

频道信息

本文档主要讲述 频道模块(channel) 提供的频道信息获取相关的 API 文档，详细内容见下文：

一、频道信息

1.1 获取频道基础信息

根据语言类型获取频道基础信息（含英文配置），如：频道名称、频道介绍、主持人名称。

Api 方法： `getChannelBasicInfo(type?: LanguageType): ChannelBasicInfo`

参数说明：

- type: 语言类型， `LanguageType` 类型，选传，默认 `'zh_CN'`

返回值说明： 频道基础信息， `ChannelBasicInfo` 类型，详细类型说明如下

属性名	说明	类型
<code>title</code>	频道标题	<code>string</code>
<code>publisher</code>	频道主持人	<code>string</code>
<code>description</code>	频道描述	<code>string</code>

示例：

```
// 获取频道的基础信息
const cnBasicInfo = watchCore.channel.getChannelBasicInfo();
console.log('频道名称', cnBasicInfo.title);
console.log('频道介绍', cnBasicInfo.description);
console.log('主持人名称', cnBasicInfo.publisher);

// 获取频道的英文基础信息
const enBasicInfo = watchCore.channel.getChannelBasicInfo('en');
console.log('英文频道名称', enBasicInfo.title);
```

1.2 获取频道场景

用于判断频道类型，如是否为三分屏的频道，是否为研讨会的频道

Api 方法： `getChannelScene(): ChannelScene`

返回值说明： `ChannelScene` 类型

示例：

```
import { ChannelScene } from '@polyv/live-watch-sdk';
const scene = watchCore.channel.getChannelScene();
const isAloneChannel = scene === ChannelScene.Alone; // 普通直播频道
const isPptChannel = scene === ChannelScene.Ppt; // 三分屏频道
const isSeminarChannel = scene === ChannelScene.Seminar; // 研讨会频道
```

1.3 获取当前频道的观看页地址

用于获取当前观看页的地址。

Api 方法： `getChannelWatchUrl(): string`

示例：

```
const watchUrl = watchCore.channel.getChannelWatchUrl();
console.log('当前观看页的打开地址', watchUrl);
```

1.4 生成观看页地址

Api 方法： `generateWatchUrl(channelId: string): string`

参数说明：

- channelId: 频道号, `string` 类型, 必传

示例：

```
const watchUrl = watchCore.channel.generateWatchUrl('频道号');
location.href = watchUrl;
```

1.5 获取讲师信息

用于获取当前讲师/主讲的信息。

Api 方法： `getTeacherInfo(): TeacherInfoType`

返回值说明： `TeacherInfoType` 类型

示例：

```
const teacherInfo = watchCore.channel.getTeacherInfo();
console.log('讲师昵称', teacherInfo.nick);
console.log('讲师头像', teacherInfo.pic);
console.log('讲师头衔', teacherInfo.actor);
```

1.6 获取渠道 id

通过后台设置的渠道进入观看页后，在创建观看页 SDK 实例时传入 `promoteId` 渠道号配置，可通过该方法获取当前观众的渠道 id，用作其他处理。

Api 方法： `getPromoteId(): undefined | string`

返回值说明： `undefined | string` 类型

示例：

```
import { createWatchCore } from '@polyv/live-watch-sdk';
const watchCore = createWatchCore({
  channelId: '频道号',
  promoteId: '渠道号',
});

// 需要重新获取渠道号时
const promoteId = watchCore.channel.getPromoteId();
```

1.7 获取频道协议配置

可以获取在不同语言下的协议配置，主要用于授权类业务

Api 方法： `getChannelAgreementConfig(): Promise<ChannelAgreementConfig>`

从 v0.10.0 版本开始支持

返回值说明： `Promise<ChannelAgreementConfig>` 类型

示例：

```
const agreementConfig = await watchCore.channel.getChannelAgreementConfig();
console.log('当前语言的协议配置', agreementConfig["zh_CN"]);
```

1.8 获取频道的直播场景

Api 方法：`getNewScene(): ChannelNewScene`

从 v2.13.0 版本开始支持

返回值说明：`ChannelNewScene` 类型

1.9 重置频道详情(慎用)

Api 方法：`resetChannelDetail(): void`

从 v2.17.0 版本开始支持

二、直播状态

2.1 频道直播状态

Enum 枚举：`LiveStatus`

常量	枚举成员	说明
'live'	<code>LiveStatus.Live</code>	直播中
'waiting'	<code>LiveStatus.Waiting</code>	等待中
'end'	<code>LiveStatus.End</code>	已结束，无直播
'playback'	<code>LiveStatus.Playback</code>	回放中
'stop'	<code>LiveStatus.Stop</code>	直播暂停中
'unStart'	<code>LiveStatus.UnStart</code>	未开始

2.2 直播状态改变事件

说明： 当主讲开始/结束直播后，频道的直播状态都会改变，通过该事件监听直播状态改变

Event 事件： `ChannelEvents.LiveStatusChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>liveStatus</code>	新的直播状态	<code>LiveStatus</code>
<code>newLiveStatus</code>	新的直播状态	<code>LiveStatus</code>
<code>oldLiveStatus</code>	旧的直播状态	<code>LiveStatus</code>

示例：

```
watchCore.channel.eventEmitter.on(ChannelEvents.LiveStatusChange, (data) => {
  console.log('频道状态改变，新状态：', data.liveStatus);
});
```

2.3 获取频道直播状态

用于获取频道当前的直播状态，通过 `ChannelEvents.LiveStatusChange` 事件监听直播状态更改事件。

Api 方法： `getLiveStatus(): LiveStatus`

返回值说明： `LiveStatus` 类型

示例：

```
import { LiveStatus } from '@polyv/live-watch-sdk';

watchCore.channel.eventEmitter.on(ChannelEvents.LiveStatusChange, () => {
  const liveStatus = watchCore.channel.getLiveStatus();
  console.log('是否正在直播', liveStatus === LiveStatus.Live);
});
```

2.4 场次号改变事件

说明： 主讲开次新的一次直播后，都会生成一个新的直播场次号，可以通过频道模块的该事件监听直播场次号改变。

Event 事件： ChannelEvents.SessionIdChange

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
sessionId	新的场次号	string

示例：

```
watchCore.channel.eventEmitter.on(ChannelEvents.LiveStatusChange, (data) => {
  console.log('频道状态改变，新状态： ', data.liveStatus);
});
```

2.5 获取频道直播场次号

通过 getCurrentSessionId 方法获取当前频道的最新场次号。

Api 方法： getCurrentSessionId(): string

返回值说明： 最新场次号

示例：

```
const currentSessionId = watchCore.channel.getCurrentSessionId();
console.log('频道最新场次号', currentSessionId);
```

2.6 关闭直播状态轮训

需要和 watchCore.channel.startLiveStatusPolling 方法配合使用

Api 方法： removeLiveStatusPolling(pollingId: string): void

参数说明：

- pollingId: 轮训 id, string 类型，必传

示例：

```
const watchCore = getWatchCore();
const pollingId = watchCore.channel.startLiveStatusPolling();
watchCore.channel.removeLiveStatusPolling(pollingId);
```

2.7 开始直播状态轮训

会根据后端配置，在条件允许的情况下进行直播状态轮询

Api 方法： `startLiveStatusPolling(pollingId?: string, options?: Object): string`

参数说明：

- pollingId: 轮训 id, `string` 类型，选传，默认 ...
- options: `Object` 类型，选传，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>timeout</code>	-	<code>number</code>	是	-
<code>maxPollingCount</code>	-	<code>number</code>	是	-

返回值说明： 轮询标识位，可用作移除轮询的参数

示例：

```
const watchCore = getWatchCore();
const pollingId = watchCore.channel.startLiveStatusPolling();
console.info('轮询标识位: ', pollingId);
```

三、推流信息

3.1 获取推流信息

用于获取当前频道的最新推流信息，如流的尺寸、推流类型等。

Api 方法： `getPushInfo(): Promise<StreamPushInfo>`

返回值说明： `Promise<StreamPushInfo>` 类型，详细类型说明如下

属性名	说明	类型
<code>isNewGuide</code>	是否导播台推流	<code>YN</code>
<code>resolutionWidth</code>	分辨率宽度	<code>number</code>
<code>resolutionHeight</code>	分辨率高度	<code>number</code>

属性名	说明	类型
streamType	推流类型	StreamType

示例：

```
const pushInfo = await watchCore.channel.getPushInfo();
console.log('流宽度', pushInfo.resolutionWidth);
console.log('流高度', pushInfo.resolutionHeight);
console.log('推流类型', pushInfo.streamType);
```

四、页面浏览次数

4.1 获取观看页浏览次数

如果不需要轮询通机制来更新浏览次数，可调用 `getPageViewCount` 方法来重新获取观众进入页面时的浏览次数。

Api 方法： `getPageViewCount(): number`

示例：

```
const pageViewCount = await watchCore.channel.getPageViewCount();
console.log('页面浏览次数', pageViewCount);
```

4.2 开启观看页浏览次数请求轮询

用于开启观看页浏览次数接口请求的轮询，在触发回调后更新页面的浏览次数。

从 v1.2.0 开始在方法调用后立即回调一次

Api 方法： `startPageViewPolling(callback?: Function): void`

参数说明：

- callback：回调函数，获取每次轮询的浏览次数，`Function` 类型，选传

示例：


```
watchCore.channel.startPageViewPolling((pageViewCount: number) => {
  console.log('页面浏览次数更新: ', pageViewCount);
});
```

4.3 获取最新的观看页浏览次数

Api 方法： `getLatestPageView(): Promise<number>`

从 v1.2.0 版本开始支持

返回值说明： `Promise<number>` 类型

4.4 停止观看页浏览次数请求轮询

当页面销毁或相关组件销毁时，调用 `stopPageViewPolling` 停止轮询器。

Api 方法： `stopPageViewPolling(): void`

示例：

```
// vue 代码
export default {
  // 组件销毁前停止定时器
  beforeDestroy() {
    watchCore.channel.stopPageViewPolling();
  }
};
```

4.5 获取浏览次数设置信息

用于获取管理后台设置的页面浏览次数信息。

Api 方法： `getPageViewSetting(): PageViewSetting`

返回值说明： 页面浏览次数设置，`PageViewSetting` 类型，详细类型说明如下

属性名	说明	类型
<code>pVShowEnabled</code>	页面浏览次数开关	<code>boolean</code>
<code>pageViewCount</code>	页面浏览次数	<code>number</code>

属性名	说明	类型
<code>mobilePvShowLocation</code>	移动端页面浏览次数位置	<code>PageViewShowLocation</code>

示例：

```
const setting = watchCore.channel.getPageViewSetting();
console.log(' 页面浏览次数开关 ', setting.pvShowEnabled);
console.log(' 页面浏览次数 ', setting.pageViewCount);
console.log(' 移动端浏览次数位置 ', setting.mobilePvShowLocation);
```

五、直播开始时间

5.1 获取直播倒计时设置

用于获取后台设置的频道开始时间等设置信息。

Api 方法： `getCountdownSetting(): ChannelCountdownSetting`

返回值说明： 倒计时设置， `ChannelCountdownSetting` 类型，详细类型说明如下

属性名	说明	类型
<code>liveStartTime</code>	直播开始时间，单位：时间戳	<code>undefined number</code>
<code>countdownEnabled</code>	倒计时开关	<code>boolean</code>
<code>playbackShowCountdownEnabled</code>	回放时是否显示下一场倒计时	<code>boolean</code>

示例：

```
const setting = watchCore.channel.getCountdownSetting();
console.log(' 开始时间： ', setting.liveStartTime);
console.log(' 回放下是否显示倒计时： ', setting.playbackShowCountdownEnabled);
```

5.2 获取信息卡片直播时间设置

`descLiveTimeType` 为 `actual` 且有实际开播时间时，卡片展示实际开播时间； 否则展示计划开播时间（`channelInfo.startTime`，新场次访问时为当前场次计划时间）。

Api 方法： `getDescLiveTimeSetting(): Object`

返回值说明： `Object` 类型，详细类型说明如下

属性名	说明	类型
<code>liveShowTime</code>	-	<code>undefined number</code>
<code>descLiveTimeType</code>	-	<code>'plan' 'actual'</code>

示例：

```
const setting = watchCore.channel.getDescLiveTimeSetting();
console.log('时间类型：', setting.descLiveTimeType);
console.log('展示时间：', setting.liveShowTime);
```

六、观看时长

6.1 获取观看时长设置

Api 方法： `getWatchDurationSetting(): ChannelWatchDurationSetting`

从 v0.8.0 版本开始支持

返回值说明： `ChannelWatchDurationSetting` 类型，详细类型说明如下

属性名	说明	类型
<code>watchDurationEnabled</code>	观看时长开关	<code>boolean</code>
<code>watchDurationType</code>	观看时长类型	<code>string</code>

6.2 结束观看时长统计轮询

Api 方法： `stopWatchDurationStatsPolling(): void`

从 v0.8.0 版本开始支持

6.3 开始观看时长统计轮询

Api 方法: `startWatchDurationStatsPolling(): void`

从 v0.8.0 版本开始支持

七、系列直播(直播新场次)

7.1 获取直播新场次开关

Api 方法: `getTargetSessionEnabled(): boolean`

从 v2.4.0 版本开始支持

7.2 获取频道场次列表

Api 方法: `getChannelSessionList(params?: GetChannelSessionListParams): Promise<PageContent<ChannelSessionDataItem>>`

参数说明:

- params: `GetChannelSessionListParams` 类型, 选传, 详细类型说明如下

参数名	说明	类型	必须	默认值
<code>pageNumber</code>	-	<code>number</code>	否	-
<code>pageSize</code>	-	<code>number</code>	否	-
<code>status</code>	-	<code>string</code>	否	-
<code>order</code>	-	<code>string</code>	否	-
<code>chapterId</code>	章节id (可选, 传入时仅返回该章节下的场次)	<code>number</code>	否	-

返回值说明: `Promise<PageContent<ChannelSessionDataItem>>` 类型

7.3 获取频道场次章节列表

Api 方法： `getChannelSessionChapterList(): Promise<ChannelSessionChapter[]>`

从 v2.18.0 版本开始支持

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

八、主讲信息

8.1 获取主讲信息

Api 方法： `getAnchorInfo(): Promise<undefined | ChannelAnchorInfo>`

从 v2.13.0 版本开始支持

返回值说明： `Promise<undefined | ChannelAnchorInfo>` 类型

8.2 获取主讲关联的频道列表

Api 方法： `getAnchorChannelList(options?: GetAnchorChannelListParams): Promise<PageContent<AnchorChannelItem>>`

从 v2.13.0 版本开始支持

参数说明：

- options: `GetAnchorChannelListParams` 类型，选传，默认 `{}`，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>pageNumber</code>	-	<code>number</code>	否	-
<code>pageSize</code>	-	<code>number</code>	否	-

返回值说明： `Promise<PageContent<AnchorChannelItem>>` 类型