

观看条件 - 独立授权

一、设置信息

1.1 独立授权设置信息

关于独立授权设置及服务端的处理流程可见帮助中心文档：[独立授权](#)

Interface 接口： `AuthSettingItemDirect`

属性名	说明	类型
<code>authType</code>	条件类型	<code>Direct</code>
<code>enabled</code>	是否启用	<code>null</code> <code>YN</code>
<code>directRedirectUri</code>	授权失败重定向 url	<code>string</code>

二、使用方式

2.1 允许验证独立授权

从授权平台跳转回观看页后，通过该方法判断当前环境下是否允许进行独立授权签名验证。

从 v0.11.0 版本开始，在授权通过，但其他配置是正常的情况下，也允许进行外部授权签名验证。另外通过新增 options 参数，开发者可以设置 `{ ignoreAuthorized:false }` 来达到和之前版本一样的效果

Api 方法： `allowToVerifyDirectAuth(signParams: DirectAuthSignParams, options?: Object): Promise<boolean>`

参数说明：

- signParams：授权签名参数， `DirectAuthSignParams` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
userid	用户 id	string	是	-
ts	时间戳	string	是	-
sign	授权签名	string	是	-
nickname	用户昵称	string	是	-

- options: 配置项, 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
ignoreAuthorized	忽略授权情况, 默认 true	boolean	否	-

返回值说明: 是否允许验证独立授权, Promise<boolean> 类型

示例:

```
import { parse } from '@polyv/utils/querystring';
import { DirectAuthSignParams } from '@polyv/live-watch-sdk';

async function example {
  const queryParams = parse(window.location.search.slice(1));
  const signParams: DirectAuthSignParams = {
    userid: queryParams.userid || '',
    ts: queryParams.ts || '',
    sign: queryParams.sign || '',
    nickname: queryParams.nickname || '',
  };

  const allowVerify = watchCore.auth.allowToVerifyDirectAuth(signParams);
  if (allowVerify) {
    // TODO 验证独立授权
  }
}
```

2.2 验证独立授权签名参数

从授权平台跳转回观看页后, 通过 verifyDirectAuth 进行外部授权签名参数验证, 在调用前请调用 allowToVerifyDirectAuth 判断签名参数是否符合要求。

Api 方法： `verifyDirectAuth(signParams: DirectAuthSignParams, queryParams: object): Promise<VerifyDirectAuthResult>`

参数说明：

- `signParams`：授权参数， `DirectAuthSignParams` 类型， 必传， 详细类型说明如下

参数名	说明	类型	必须	默认值
<code>userid</code>	用户 id	<code>string</code>	是	-
<code>ts</code>	时间戳	<code>string</code>	是	-
<code>sign</code>	授权签名	<code>string</code>	是	-
<code>nickname</code>	用户昵称	<code>string</code>	是	-

- `queryParams`：链接参数， `object` 类型， 必传

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

示例：

```
import { parse } from '@polyv/utils/querystring';
import { ExternalAuthSignParams } from '@polyv/live-watch-sdk';

async function example {
  const queryParams = parse(window.location.search.slice(1));
  const signParams: ExternalAuthSignParams = {
    userid: queryParams.userid || '',
    ts: queryParams.ts || '',
    sign: queryParams.sign || '',
    nickname: queryParams.nickname || '',
  };

  const allowVerify = watchCore.auth.allowToVerifyDirectAuth(signParams);
  if (!allowVerify) {
    return;
  }

  const result = await watchCore.auth.verifyDirectAuth(signParams, queryParams);
  if (result.success) {
    handleAuthVerifySuccess(result);
  } else {
    handleAuthVerifyFail(result);
  }
}
```

2.3 允许自动重定向到独立授权失败页面

当独立授权失败或没有观看页地址没有独立授权签名参数时，通过 `allowAutoRedirectDirectAuthFailUrl` 方法判断是否自动跳转到独立授权失败页面，内部判断条件如下：

- 观众未授权
- 管理后台只设置了独立授权

Api 方法： `allowAutoRedirectDirectAuthFailUrl(): Promise<boolean>`

返回值说明： 是否自动重定向， `Promise<boolean>` 类型

示例：

```
// 验证独立授权失败之后 / 无独立授权签名参数
const allowAutoRedirect = await watchCore.auth.allowAutoRedirectDirectAuthFailUrl();
if (allowAutoRedirect) {
  watchCore.auth.redirectDirectAuthFailUrl();
}
```

2.4 获取独立授权失败的地址

Api 方法： `getDirectAuthFailUrl(): Promise<DirectAuthFailUrlData>`

从 v2.5.0

当独立授权失败或没有观看页地址没有独立授权签名参数时，通过该方法获取授权失败自定义 URL 并进行跳转 版本开始支持

返回值说明： 独立授权失败地址信息， `Promise<DirectAuthFailUrlData>` 类型，详细类型说明如下

属性名	说明	类型
<code>directAuthFailUrl</code>	独立授权失败跳转地址	<code>undefined string</code>

示例：

```
document.querySelector('button').addEventListener('click', async () => {
  const data = await watchCore.auth.getDirectAuthFailUrl();
  window.location.href = data.directAuthFailUrl;
});
```

2.5 重定向到独立授权失败地址

Api 方法： `redirectDirectAuthFailUrl(): Promise<boolean>`

从 v2.5.0 如果不调用 `getDirectAuthFailUrl` 获取重定向地址，开发者可以调用 `redirectDirectAuthFailUrl` 进行跳转，通过方法返回的结果判断页面是否被跳转。版本开始支持

返回值说明： 是否重定向， `Promise<boolean>` 类型

示例：

```
const result = await watchCore.auth.redirectDirectAuthFailUrl();
if (result) {
  console.log('页面已重定向');
} else {
  console.log('重定向失败');
}
```