

单点登录后台

描述

单点登录可以实现在登录已有系统的状态下，免登录操作进入保利威的账号、保利威的直播子账号、频道、助教，做直播相关设置

适用场景

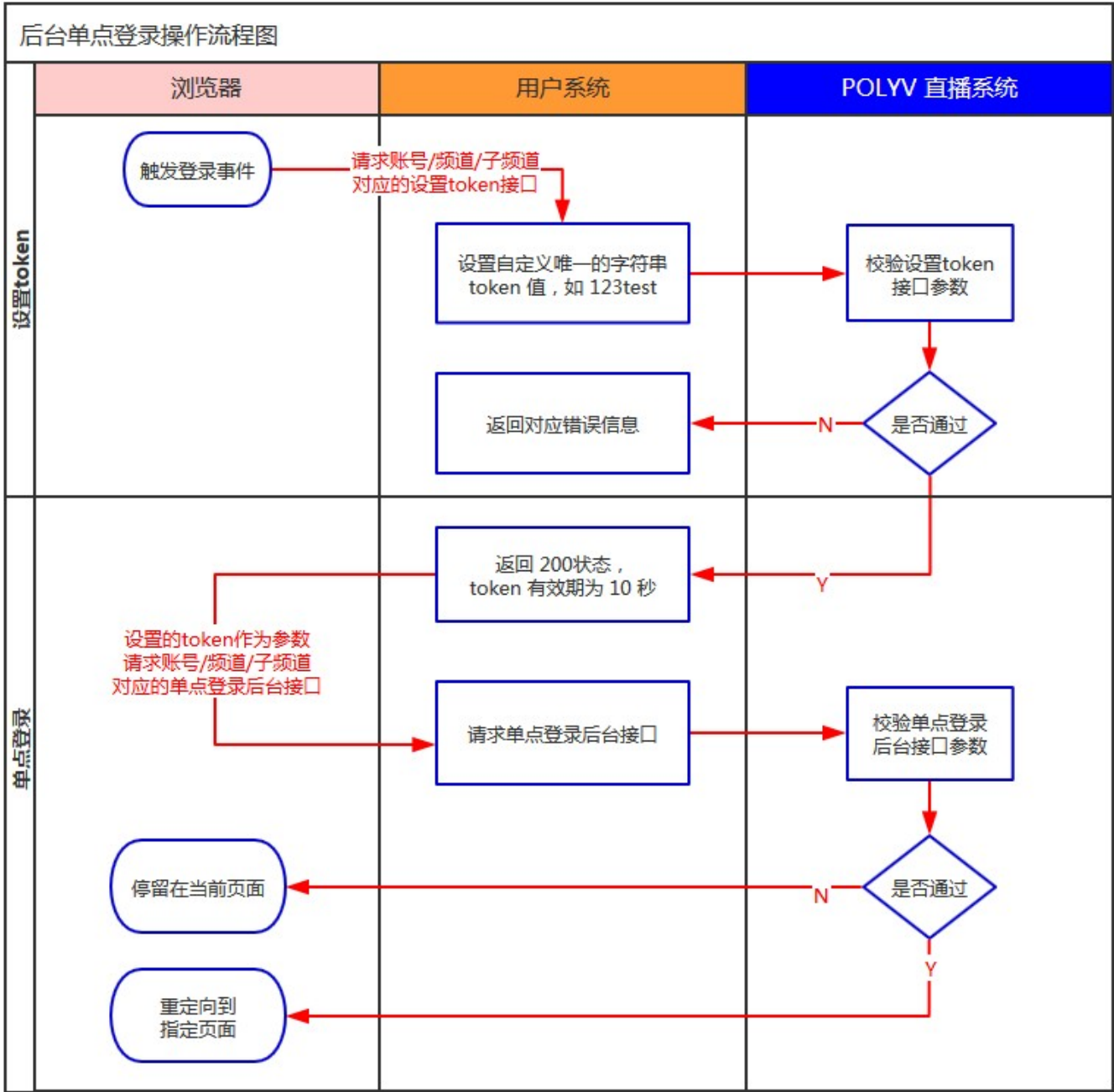
保利威后台单点登录分：账号、直播子账号、频道、助教四种，其中账号和子账号的对接方式一致。

- 账号：进入账号后台，获取最高的管理员权限，管理所有频道、子频道，适合机构管理员的登录跳转
- 直播子账号：进入账号后台，根据后台配置的对应账号角色权限，拥有不同操作后台功能权限，适合机构不同角色员工的登录跳转
- 频道：进入频道后台，仅获取当前频道的管理员权限，可管理当前频道及频道下的子频道，适合讲师的登录跳转
- 助教：仅可进入助教页面，协助讲师在直播中管理聊天室、发起互动功能等，适合助教、助理的登录跳转
- 嘉宾：仅可进入嘉宾连麦页面，协助讲师解答问题、完成直播等
- 讲师：仅可进入直播页面，直接进入直播界面开始直播、发起互动功能等，适合讲师、主持人的登录跳转

温馨提示

- 1、区分新版后台、旧版后台请[点这里](#)。
- 2、单点登录后跳转的地址redirect参数区分新版后台和旧版后台，请按照实际情况跳转。
- 3、新版直播后台默认域名为 <https://console.polyv.net>，旧版直播后台默认域名为 <https://live.polyv.net>，如果您做了CNAME定制，可以将redirect参数中的默认域名替换为CNAME设置的域名。
- 4、单点登录在线demo访问请[点这里](#)。

流程图



账号单点登录

1、设置单点登录token

设置账号单点登录token

2、账号授权登录

账号级授权地址为 <https://console.polyv.net/v2/sso/userLogin.do> ， 拼接后的地址为 <https://console.polyv.net/v2/sso/userLogin.do?userId={userId}&token={token}&redirect={redirect}> ， 您

在浏览器中打开授权地址即可，请求参数说明如下：

参数说明

参数名	必选	类型	说明
userId	true	String	账号id
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	false	String	完成授权后重定向地址，使用url编码，取值可参照【账号级redirect取值】默认为直播列表页

账号级redirect取值（url编码前）

版本	取值	说明
新版后台	https://console.polyv.net/live/index.html#/channel	直播列表页，支持参数hideTop、hideLeft隐藏顶部和左侧菜单项，如： https://console.polyv.net/live/index.html#/channel?hideTop=true&hideLeft=true
旧版后台	https://live.polyv.net/#/channel	直播列表页
旧版后台	https://live.polyv.net/#/develop/appld	直播后台开发设置

请求示例

```
https://console.polyv.net/v2/sso/userLogin.do?
userId=1b448be323&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Flive%2Findex.html
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的HttpUtil.java 和 LiveSignUtil.java 都包含在下载文件中。

```

//账号单点登录
@Test
public void testAccountSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    String url = "https://api.polyv.net/live/v3/user/set-sso-token";

    //1、设置频道单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成账号授权登录地址
    String redirectUrl = "https://console.polyv.net/live/index.html";
    String authURL = "https://console.polyv.net/v2/sso/userLogin.do";
    authURL +=
        "?userId=" + userId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("账号单点登录设置成功, 跳转地址为: {}", authURL);
}

```

子账号单点登录

1、设置单点登录token

设置子账号单点登录token 其中请求参数childEmail为子账号邮箱，如：wcc@polyv.net

2、子账号授权登录

子账号级授权地址为 <https://console.polyv.net/v2/sso/userLogin.do>，拼接后的地址为
<https://console.polyv.net/v2/sso/userLogin.do?userId={userId}&token={token}&redirect={redirect}&childEmail={childEmail}>，您在浏览器中打开授权地址即可，请求参数说明如下：

参数说明

参数名	必选	类型	说明
userId	true	String	账号id

参数名	必选	类型	说明
token	true	String	第一步设置的token值，10秒内且一次验证有效
childEmail	true	String	直播子账号的邮箱账号
redirect	false	String	完成授权后重定向地址，使用url编码，取值可参照【子账号级redirect取值】默认为直播列表页

子账号级redirect取值（url编码前）

版本	取值	说明
新版后台	https://console.polyv.net/live/index.html	直播列表页
旧版后台	https://live.polyv.net/#/channel	直播列表页
旧版后台	https://live.polyv.net/#/develop/appld	直播后台开发设置

请求示例

```
https://console.polyv.net/v2/sso/userLogin.do?
userId=1b448be323&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Flive%2Findex.html&childEmail=wcc@polyv.net
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的HttpUtil.java 和 LiveSignUtil.java 都包含在下载文件中。

```

//子账号单点登录
@Test
public void testChildAccountSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    String url = "https://api.polyv.net/live/v3/user/set-sso-token";
    //子账号邮箱
    String childEmail = "wzk@polyv.net";

    //1、设置频道单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("childEmail", childEmail);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成子账号授权登录地址
    String redirectUrl = "https://console.polyv.net/live/index.html";
    String authURL = "https://console.polyv.net/v2/sso/userLogin.do";
    authURL += "?userId=" + userId + "&childEmail=" + childEmail + "&token=" + token + "&redirect=" +
        URLEncoder.encode(redirectUrl, "utf-8");
    log.info("子账号单点登录设置成功, 跳转地址为: {}", authURL);
}

```

频道单点登录

1、设置单点登录token

设置频道单点登录token

2、频道授权登录

频道级授权地址为 <https://console.polyv.net/teacher/auth-login> , 拼接后的地址为
<https://console.polyv.net/teacher/auth-login?channelId={channelId}&token={token}&redirect={redirect}>
 , 您在浏览器中打开授权地址即可, 请求参数说明如下:

参数说明

参数名	必选	类型	说明
channelId	true	String	频道号
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	false	String	完成授权后重定向地址，使用url编码，取值可参照【频道级redirect取值】默认为直播间信息页
failRedirect	false	String	授权失败后重定向地址，使用url编码。

频道级redirect取值（url编码前）

版本	取值	说明
新版后台	https://console.polyv.net/live/#/teacher/{channelId}/base-info/channel-info	直播间信息页，其中{channelId}需替换为对应频道号
新版后台	https://console.polyv.net/live/#/teacher/{channelId}/channel-statistics	直播间统计页，其中{channelId}需替换为对应频道号
旧版后台	https://live.polyv.net/#/teacher/{channelId}/monitoring	直播间监控页，其中{channelId}需替换为对应频道号
旧版后台	https://live.polyv.net/#/teacher/{channelId}/room/detail	直播间信息页，其中{channelId}需替换为对应频道号
旧版后台	https://live.polyv.net/#/teacher/{channelId}/statistic	直播间统计页，其中{channelId}需替换为对应频道号

请求示例

```
https://console.polyv.net/teacher/auth-login?channelId=1965681&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Flive%2F%23%2Fteacher%2F1965681%2Fbase-info%2Fchannel-info
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的[HttpUtil.java](#) 和 [LiveSignUtil.java](#) 都包含在下载文件中。

```
//频道单点登录
@Test
public void testChannelSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    //频道号
    String channelId = "1965681";
    String url = String.format("http://api.polyv.net/live/v2/channels/%s/set-token", channelId);

    //1、设置频道单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成频道授权登录地址
    String redirectUrl = String.format("https://console.polyv.net/live/#/teacher/%s/base-info/channel-info",
        channelId);
    String authURL = "https://console.polyv.net/teacher/auth-login";
    authURL +=
        "?channelId=" + channelId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("频道单点登录设置成功, 跳转地址为: {}", authURL);
}
```

助教单点登录

1、设置单点登录token

[设置子频道单点登录token](#) 其中请求参数accountId为助教账号id, 如: 0041965681

2、助教授权登录

助教级授权地址为 <https://console.polyv.net/teacher/auth-login> , 拼接后的地址为
<https://console.polyv.net/teacher/auth-login?channelId={channelId}&token={token}&redirect={redirect}>
，您在浏览器中打开授权地址即可，请求参数说明如下：

参数说明

参数名	必选	类型	说明
channelId	true	String	助教账号，如： 0041965681，不能去除前面的00
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	false	String	完成授权后重定向地址，使用url编码，取值可参照【助教级redirect取值】默认为旧版助教页
failRedirect	false	String	授权失败后重定向地址，使用url编码。

助教级redirect取值（url编码前）

版本	取值	说明
新版后台	https://console.polyv.net/assistant/?accountId={channelId}	新版助教页，其中{channelId}需替换为对应助教账号，如：0041965681
旧版后台	https://live.polyv.net/assistant.html	旧版助教页，如果开通新版助教页，则会自动跳转至新版

请求示例

```
https://console.polyv.net/teacher/auth-login?channelId=0041965681&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Fassistant%2F%3FaccountId%3D0041965681
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#) , 下载后加入到自己的源码工程中即可。测试用例中的HttpUtil.java 和 LiveSignUtil.java 都包含在下载文件中。

```

//助教单点登录
@Test
public void testAssistantSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //自定义的token,只能使用一次,且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    //助教账号
    String accountId = "0041965681";
    String url = String.format("http://api.polyv.net/live/v2/channels/%s/set-account-token", accountId);

    //1、设置助教单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成助教授权登录地址
    String redirectUrl = "https://console.polyv.net/assistant/?accountId=" + accountId;
    String authURL = "https://console.polyv.net/teacher/auth-login";
    authURL +=
        "?channelId=" + accountId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("助教单点登录设置成功,跳转地址为: {}", authURL);
}

```

嘉宾单点登录

1、设置单点登录token

设置子频道单点登录token 其中请求参数accountId为嘉宾账号id, 如: 0021965681

2、嘉宾授权登录

嘉宾级授权地址为 <https://console.polyv.net/teacher/auth-login> , 拼接后的地址为
<https://console.polyv.net/teacher/auth-login?channelId={channelId}&token={token}&redirect={redirect}>
 , 您在浏览器中打开授权地址即可, 请求参数说明如下:

参数说明

参数名	必选	类型	说明
channelId	true	String	嘉宾账号，如： 0021965681，不能去除前面的00
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	true	String	完成授权后重定向地址，使用url编码，取值可参照【 嘉宾级redirect取值 】
failRedirect	false	String	授权失败后重定向地址，使用url编码。

嘉宾级redirect取值（url编码前）

取值	说明
https://console.polyv.net/web-start?channelId={channelId}	嘉宾页，其中{channelId}需替换为对应嘉宾账号，如：0021965681

请求示例

```
https://console.polyv.net/teacher/auth-login?channelId=0021965681&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Fweb-start%2F%3FchannelId%3D0021965681
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的HttpUtil.java 和 LiveSignUtil.java 都包含在下载文件中。

```

//嘉宾单点登录
@Test
public void testGuestSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    //嘉宾账号
    String accountId = "0021965681";
    String url = String.format("http://api.polyv.net/live/v2/channels/%s/set-account-token", accountId);

    //1、设置嘉宾单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成嘉宾授权登录地址
    String redirectUrl = "https://console.polyv.net/web-start/?channelId=" + accountId;
    String authURL = "https://console.polyv.net/teacher/auth-login";
    authURL +=
        "?channelId=" + accountId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("嘉宾单点登录设置成功, 跳转地址为: {}", authURL);
}

```

讲师单点登录

1、设置单点登录token

设置频道单点登录token

2、讲师授权登录

讲师级授权地址为 <https://console.polyv.net/teacher/auth-login> , 拼接后的地址为

<https://console.polyv.net/teacher/auth-login?channelId={channelId}&token={token}&redirect={redirect}>

, 您在浏览器中打开授权地址即可, 请求参数说明如下:

参数说明

参数名	必选	类型	说明
channelId	true	String	频道号，如：1965681
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	true	String	完成授权后重定向地址，使用url编码，取值可参照【讲师级redirect取值】
failRedirect	false	String	授权失败后重定向地址，使用url编码。【注：如果登陆时频道已有讲师在开播，也会触发授权失败重定向】

讲师级redirect取值（url编码前）

取值	说明
https://console.polyv.net/web-start?channelId={channelId}	讲师开播页，其中{channelId}需替换为对应频道号，如：1965681

请求示例

```
https://console.polyv.net/teacher/auth-login?channelId=1965681&token=yourToken&redirect=https%3A%2F%2Fconsole.polyv.net%2Fweb-start%2F%3FchannelId%3D1965681
```

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的HttpUtil.java 和 LiveSignUtil.java 都包含在下载文件中。

```

@Test
public void testTeacherSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //频道号
    String channelId = "1965681";
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    String url = String.format("http://api.polyv.net/live/v2/channels/%s/set-token", channelId);

    //1、设置频道单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成讲师授权登录地址
    String redirectUrl = "https://console.polyv.net/web-start/?channelId=" + channelId;
    String authURL = "https://console.polyv.net/teacher/auth-login";
    authURL +=
        "?channelId=" + channelId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("讲师单点登录地址设置成功, 跳转地址为: {}", authURL);
}

```

MR控制台单点登录

1、设置单点登录token

设置频道单点登录token

2、MR控制台授权登录

授权地址为 <https://console.polyv.net/teacher/mr-auth-login> , 拼接后的地址为
<https://console.polyv.net/teacher/mr-auth-login?channelId={channelId}&token={token}&mrLiveLoginType={mrLiveLoginType}&redirect={redirect}> , 您在浏览器中打开授权地址即可, 请求参数说明如下:

参数说明

参数名	必选	类型	说明
channelId	true	String	频道号，如：3265597
token	true	String	第一步设置的token值，10秒内且一次验证有效
redirect	true	String	完成授权后重定向地址，使用url编码，取值可参照【MR控制台redirect取值】
mrLiveLoginType	false	String	登录模式 cloud：云端模式 client：本地模式 不传值时，若用户只有一种模式权限，则使用已有模式，若用户拥有两种模式权限，默认使用【云端模式】

MR控制台redirect取值（url编码前）

取值	说明
https://live.polyv.net/mr-controller/scene?channelId={channelId}	MR控制台页，其中{channelId}需替换为对应MR频道号，如：3265597

请求示例

```
https://console.polyv.net/teacher/mr-auth-login?channelId=3265597&token=57e46132c74b4e398aeaeccc5340b1ab&redirect=https%3A%2F%2Flive.polyv.net%2Fmr-controller%2Fscene%3FchannelId%3D3265597
```

Java请求示例

```
//MR控制台单点登录
@Test
public void testMrControllerSSO() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //频道号
    String channelId = "3265597";
    //自定义的token, 只能使用一次, 且10秒内有效
    String token = UUID.randomUUID().toString().replaceAll("-", "");
    String url = String.format("http://api.polyv.net/live/v2/channels/%s/set-token", channelId);

    //1、设置频道单点登录token
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("token", token);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    //TODO 判断response返回是否成功

    //2、生成MR控制台授权登录地址
    String redirectUrl = "https://live.polyv.net/mr-controller/scene?channelId=" + channelId;
    String authURL = "https://console.polyv.net/teacher/mr-auth-login";
    authURL +=
        "?channelId=" + channelId + "&token=" + token + "&redirect=" + URLEncoder.encode(redirectUrl, "utf-8");
    log.info("MR控制台单点登录地址设置成功, 跳转地址为: {}", authURL);
}
```

区分新旧版后台

新版后台截图



旧版后台截图

