

使用方式

一、加载表情解析器

开发者需要在调用表情模块的其他方法前，需要优先调用该方法

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

示例：

```
// 安装观看页 SDK
await watchCore.setup();
// 加载表情功能
await watchCore.interactReceive.setupEmotionParser();
// 加载后即可调用表情模块的其他方法
```

二、获取黄脸表情节点

开发者可通过 `getEmotionFaceDomList` 方法获取节点并插入到页面容器中。

监听容器的点击事件并调用 `getEmotionFaceClickTarget` 方法可获取观众点击时的黄脸表情。

Api 方法： `getEmotionFaceDomList(lang?: LanguageType): HTMLElement`

参数说明：

- lang: `LanguageType` 类型，选传

返回值说明：黄脸表情 dom 节点，`HTMLElement` 类型

示例：

```
const containerElem = document.querySelector('容器选择器');
// 插入表情节点
const elems = watchCore.chat.getEmotionFaceDomList();
containerElem.appendChild(elems);
// 监听容器节点的点击事件
containerElem.addEventListener('click', function(event) {
  const title = watchCore.emotion.getEmotionFaceClickTarget(event);
  console.log('点击的表情', title);
});
```

三、获取点击事件中的黄脸表情

用于获取观众点击黄脸表情容器时的表情对象，获取后插入到观众正在输入的聊天消息文本中。

Api 方法： `getEmotionFaceClickTarget(event: Event): undefined | string`

参数说明：

- event：点击事件，`Event` 类型，必传

返回值说明： 黄脸表情字符，`undefined | string` 类型

示例：

```
// 观众正在输入的聊天消息
const inputMsg = '今天天气真好';

// 监听容器节点的点击事件
containerElem.addEventListener('click', function(event) {
  const title = watchCore.emotion.getEmotionFaceClickTarget(event);
  if (title) {
    console.log('点击的表情', title); // 如：[呲牙]、[害羞]
    inputMsg = inputMsg + title;
  }
});
```

四、把表情名称转成中文

Api 方法： `translateEmotions(content: string, lang?: WatchSdkLangEnum): string`

参数说明：

- content：发言内容，`string` 类型，必传
- lang：`WatchSdkLangEnum` 类型，选传

返回值说明： 转换后的内容

示例：

```
// 聊天发言内容
let content = '';
// 返回转换后的内容
let translatedContent = await watchCore.emotion.translateEmotions(content);
```

五、获取表情图片列表

通过 `getEmotionImages` 获取标题图片列表，获取该列表后，使用 `id`、`url` 调用 `ChatModule.sendEmotionImageMsg` 发送表情图片消息。

Api 方法： `getEmotionImages(): Promise<EmotionImageData[]>`

返回值说明： `Promise<EmotionImageData[]>` 类型

示例：

```
const emotionImages = await watchCore.emotion.getEmotionImages();
// 表情图片列表数据示例：
[
  { id: '0', title: '收到', url: 'https://s2.videocc.net/default-img/img-emotion/v1/shoudao.png' },
  ...
]

// 发送表情图片
const item = emotionImages[2];
watchCore.chat.sendEmotionImageMsg({
  emotionId: item.id,
  emotionUrl: item.url,
});
```