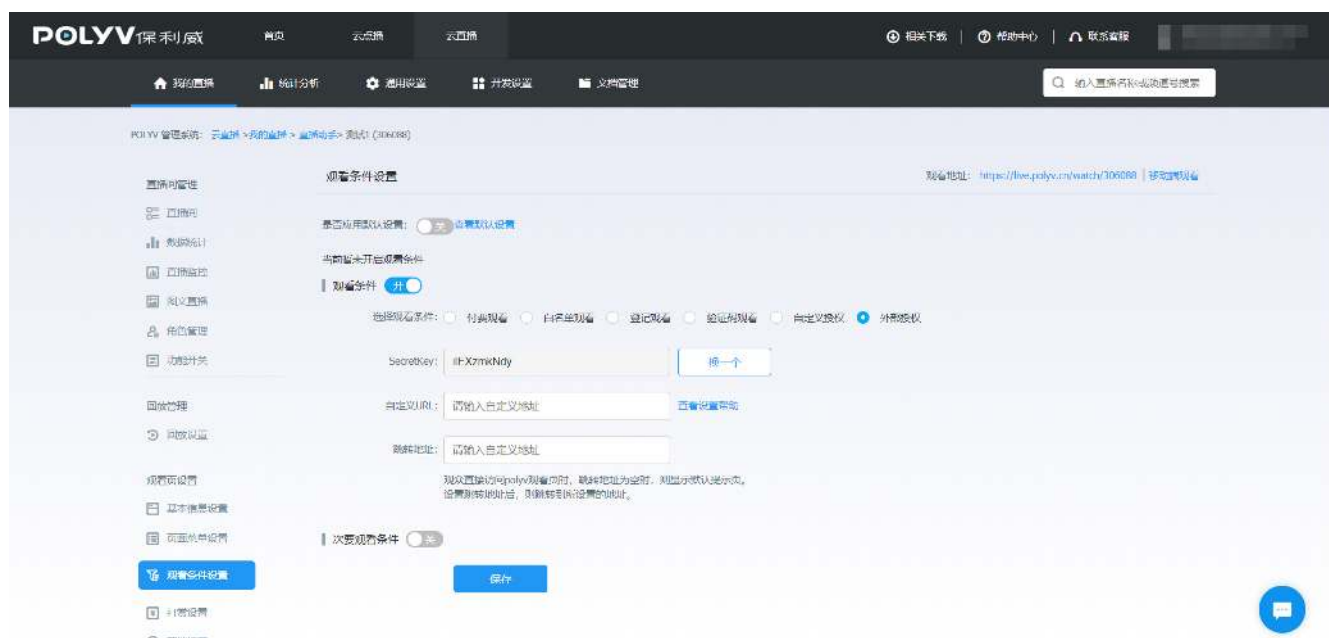


外部授权

功能介绍

- 1、当需要观众登录机构的用户系统，验证通过后才能进入直播观看页时，可使用外部授权。验证观众是否观看由机构实现，验证通过后打开的观看页面由保利威实现。
- 2、机构后台针对通过验证的观众，打开直播观看页时URL需带上指定的参数，经过直播系统验证请求合法后，直播系统会调用客户在后台设置的自定义授权验证接口，进行接口授权验证，两次验证通过，才能进入直播观看页，并且接口返回的观众账号具有唯一性，即同一个账号不能在两个地方同时登录，较早登录的账号会被踢出。



- 1、Secretkey：用于校验签名的生成。
- 2、自定义URL：用于外部授权验证的API接口。
- 3、跳转地址：观众直接访问Polyv观看页，会跳转到该地址；若跳转地址为空，则显示默认提示页。

展示效果

<https://demo.ipolyv.cn/chenwb/open.php>

外部授权流程详解

1、在自定义URL处填写用户的授权验证API接口，需要完整的不带参数的url地址（不能是localhost等本地服务器地址，且不能带？号），如：<http://myWebsite.com/auth>

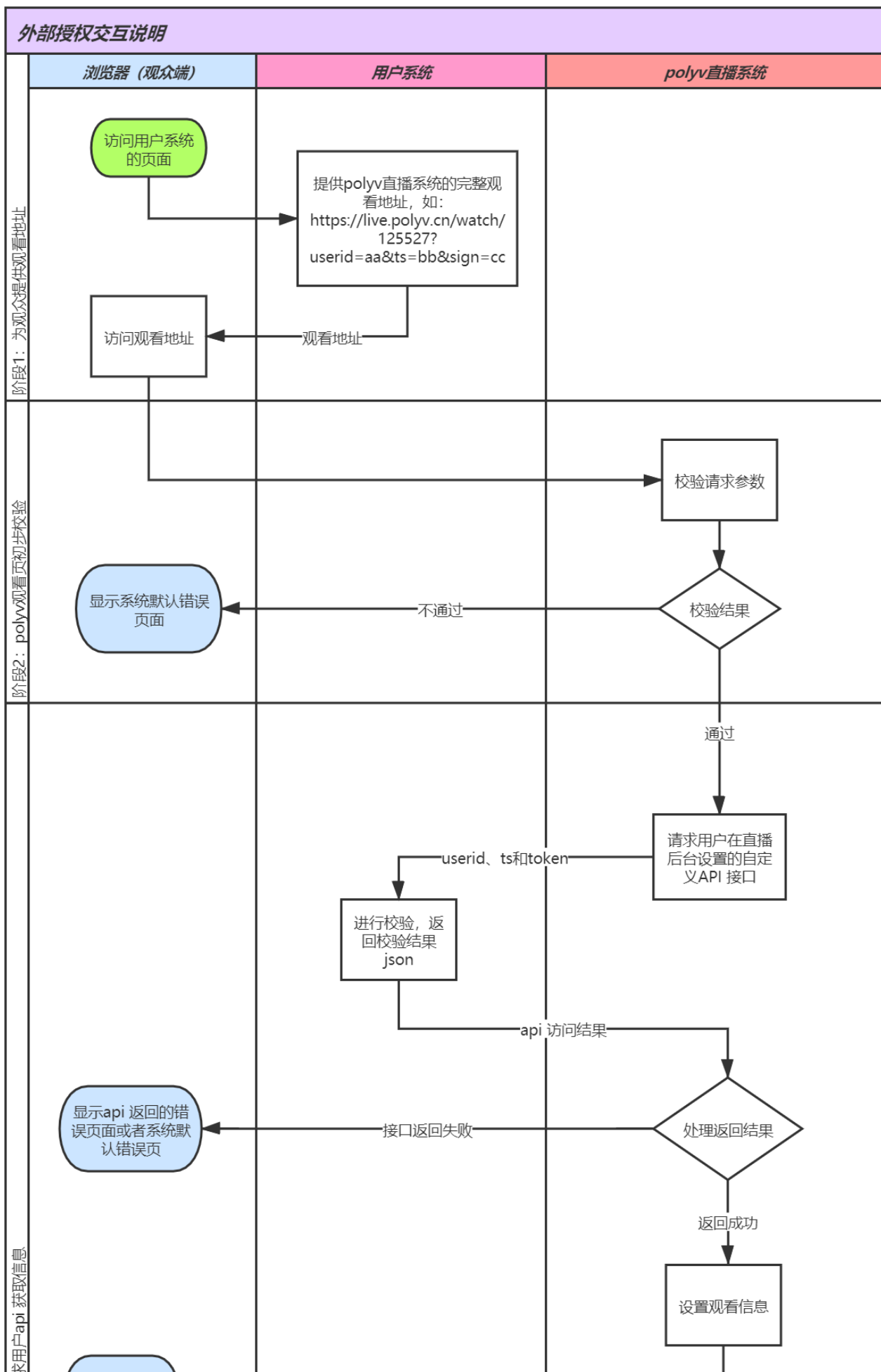
2、在请求保利威视直播观看页时需带上userid（用户ID，仅支持英文大小写、数字和下划线）、ts（当前时间的毫秒级时间戳）和sign（用于校验的签名,生成规则是secretkey + userid + secretkey + ts进行MD5加密），如<https://live.polyv.cn/watch/125527?userid=6b3a43&ts=1498547407000&sign=dd9dc9e42ad7c0204398e925a4ee0f46>

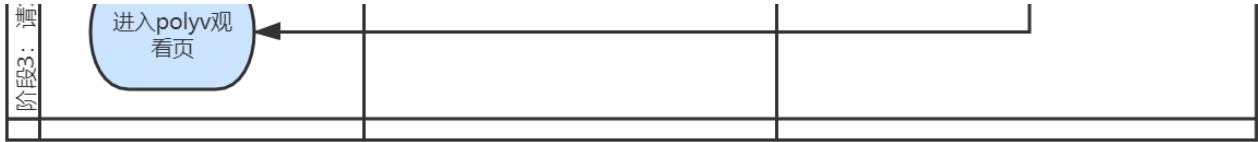
3、直播系统会对字符串secretkey+userid+secretkey+ts进行MD5加密后与用户提交的sign参数的值做比较判断是否合法。一次成功请求后，该链接将失效（sign只能成功使用一次）。如果合法，直播系统将调用用户的api接口，并把userid（用户ID）、ts（当前时间的毫秒级时间戳）、channelId（频道号）和token（用于校验的签名）四个参数通过GET请求传给用户。如果不合法，则给出错误提示。

4、用户API接口获取userid、ts和token参数后，进行签名验证。如果验证通过，则将学员相关信息和可选的观看权限watchAccess（详参“用户系统返回观众信息响应参数描述”）返回给直播系统。

5、直播系统接收用户API接口返回的数据，如果验证成功，则进入到保利威视直播观看页，聊天区将显示学员的昵称和头像。watchAccess为trial时，直播系统将结合频道的外部授权试看配置判断是否进入试看；如果验证失败，则给出错误提示。

交互图如下





参数描述

请求观看页所带参数

用户将以下的参数提交给直播的观看页，（例如：<https://live.polyv.cn/watch/125527?userid=test&ts=1565948760108&sign=b0b6eb22b6fa5e5684873052c27a6cef>）

直播系统会对sign 进行验证，判断是否合法，一次成功请求后，该链接将失效(sign只能成功使用一次)

参数名	必选	说明
userid	true	观众id，重复id时先登录的观众会被踢出直播间，【仅支持英文大小写、数字和下划线，长度最大64位字符，超出64位的部分将被截取不做记录】
ts	true	13位毫秒级时间戳
sign	true	用户校验的签名（字母小写），加密规则：secretkey + userid + secretkey + ts
vid	false	回放视频id，如需回放某个回放视频，则需要传该参数，如： e07738ddd6 该值可使用接口【 查询视频库列表 】返回的videoid

观看页请求观众信息接口参数

观看页请求用户在后台设置的自定义URL获取观众信息，以下为请求所带的参数

参数名	必填	参数说明
userid	true	观众id，重复id时先登录的观众会被踢出直播间，【仅支持英文大小写、数字和下划线，长度最大64位字符】
channelId	true	频道号

参数名	必填	参数说明
ts	true	当前时间的13位毫秒级时间戳
token	true	用于校验的签名,生成的规则:对字符串secretkey + userid + secretkey + ts进行MD5加密生成的字符串（字母小写）

用户系统返回观众信息响应参数描述

字段	类型	必填	字段说明
status	int	true	请求结果，1表示成功，0表示失败
userid	string	true	观众id，【仅支持英文大小写、数字和下划线，长度最大64位字符】
nickname	string	true	观众昵称
marqueeName	string	false	自定义跑马灯字段，该字段会通过【 URL自定义跑马灯 】中code参数回调
avatar	string	true	观众头像地址，头像尺寸200*200，体积不超过30KB。
actor	string	false	观众头衔
actorFColor	string	false	观众头衔字体颜色，非必须，请使用CSS Hex值并且带#号
actorBgColor	string	false	观众头衔背景颜色，非必须，请使用CSS Hex值带#号
param4	string	false	统计观众观看日志的自定义参数
param5	string	false	统计观众观看日志的自定义参数

字段	类型	必填	字段说明
errorUrl	string	false	请求失败时观看页跳转的地址（会带上channelId和userid）
userTags	string数组	false	观众业务标签，用于问卷定向弹出匹配（仅外部授权场景生效）。支持传入一个或多个标签；客户创建问卷配置匹配条件后，携带任一匹配标签的观众进入直播间将弹出对应问卷，未携带或为空时不弹出
watchAccess	string	false	外部授权观看权限。 allow：正式观看；trial：结合频道的外部授权试看配置判断是否进入试看。 未传、空值或其他值按allow处理。该字段仅在status为1时生效

响应示例

成功示例

```
{
  "status":1,
  "userid":"2qwerty",
  "nickname":"testNick",
  "actor":"paul",
  "actorFColor":"#123123",
  "actorBgColor":"#FFFFFF",
  "param4":"param4test",
  "userTags":["vip","registered"],
  "watchAccess":"trial",
  "avatar":"http://live.polyv.net/assets/images/avatars/9avatar.jpg"
}
```

异常示例

```
{
  "status":0,
  "errorUrl":"http://test.com"
}
```

外部授权试看

频道可通过[修改频道信息](#)接口配置外部授权试看开关、试看时长、有效截止时间和试看结束跳转地址，对应字段如下：

字段	说明
extTrialWatchEnabled	外部授权试看开关，Y：开启，N：关闭
extTrialWatchTime	试看时长，单位为分钟
extTrialWatchEndTime	试看有效截止时间，不传表示永久有效
extTrialRedirectUrl	试看耗尽后的跳转地址

以上字段属于频道配置，不需要在客户外部授权接口的响应中返回。客户外部授权接口只需通过watchAccess表达当前观众的观看权限：

watchAccess	处理方式
allow	正式观看，不受外部授权试看时长限制
trial	结合频道外部授权试看配置判断是否进入试看
未传、空值或其他值	按allow处理，保持原有外部授权逻辑

status的优先级高于watchAccess：status不为1时仍按外部授权失败处理；只有status为1时watchAccess才会生效。watchAccess为trial但频道未开启外部授权试看、试看配置已过期或配置不完整时，不进入试看限制。

付费或业务完成后的状态刷新

客户完成付费、报名、资料补充或认证后，应使该观众再次请求客户外部授权接口时返回watchAccess=allow，例如：

```
{
  "status":1,
  "userid":"2qwerty",
  "nickname":"testNick",
  "watchAccess":"allow",
  "avatar":"http://live.polyv.net/assets/images/avatars/9avatar.jpg"
}
```

同一浏览器会话内，当前直播场次已经存在试看记录时，观众返回原观看页并重新获取频道观看信息，直播系统会再次请求客户外部授权接口。该刷新不要求试看时长已经耗尽，也不要求频道当前仍在直播中；客户返回allow后，观众切换为正式观看。返回trial或接口调用失败时，继续保持当前试看状态。

客户外部授权接口应支持重复、幂等调用，并根据userid查询最新业务状态。短时间内的重复刷新可能被合并，客户不应依赖每次观看页请求都产生一次外部授权回调。同一浏览器会话仍然有效时，不需要重新生成带ts、sign的观看地址；会话失效、跨浏览器或更换设备时，仍需重新执行完整外部授权流程。

代码示例（JAVA）

注：LiveSignUtil属于[直播SDK](#)，如不使用直播SDK可使用以下第三点中的“MD5签名方法”。

1、用户系统生成观看链接

```

public static void main(String[] args) {
    //TODO 设置频道号
    String channelId = "2275495";

    //TODO 设置externalKey
    String secret = "";

    // TODO 设置直播观看页地址
    //      如果使用定制域名, 可在“查询频道信息”接口中获取到频道的观看地址 watchUrl
    //      https://help.polyv.net/index.html#/live/api/v4/channel/operate/get_channel_detail
    String url = "https://live.polyv.cn/watch/"+channelId;

    // TODO 根据实际情况设置userid
    String userid = "sadboy";

    String ts = String.valueOf(System.currentTimeMillis());
    String signText = secret+ userid +secret+ts;
    try {
        String sign = LiveSignUtil.md5Hex(signText);
        url += "?userid="+userid+"&ts="+ts+"&sign="+sign;
        log.info(url);
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

```

2、用户服务器校验polyv直播系统回调

```

@Slf4j
@Controller
@RequestMapping(value = "/polyv")
public class PolyvController {

    //TODO 修改secretKey
    private static final String secret = "*****";

    @GetMapping("external")
    @ResponseBody
    public Map<String, Object> external(String channelId,String userid, Long ts, String token) {
        Assert.assertNotBlack(userid);
        Assert.assertNotBlack(token);
        Assert.assertNotNull(ts);
        HashMap<String, Object> map = new HashMap<>();
        long timeMillis = System.currentTimeMillis();
        long diffTime = Math.abs(timeMillis - ts);
        //1、时间戳判断
        if (diffTime > 5 * 60 * 1000) {
            log.error("时间戳验证错误");
            map.put("status", 0);
            //抛出异常时, 如果设置errorUrl, 则会跳转到 errorUrl ,如果未返回 errorUrl, 则先查询 外部授权参数
            // externalRedirectUri, externalRedirectUri不为空则跳转externalRedirectUri地址, externalRedirectUri为空
            则跳转保利威默认页面
            map.put("errorUrl", "https://www.polyv.net");
            return map;
        }
        //2、签名判断
        String signText = secret + userid + secret + ts;
        String sign = null;
        try {
            sign = LiveSignUtil.md5Hex(signText);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        if (sign == null || !sign.equals(token)) {
            log.error("签名验证错误");
            map.put("status", 0);
            //抛出异常时, 如果设置errorUrl, 则会跳转到 errorUrl ,如果未返回 errorUrl, 则先查询 外部授权参数
            // externalRedirectUri, externalRedirectUri不为空则跳转externalRedirectUri地址, externalRedirectUri为空
            则跳转保利威默认页面
            map.put("errorUrl", "https://www.polyv.net");
            return map;
        }
        //TODO 业务逻辑处理, 后续需根据具体需求进行数据库操作
        //3、正常返回
        map.put("status", 1);
        map.put("userid", userid);
        map.put("nickname", "保利威测试用户");
        map.put("marqueeName", "保利威测试跑马灯");
        map.put("actor", "学生");
        map.put("actorFColor", "#2469f3");
        map.put("actorBgColor", null);
    }
}

```

```

        map.put("param4", null);
        map.put("param5", null);
        // 未完成付费或业务流程时返回trial, 完成后返回allow
        map.put("watchAccess", "trial");
        map.put("avatar", "https://help.polyv.net/favicon.ico");
        return map;
    }

}

```

3、MD5签名方法

```

/**
 * 对字符串做MD5加密, 返回加密后的字符串。
 * @param text 待加密的字符串。
 * @return 加密后的字符串。
 * @throws NoSuchAlgorithmException 签名异常
 * @throws UnsupportedEncodingException 编码异常
 */
public static String md5Hex(String text) throws NoSuchAlgorithmException, UnsupportedEncodingException {

    MessageDigest messageDigest = MessageDigest.getInstance("MD5");
    byte[] inputByteArray = text.getBytes(LiveConstant.UTF8);
    messageDigest.update(inputByteArray);
    byte[] resultByteArray = messageDigest.digest();
    return byteArrayToHex(resultByteArray).toLowerCase();

}

/**
 * 将字节数组换成16进制的字符串
 * @param byteArray 字节
 * @return 字符串
 */
public static String byteArrayToHex(byte[] byteArray) {
    // 初始化一个字符数组用来存放每个16进制字符
    char[] hexDigits = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    // new一个字符数组, 这个就是用来组成结果字符串的 (一个byte是八位二进制, 也就是2位十六进制字符 (2的8次方等于16的2
    次方))
    char[] resultCharArray = new char[byteArray.length * 2];
    // 遍历字节数组, 通过位运算 (位运算效率高), 转换成字符放到字符数组中去
    int index = 0;
    for (byte b : byteArray) {
        resultCharArray[index++] = hexDigits[b >> 4 & 0xf];
        resultCharArray[index++] = hexDigits[b & 0xf];
    }
    // 字符数组组合成字符串返回
    return new String(resultCharArray);
}

```

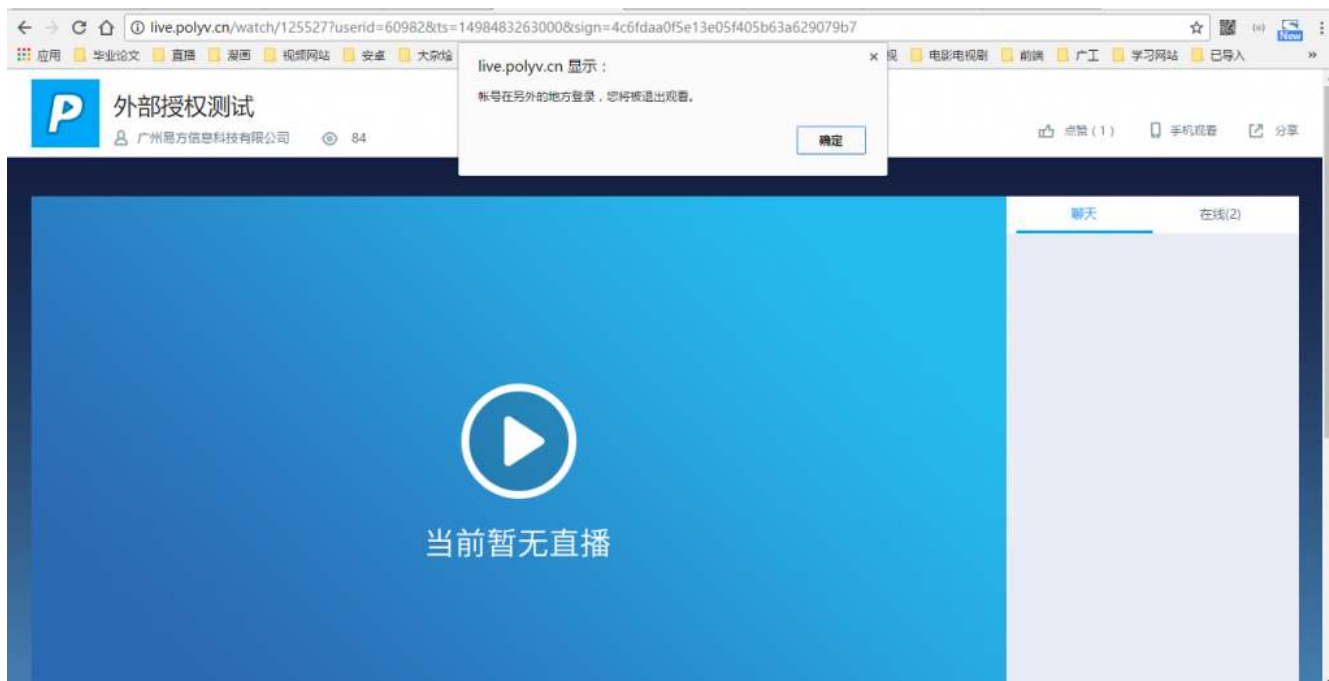
代码示例（PHP）

```
<?php
header("Content-type:application/json;charset=UTF-8"); //媒体格式类型为JSON数据格式
$secretkey = "aDr0t0Cpy8";
$userid    = isset($_GET["userid"]) ? $_GET["userid"] : "";
$ts        = isset($_GET["ts"]) ? $_GET["ts"] : "";
$channelId = isset($_GET["channelId"]) ? $_GET["channelId"] : "";
$token     = isset($_GET["token"]) ? $_GET["token"] : "";
$sign      = md5($secretkey . $userid . $secretkey . $ts);

//用户进行授权验证，返回对应的数据（json格式）
if ($sign == $token) {
    //验证正确
    $array1 = array(
        "status"    => 1, //返回状态
        "userid"    => $userid, //学员唯一标识
        "nickname"  => "保利威", //学员昵称
        "marqueeName" => "polyv", //自定义跑马灯字段
        "avatar"    => "http://live.polyv.net/assets/images/avatars/9avatar.jpg", //学员头像
        "actor"     => "VIP", // 学员头衔，可以不传递
        "actorFColor" => "#5C96E5", // 学员头衔字体颜色，可以不传递
        "actorBgColor" => "#FFFFFF", // 学员头衔背景颜色，可以不传递
        "watchAccess" => "trial" // 未完成付费或业务流程时返回trial，完成后返回allow
    );
    $json1 = json_encode($array1);
    echo $json1;
} else {
    //验证错误
    $array0 = array(
        "status"    => 0,
        "errorUrl"  => "http://xxx.xx.xxxx/error.html", //验证错误跳转的自定义页面
    );
    $json0 = json_encode($array0);
    echo $json0;
}
```

注意事项

1、要保证自定义验证接口返回的userid的唯一性，当多个观众使用同一个userid进入观看页时，较早登录的观众会被后面登录的观众踢出，观看页会提示 "帐号在另外的地方登录,您将被退出观看。如下图：



- 2、自定义验证接口需要填写完整的URL地址，且不能是localhost等本地服务器地址。
- 3、自定义验证接口返回给直播系统的数据格式是json格式。
- 4、同时传入了nickname和marqueeName，在观看日志中，marqueeName将作为用户昵称的统计字段。
- 5、watchAccess仅在status为1时生效。客户接口应支持重复、幂等调用，并根据userid返回最新的观看权限。

错误提示

- 1、user not found：请求自定义验证接口错误，或者接口返回的格式不对。
- 2、invalid sign：签名错误，sign的生成规则是secretkey+userid+secretkey+ts进行MD5加密。
- 3、sign expired：签名过期，每一个sign只能使用一次，使用后需重新生成。