

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-miniprogram-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
```

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： 当主播开始/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>

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-miniprogram-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：主播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
<code>sessionId</code>	新的场次号	<code>string</code>

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

Api 方法: `removeLiveStatusPolling(pollingId: string): void`

参数说明:

- `pollingId`: 轮训 id, `string` 类型, 必传

2.7 开始直播状态轮训

Api 方法: `startLiveStatusPolling(pollingId?: string): string`

参数说明:

- `pollingId`: 轮训 id, `string` 类型, 选传, 默认 ...

返回值说明: `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>
<code>streamType</code>	推流类型	<code>StreamType</code>

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);
```

4.2 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-miniprogram-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
```

4.3 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-miniprogram-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();
```

4.4 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);
```

4.5 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-miniprogram-sdk';

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

4.6 关闭直播状态轮训

Api 方法: `removeLiveStatusPolling(pollingId: string): void`

参数说明:

- `pollingId`: 轮训 id, `string` 类型, 必传

4.7 开始直播状态轮训

Api 方法: `startLiveStatusPolling(pollingId?: string): string`

参数说明:

- `pollingId`: 轮训 id, `string` 类型, 选传, 默认 ...

返回值说明： `pollingId`

4.8 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);
```

4.9 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。

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);
});
```

4.10 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();
  }
};
```

4.11 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>
<code>streamType</code>	推流类型	<code>StreamType</code>

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);
```

4.12 获取观看页设置

用于获取管理后台设置的观看页设置信息。

Api 方法： `getWatchSetting(): ChannelWatchSetting`

返回值说明： 频道观看页设置， `ChannelWatchSetting` 类型，详细类型说明如下

属性名	说明	类型
<code>watchEnabled</code>	观看页开关	<code>boolean</code>
<code>mobileWatchEnabled</code>	移动端观看页开关	<code>boolean</code>
<code>splashEnabled</code>	引导页开关	<code>boolean</code>

示例：

```
const setting = watchCore.channel.getWatchSetting();
if (!setting.watchEnabled) { alert('当前观看页暂未开放'); }
if (isMobile && !setting.mobileWatchEnabled) { alert('暂不支持移动端观看'); }
```

4.13 获取频道布局设置

用于获取管理后台的页面布局相关设置。

Api 方法：`getLayoutSetting(): ChannelLayoutSetting`

返回值说明：频道布局设置，`ChannelLayoutSetting` 类型，详细类型说明如下

属性名	说明	类型
<code>mainScreenLayoutMode</code>	三分屏主屏幕布局模式	<code>MainScreenLayoutMode</code>
<code>mobileSplashLayout</code>	移动端引导页布局	<code>MobileSplashLayout</code>
<code>mobileWatchLayout</code>	移动端观看页布局	<code>MobileWatchLayout</code>

示例：

```
const setting = watchCore.channel.getLayoutSetting();
console.log('三分屏主屏幕布局模式：', setting.mainScreenLayoutMode);
console.log('移动端引导页布局：', setting.mobileSplashLayout);
console.log('移动端观看页布局：', setting.mobileWatchLayout);
```

4.14 获取频道皮肤主题设置

用于获取管理后台的皮肤主题设置。

Api 方法：`getThemeSetting(): ChannelThemeSetting`

返回值说明： 频道皮肤主题设置， ChannelThemeSetting 类型，详细类型说明如下

属性名	说明	类型
pageSkin	观看页皮肤	ChannelWatchPageSkin
browserFavIcon	浏览器标签页图标	undefined string
channelCoverImg	频道图标图片地址	string
splashImg	引导页封面图	undefined string
mobileSplashLargeImg	移动端引导页大图	undefined string
pcWatchBackgroundImage	PC 端观看页背景图	undefined string
mobileChatBackgroundImage	移动端聊天室背景图	undefined string
mobileChatBackgroundImageAmbiguity	聊天室背景图模糊，0 ~ 100	undefined number
portraitBackgroundImage	竖屏背景图	undefined string
portraitBackgroundImageAmbiguity	竖屏背景图透明度，0 ~ 100	undefined number

示例：

```
const setting = watchCore.channel.getThemeSetting();
console.log('皮肤风格：', setting.pageSkin);
console.log('直播间图标：', setting.channelCoverImg);
```

4.15 获取讲师信息

用于获取当前讲师/主播的信息。

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);
```

4.16 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>
<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);
```

4.17 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>

属性名	说明	类型
<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);
```

4.18 获取分享信息

用于获取当前频道的分享信息，包括是否开启分享、封面图、标题

Api 方法： `getShareInfo(): Promise<ShareInfo>`

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

属性名	说明	类型
<code>shareEnabled</code>	是否开启分享功能	<code>boolean</code>
<code>shareImageUrl</code>	分享封面	<code>string</code>
<code>shareTitle</code>	分辨率高度	<code>string</code>

示例：

```
const shareInfo = await watchCore.channel.getShareInfo();
console.log('是否开启分享', shareInfo.shareEnabled);
console.log('分享的封面', shareInfo.shareImageUrl);
console.log('分享的标题', shareInfo.shareTitle);
```

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);
```