id);
});
```

十二、聊天室关闭事件

说明： 讲师或管理员关闭聊天室后触发该事件

Event 事件： `ChatEvents.CloseChatRoom`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.CloseChatRoom, () => {
  watchCore.chat.sendSystemMsg('聊天室已关闭');
});
```

十三、聊天室开启事件

说明： 讲师或管理员开启聊天室后触发该事件

Event 事件： `ChatEvents.OpenChatRoom`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.OpenChatRoom, () => {
  watchCore.chat.sendSystemMsg('聊天室已打开');
});
```

十四、在线人数改变事件

说明： 当观众上线/下线时，会回调该事件用于实时获取在线人数

Event 事件： `ChatEvents.OnlineUserCountChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>onlineUserCount</code>	实时在线人数	<code>number</code>

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.OnlineUserCountChange, (data) => {
  console.log('在线人数改变： ', data.onlineUserCount);
});
```