模块事件

一、V2 的聊天室 SDK 安装完成回调

Event 事件： `ChatEvents.ChatSdkV2Setuped`

二、聊天室信息修改

聊天室模块会保存基本的聊天室状态信息，通过该事件监听聊天室状态改变

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
`chatInfo`	聊天室信息	`ChatModuleInfo`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatInfoChange, (evt) => {
  const chatInfo = evt.chatInfo;
  console.log('聊天室是否已关闭', chatInfo.chatRoomIsClosed);
});
```

三、聊天室重连成功事件

说明： 当聊天室断连重连成功时触发该事件

Event 事件： `ChatEvents.ChatReconnectSuccess`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatReconnectSuccess, () => {
  console.info("重连成功！")
});
```

四、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
`reason`	连接失败原因	`ChatConnectFailReason`

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(),
  });
});
```

五、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...
});
```

六、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
`id`	替换的消息 id	`string`
`chatMsg`	新的消息对象	`ChatMsgType`

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...
});
```

七、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);
});
```

八、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
`realtimeLikes`	实时点赞数	`number`

示例：

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

九、情绪反馈事件

说明： 通过该事件监听用户的情绪反馈事件

Event 事件： `ChatEvents.ChatEmotionalFeedback`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
`type`	情绪类型	`EmotionalFeedbackType`
`count`	数量	`number`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.ChatEmotionalFeedback, (data) => {
  console.log('类型：', data.type); // EmotionalFeedbackType
  console.log('数量：', data.count);
});
```

十、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
`user`	用户信息	`ChatMessageUser`

示例：

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

十一、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);
});
```

十二、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`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
`causeBy`	错误来源	`string`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.CurrentUserRelogin, (data) => {
  toast.errot('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);
});
```

十三、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');
});
```

十四、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
`id`	删除的消息 id	`string`

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);
});
```

十五、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');
});
```

十六、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.CloseChatRoom, () => {
  watchCore.chat.sendSystemMsg('聊天室已打开');
});
```

十七、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
`onlineUserCount`	实时在线人数	`number`

示例：

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

十八、在线用户列表改变事件

说明： 当开始轮询在线用户列表时，会回调该事件\

Event 事件： `ChatEvents.OnlineUserListChange`

回调参数： `UserListResult`

示例：

```
watchCore.chat.eventEmitter.on(ChatEvents.OnlineUserListChange, (data) => {
  console.log('在线用户列表：', data.userlist);
});
```

十九、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');
});
```

二十、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);
});
```

二十一、超出直播间最大在线人数

Event 事件： `ChatEvents.OverMaxOnlineCount`

二十二、白名单移除用户

Event 事件： `ChatEvents.WhiteListRemoveUser`

二十三、评论上墙事件

Event 事件： `ChatEvents.SpeakToTop`

回调参数： `ChatMsgSpeakTopType`

二十四、评论取消上墙事件

Event 事件： `ChatEvents.SpeakCancelTop`

回调参数： `Object | undefined`

二十五、超出频道登录限制事件

Event 事件： `ChatEvents.OverLoginLimit`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
`isStartLiveOverLogin`	刚开始直播时, 由于超过频道限制人数所触发的RELOGIN	`boolean`