

观看条件 - 付费观看

一、设置信息

1.1 付费观看条件设置信息

Interface 接口： `AuthSettingItemPay`

属性名	说明	类型
<code>authType</code>	条件类型	<code>Pay</code>
<code>enabled</code>	是否启用	<code>null YN</code>
<code>payAuthTips</code>	欢迎标题	<code>string</code>
<code>price</code>	观看价格	<code>number</code>
<code>payEntryText</code>	入口文本	<code>null string</code>
<code>trialWatchEnabled</code>	试看开关	<code>YN</code>
<code>trialWatchTime</code>	试看时长，单位：分钟	<code>number</code>

二、使用方式

2.1 获取付费观看的支付信息

用于获取观众进行付费观看的支付信息，如：PC 端的支付二维码、微信 H5 的微信支付签名等，调用后可根据支付 id `payId` 查询观众的付费状态。

Api 方法： `getAuthPayData(): Promise<CommonResult<AuthPayFailReason, AuthPayData>>`

返回值说明： 支付数据， `Promise<CommonResult<AuthPayFailReason, AuthPayData>>` 类型

示例：

```
const { success, ...payData } = await watchCore.auth.getAuthPayData();
if(!success) return
console.log('支付 id', payData.payId);
console.log('微信支付地址', payData.codeUrl); // PC 端下使用
console.log('微信支付二维码图片地址', payData.qrcodeUrl); // PC 端下使用
console.log('微信 H5 支付签名', payData.wxPaySignData); // 微信 H5 下使用
```

2.2 获取付费观看的支付状态

当调用 `getAuthPayData` 后，根据返回的 `payId` 查询支付状态，返回 `true` 时表示支付成功，随后即可进入观看页。

Api 方法： `checkPayStatus(payId?: string): Promise<boolean>`

参数说明：

- `payId`：支付 id，`string` 类型，选传

返回值说明： 是否支付成功，`Promise<boolean>` 类型

示例：

```
// 获取支付数据
const payData = await watchCore.auth.getAuthPayData();
// 检查支付状态
const payStatus = await watchCore.auth.checkPayStatus(payData.payId);
// 支付成功
if (payStatus) {
  toast.success('支付成功');
  handleAuthVerifySuccess({
    success: true,
    authType: AuthType.Pay,
  });
}
```

2.3 获取微信扫码观看信息

已付费过的观众可以通过微信扫码二维码直接进入观看页，无需二次支付，通过该方法获取扫码信息，调用后可根据记录 id `logId` 查询观众的扫码状态。

Api 方法： `getWechatPayCheckData(): Promise<AuthWechatPayCheckData>`

返回值说明： 微信扫码观看信息，`Promise<AuthWechatPayCheckData>` 类型，详细类型说明如下

属性名	说明	类型
logId	记录 id	string
codeUrl	微信二维码内容	string
qrcodeUrl	二维码图片地址	string

示例：

```
const wechatQrcodeData = await watchCore.auth.getWechatPayCheckData();
console.log('记录 id', wechatQrcodeData.logId);
console.log('微信扫描二维码图片地址', wechatQrcodeData.qrcodeUrl);
console.log('二维码内容', wechatQrcodeData.codeUrl);
```

2.4 获取微信扫码观看状态

当调用 `getWechatPayCheckData` 后，根据返回的 `logId` 查询支付状态，返回 `true` 时表示使用已支付的微信扫码，随后即可进入观看页。

Api 方法： `checkWechatPayStatus(logId?: string): Promise<boolean>`

参数说明：

- logId: 记录 id, `string` 类型，选传

返回值说明： 是否使用已支付的微信扫码成功, `Promise<boolean>` 类型

示例：

```
// 获取扫码数据
const wechatQrcodeData = await watchCore.auth.getWechatPayCheckData();
// 检查扫码状态
const status = await watchCore.auth.checkWechatPayStatus(wechatQrcodeData.logId);
// 扫码成功
if (status) {
  toast.success('扫码成功');
  handleAuthVerifySuccess({
    success: true,
    authType: AuthType.Pay,
  });
}
```

三、PC 端付费观看处理

3.1 PC 端微信扫码支付

PC 端页面需要通过扫描微信支付二维码进行支付，开发者可通过 `getAuthPayData` 获取微信支付二维码地址，并调用 `startCheckWechatPayStatusPolling` 轮询微信支付状态。

示例代码：

```
/** 获取微信支付二维码 */
async function getPcPayData() {
  const payData = await watchCore.auth.getAuthPayData();

  // 显示支付二维码
  const imageElem = document.createElement('img');
  imageElem.src = payData.qrcodeUrl;

  // 启动前先结束上一次轮询，避免创建多次轮询
  watchCore.auth.closeCheckPayStatusPolling();

  // 启动轮询检查支付状态
  watchCore.auth.startCheckPayStatusPolling({
    // 如果已调用 getAuthPayData，则可以不传 payId
    payId: payData.payId,
    // 支付成功回调，详细的处理流程可见当前文档 [PC 端支付成功处理]
    successCallback: () => {
      onPaySuccess()
    },
    // 支付超时回调
    timeoutCallback: () => {
      console.log('支付超时! ');
    },
  });
}
```

3.2 PC 端微信扫码观看

已付费过的观众可以通过微信扫描二维码直接进入观看页，无需二次支付，开发者可通过 `getWechatPayCheckData` 获取微信扫码信息，并调用 `startCheckWechatPayStatusPolling` 轮询微信扫码状态。

示例代码：

```

async function getPcWechatPayData() {
  const wechatQrcodeData = await watchCore.auth.getWechatPayCheckData();

  // 显示扫描二维码
  const imageElem = document.createElement('img');
  imageElem.src = wechatQrcodeData.qrcodeUrl;

  // 扫码记录 id
  const logId = wechatQrcodeData.logId;

  watchCore.auth.closeCheckWechatPayStatusPolling();

  // 开启微信扫码状态检查轮询
  watchCore.auth.startCheckWechatPayStatusPolling({
    // 如果已调用 getWechatPayCheckData, 则可以传 logId
    logId: logId,
    // 扫码成功回调, 详细的处理流程可见当前文档 [PC 端支付成功处理]
    successCallback: () => {
      onPaySuccess()
    },
    // 扫码超时回调
    timeoutCallback: () => {
      console.log('支付超时! ');
    },
  });
}

```

如果您不使用 `startCheckWechatPayStatusPolling` 来轮询检测微信扫码, 可调用 `checkWechatPayStatus` 来获取观众的微信扫码登录状态, 示例代码:

```

async function example() {
  const data = await watchCore.auth.getWechatPayCheckData();
  const status = await watchCore.auth.checkWechatPayStatus(data.logId);
  if (status) {
    console.log('扫码成功');
  } else {
    console.log('未扫码登录');
  }
}

```

3.3 PC 端支付成功处理

由于 PC 端微信打开观看页时需要进行微信授权处理, 因此在 PC 端支付成功后需要判断当前环境与微信授权状态, 在微信端且未授权时进行微信非静默授权, 示例代码:

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

function onPaySuccess() {
  // 针对于 PC 端微信客户端，如果需要显示微信用户信息，需要在付费完后做微信非静默授权
  if (isWeixin && !watchCore.weixin.isWeixinWatchAuthorized()) {
    // TODO: 执行微信非静默授权
    return;
  }

  handleAuthVerifySuccess({
    success: true,
    authType: AuthType.Pay,
  });
}
```

四、微信 H5 付费观看处理

4.1 微信 H5 微信支付观看

在微信 H5 环境下，可通过 `getAuthPayData` 获取微信 JSSDK 支付 API 的签名参数，当观众支付成功后，调用 `checkPayStatus` 检查微信支付状态，当返回成功后即可进入直播观看页，代码示例如下：

```

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

async function toDoAuthPayMobile() {
  // 非微信打开
  if (!isWeixin) {
    alert('请在微信中打开本页进行支付');
    return;
  }

  const payData = await watchCore.auth.getAuthPayData();
  if (payData.wxPaySignData) {
    // 调用微信 JSSDK 支付 API, 唤起支付窗口
    wx.chooseWXPay({
      timestamp: payData.timestamp,
      nonceStr: payData.nonceStr,
      package: payData.package,
      signType: payData.signType,
      paySign: payData.paySign,
      success: () => {
        // 支付成功, 详细的处理流程可见文档 4.2
        onPaySuccess();
      },
      cancel: () => {
        alert('您取消了支付');
      },
    });
  }
}

```

4.2 微信 H5 支付成功处理

在微信 H5 支付成功后, 需要观众进行微信非静默授权, 示例代码如下:

```

async function onPaySuccess() {
  // 如果未进行非静默授权, 则先进行一次非静默授权
  if (!watchCore.weixin.isWeixinWatchAuthorized()) {
    // TODO: 执行微信非静默授权
    return;
  }

  // 是否已支付成功
  const payed = await watchCore.auth.checkPayStatus();
  if (payed) {
    handleAuthVerifySuccess({
      success: true,
      authType: AuthType.Pay,
    });
  }
}

```