

观看条件 - 自定义授权

一、设置信息

1.1 自定义授权设置信息

关于自定义授权设置及服务端的处理流程可见帮助中心文档：[自定义授权](#)

Interface 接口： `AuthSettingItemCustom`

属性名	说明	类型
<code>authType</code>	条件类型	<code>Custom</code>
<code>enabled</code>	是否启用	<code>null</code> <code>YN</code>
<code>customUri</code>	自定义 url	<code>string</code>
<code>customEntryText</code>	入口文本	<code>null</code> <code>string</code>

二、使用方式

2.1 允许验证自定义授权

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自定义授权签名验证。

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： `allowToVerifyCustomAuth(signParams: CustomAuthSignParams, options?: Object): Promise<boolean>`

参数说明：

- `signParams`：授权签名参数， `CustomAuthSignParams` 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	-

- 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: 是否允许验证自定义授权, Promise<boolean> 类型

示例:

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

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

  const allowVerify = watchCore.auth.allowToVerifyCustomAuth(signParams);
  if (allowVerify) {
    // TODO 验证自定义授权
  }
}
```

2.2 验证自定义授权签名参数

从授权平台跳转回观看页后, 通过 verifyCustomAuth 进行自定义授权参数验证, 在调用前请调用 allowToVerifyCustomAuth 判断签名参数是否符合要求。

Api 方法: verifyCustomAuth(signParams: CustomAuthSignParams, queryParams: object): Promise<VerifyCustomAuthResult>

参数说明:

- signParams: 授权参数, CustomAuthSignParams 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	-

- queryParams: 链接参数, object 类型, 必传

返回值说明: 授权结果, Promise<VerifyCustomAuthResult> 类型

示例:

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

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

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

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

2.3 允许自动重定向到自定义授权地址

授权模块提供方法用于在观众进入观看页, 直接重定向到自定义授权地址, 无需经过引导页, 内部判断条件如下:

- 观众未授权
- 管理后台只设置了自定义授权

- 管理后台关闭引导页开关

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

返回值说明： 是否自动重定向， `Promise<boolean>` 类型

示例：

```
const allowAutoRedirect = await watchCore.auth.allowAutoRedirectCustomAuthUrl();
// 当前不需要显示引导页，直接跳到自定义授权地址
if (allowAutoRedirect) {
  watchCore.auth.redirectCustomAuthUrl();
}
```

2.4 获取自定义授权的地址

当用户点击授权按钮时，通过该方法获取自定义授权链接并进行跳转。

Api 方法： `getCustomAuthUrl(): Promise<CustomAuthUrlData>`

返回值说明： 自定义授权跳转地址信息， `Promise<CustomAuthUrlData>` 类型，详细类型说明如下

属性名	说明	类型
<code>customAuthUrl</code>	自定义授权跳转地址	<code>string</code>

示例：

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

2.5 重定向到自定义授权地址

如果不调用 `getCustomAuthUrl` 获取授权地址，开发者可以调用 `redirectCustomAuthUrl` 进行跳转，通过方法返回的结果判断页面是否被跳转。

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

返回值说明： 是否重定向， `Promise<boolean>` 类型

示例：

```
const result = await watchCore.auth.redirectCustomAuthUrl();
if (result.success) {
  console.log('页面已重定向');
} else {
  console.log('重定向失败, 原因: ', result.failReason);
}
```