

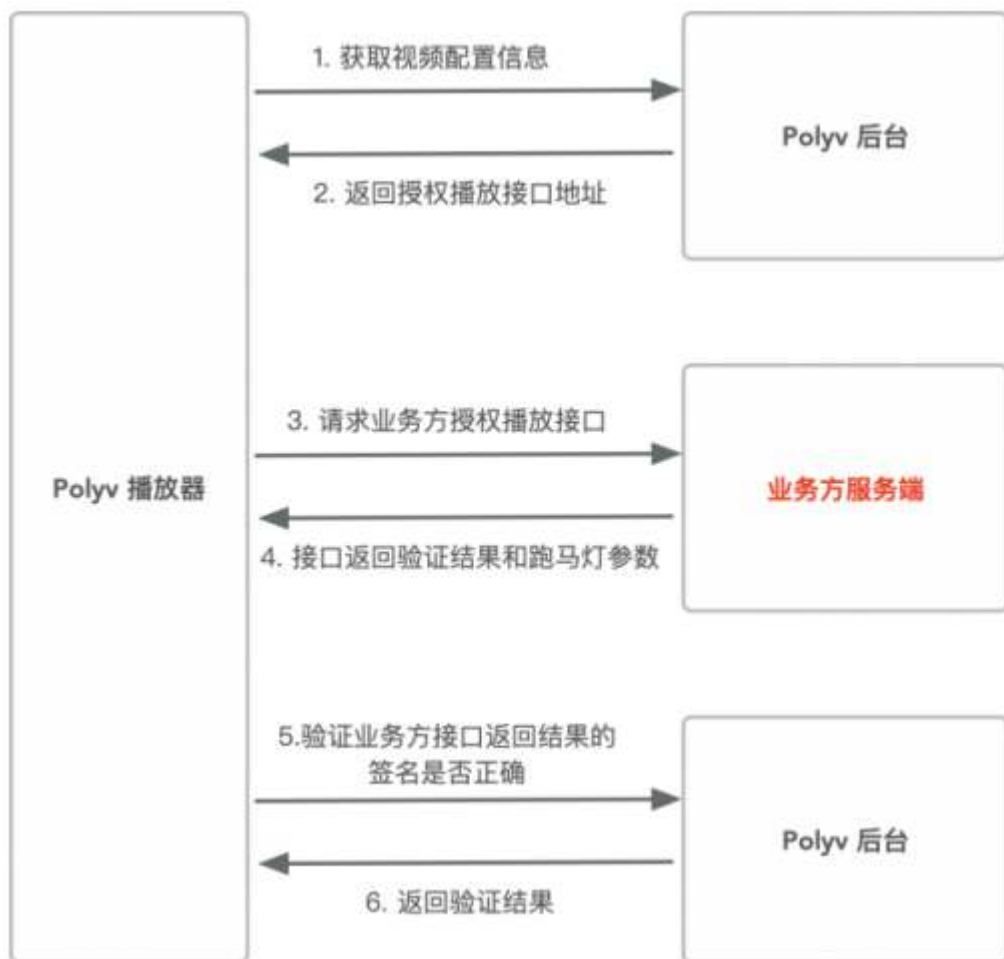
授权播放和跑马灯

屏幕录像是最难防范的一种视频盗版方式，保利威播放器提供的防录屏跑马灯功能，通过设定文字内容（一般是观众的身份ID信息）在视频上不规则滚动，以此来警示盗版者，达到视频版权保护的效果。

另外在用户网站中，除了通过登录信息（cookies）验证观众是否有权限访问视频播放页面外，还可以通过保利威播放器验证观众是否有播放某一个视频的权限，从而实现对观众权限的双重验证。

防录屏跑马灯的视频介绍：[防录屏神器-跑马灯（视频）](#)。

实现流程



实现步骤

一、管理后台设置

1. 登录云点播管理后台，点击 **【设置】** → **【视频设置】**，进入视频设置页面。
2. 在授权播放和防录屏跑马灯接口设置栏中，填写业务方的接口服务URL。

授权播放和防录屏跑马灯接口设置

`https://myDomain.com/auth`

注：目前跑马灯需要通过h5或flash来实现，移动端网页暂不支持，请查看[跑马灯使用帮助](#)了解更多

播放器在请求授权接口时，会自动适配以下情况：

1. 当填写的是完整的URL时，播放器会直接请求，例如：
<http://mywebsite.com/interface/validate>。
2. 当填写的是不带协议头的接口地址时，播放器会根据当前页面的请求协议自动补全。例如填写的是：`//mywebsite.com/interface/validate`，当前页面使用https协议访问，那么实际请求的是：
<https://mywebsite.com/interface/validate>。
3. 当填写的是不带协议头和HOST的接口地址时，播放器会根据当前页面的请求协议和HOST自动补全。例如填写的是：`/interface/validate` 或 `interface/validate`，当前页面的域名是 `mywebsite2.com`，且页面使用http协议访问，那么实际请求的是：
<http://mywebsite2.com/interface/validate>。

二、业务方服务端实现

1、播放器端请求

当管理后台设置授权播放和跑马灯接口后，Polyv播放器在播放视频时会首先请求后台设置的接口，请求的方式为GET，并且会附带vid、code、t、callback这四个参数，例如：https://www.mywebsite.com/validate?vid=e2e84a73837363106d8d257f60e55c4c_e&code=&t=1457938821973&callback=polyvObject16209048491895664483_1457938783908&_=1457938784101。

其中code为播放器嵌入代码中的参数，值可以自定义；t为播放器产生的随机数。播放器示例代码如下：

```
<script src='https://player.polyv.net/script/player.js'></script>
<div id='player'></div>
<script>
var player = polyvPlayer({
  wrap: '#player',
  width: 800,
  height: 533,
  vid: '88083abbf5bcf1356e05d39666be527a_8',
  code: 'myCodeValue' // 用户可自定义参数值，也可以不设置此参数，那么在请求接口时该参数值为空。参数值为中文时需要做
base64URLSafe。
});
</script>
```

由于H5播放器通过Ajax方式请求用户接口，需要跨域请求，因此需要callback参数。而Flash播放器是通过跨域文件实现跨域，所以不需要callback参数，请求接口时只会提交vid、code和t三个参数。Flash播放器实现跨域请参考[跨域访问设置](#)。

2、服务端接口实现

业务方服务端接口如果只需实现授权验证的功能，只需返回 **status**、**username**、**sign**三个参数给播放器验证即可。

服务端实现的PHP示例如下：

```

// validate.php
<?php
$username = "elvis"; // 用户昵称，若值为中文需要urlencode('张三')，可从session获取
$secretkey = "secretkey"; // 登录保利威管理后台，点击 【设置】 → 【API接口】获取
$vid=$_GET["vid"];
$t = $_GET["t"];
$code = $_GET["code"];

if($username=="elvis"){
    $status = 1; // 业务方可自定义授权验证逻辑
}else {
    $status = 2;
}
if(!empty($_GET["callback"])){
    $callback = $_GET["callback"];
}else{
    $callback = '';
}

$sign=md5("vid=$vid&secretkey=$secretkey&username=$username&code=$code&status=$status&t=$t");
$array=Array("status"=>$status,"username"=>$username,"sign"=>$sign);
$json = json_encode($array);

if($callback!=''){ //PC H5播放器会提交callback参数
    echo $callback."(".$json.")";
} else{ //Flash播放器不提交callback参数
    echo "(".$json.")";
}
?>

```

Java SpringMvc示例如下：

```

@ResponseBody
@RequestMapping("/player-auth")
public String playerAuth(String vid, long t, String code, String callback) {
    String username = "elvis";
    String secretKey = "secretkey";
    int status;

    // 业务方可自定义授权验证逻辑
    if ("elvis".equals(username)) {
        status = 1;
    } else {
        status = 2;
    }

    // md5签名, 自行选择md5库
    String plain = "vid=" + vid + "&secretkey=" + secretKey + "&username=" + username + "&code=" + code +
"&status=" + status + "&t=" + t;
    String sign = md5Hex(plain);

    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("username", username);
    resultMap.put("status", status);
    resultMap.put("sign", sign);

    // json序列化
    ObjectMapper objectMapper = new ObjectMapper();
    String resultJson = "";
    try {
        resultJson = objectMapper.writeValueAsString(resultMap);
    } catch (JsonProcessingException e) {
        e.printStackTrace();
    }

    if (callback != null && !"".equals(callback)) {
        return callback + "(" + resultJson + ")";
    } else {
        return resultJson;
    }
}

```

其中，sign的计算规则为：拼接vid、secretkey、username、code、status、t参数做MD5计算：

```

Plain = "vid=" + vid + "&secretkey=" + secretKey + "&username=" + username + "&code=" + code + "&status=" + status
+ "&t=" + t
sign = MD5.hash(Plain);

```

以下为接口返回示例：

```
polyvObject16208229674372271079_1478765178186({  
  "status":1,  
  "username":"elvis",  
  "sign":"1cca74bd55c6076091ed84807065e5b7"  
})  
// 不提交callback参数时  
{  
  "status":1,  
  "username":"elvis",  
  "sign":"2c2bfb00314da7d768d50a7d1e93bd9f"  
}
```

如果除了授权验证，还需实现跑马灯功能，则接口需要返回跑马灯相关的参数。

服务端实现的PHP示例如下：

```

// validate.php
<?php
$username = "elvis"; // 用户昵称，若值为中文需要urlencode('张三')，可从session获取
$secretkey = "secretkey"; // 登录保利威管理后台，点击 【设置】 → 【API接口】获取
$vid=$_GET["vid"];
$t = $_GET["t"];
$code = $_GET["code"];
$fontSize="40";
$fontColor="0xFFE900";
$speed="200";
$filter="on";
$setting="3";
$alpha="1";
$filterAlpha="1";
$filterColor="0x3914AF";
$blurX="2";
$blurY="2";
$tweenTime="1";
$interval="5";
$lifeTime="3";
$strength="4";
$show="on";
$msg="ErrorMessage!";

if($username=="elvis"){ // 业务方可自定义授权验证逻辑
    $status = 1;
}else {
    $status = 2;
}

if(!empty($_GET["callback"])){
    $callback = $_GET["callback"];
}else{
    $callback = '';
}

$sign=md5("vid=$vid&secretkey=$secretkey&username=$username&code=$code&status=$status&t=$t&msg=$msg&fontSize=$font
ntSize&fontColor=$fontColor&speed=$speed&filter=$filter&setting=$setting&alpha=$alpha&filterAlpha=$filterAlpha&f
ilterColor=$filterColor&blurX=$blurX&blurY=$blurY&interval=$interval&lifeTime=$lifeTime&tweenTime=$tweenTime&str
ength=$strength&show=$show");
$array =
Array("status"=>$status,"username"=>$username,"sign"=>$sign,"msg"=>$msg,"fontSize"=>$fontSize,"fontColor"=>$font
Color,"speed"=>$speed,"filter"=>$filter,"setting"=>$setting,"alpha"=>$alpha,"filterAlpha"=>$filterAlpha,"filterC
olor"=>$filterColor,"blurX"=>$blurX,"blurY"=>$blurY,"tweenTime"=>$tweenTime,"interval"=>$interval,"lifeTime"=>$l
ifeTime,"strength"=>$strength,"show"=>$show,);
$json = json_encode($array);

if($callback!=''){
    echo $callback."(".$json.")";
} else{
    echo $json;
}
?>

```

Java SpringMvc示例如下：

```

@ResponseBody
@RequestMapping("/player-auth-marquee")
public String playerAuthMarquee(String vid, long t, String code, String callback) {
    String username = "suki";
    String secretKey = "AiDQw1mAmi";
    String fontSize = "40", fontColor = "0xFFE900", speed = "200",
        filter = "on", setting = "3", alpha = "1", filterAlpha = "1",
        filterColor = "0x3914AF", blurX = "2", blurY = "2", tweenTime = "1",
        interval = "5", lifeTime = "3", strength = "4", show = "on", msg = "ErrorMessage!";
    int status;

    // 业务方可自定义授权验证逻辑
    if ("suki".equals(username)) {
        status = 1;
    } else {
        status = 2;
    }

    code = code == null ? "" : code;

    // md5签名, 自行选择md5库
    String plain = "vid=" + vid + "&secretkey=" + secretKey + "&username=" + username + "&code=" + code +
        "&status=" + status + "&t=" + t +
        "&msg=" + msg + "&fontSize=" + fontSize + "&fontColor=" + fontColor + "&speed=" + speed + "&filter="
        + filter + "&setting=" + setting +
        "&alpha=" + alpha + "&filterAlpha=" + filterAlpha + "&filterColor=" + filterColor + "&blurX=" +
        blurX + "&blurY=" + blurY +
        "&interval=" + interval + "&lifeTime=" + lifeTime + "&tweenTime=" + tweenTime + "&strength=" +
        strength + "&show=" + show;
    String sign = md5Hex(plain);

    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("username", username);
    resultMap.put("status", status);
    resultMap.put("sign", sign);
    resultMap.put("msg", msg);
    resultMap.put("fontSize", fontSize);
    resultMap.put("fontColor", fontColor);
    resultMap.put("speed", speed);
    resultMap.put("filter", filter);
    resultMap.put("setting", setting);
    resultMap.put("alpha", alpha);
    resultMap.put("filterAlpha", filterAlpha);
    resultMap.put("filterColor", filterColor);
    resultMap.put("blurX", blurX);
    resultMap.put("blurY", blurY);
    resultMap.put("tweenTime", tweenTime);
    resultMap.put("interval", interval);
    resultMap.put("lifeTime", lifeTime);
    resultMap.put("strength", strength);
    resultMap.put("show", show);

    // json序列化, 可自行选择json序列化库
    ObjectMapper objectMapper = new ObjectMapper();
    String resultJson = "";

```

```

try {
    resultJson = objectMapper.writeValueAsString(resultMap);
} catch (JsonProcessingException e) {
    e.printStackTrace();
}

if (callback != null && !"".equals(callback)) {
    return callback + "(" + resultJson + ")";
} else {
    return resultJson;
}
}

```

其中，**sign**的计算规则为（参数必须按照示例中的顺序拼接）：

```

Plain = "vid=" + vid + "&secretkey=" + secretKey + "&username=" + username + "&code=" + code + "&status=" +
status + "&t=" + t +
"&msg=" + msg + "&fontSize=" + fontSize + "&fontColor=" + fontColor + "&speed=" + speed + "&filter=" + filter +
"&setting=" + setting +
"&alpha=" + alpha + "&filterAlpha=" + filterAlpha + "&filterColor=" + filterColor + "&blurX=" + blurX +
"&blurY=" + blurY +
"&interval=" + interval + "&lifeTime=" + lifeTime + "&tweenTime=" + tweenTime + "&strength=" + strength +
"&show=" + show;
sign = MD5.hash(Plain);

```

例如：当

vid="8f8482aaab11dd5f45f183a9192a04c5_8",secretkey="AiDQw1mAmi",username="suki",code="abc",status="1",t="143020010115550947",msg="Errormessage!",fontSize="40",fontColor="0xFFE900",speed="200",filter="on",setting="3",alpha="1",filterAlpha="1",filterColor="0x3914AF",blurX="2",blurY="2",interval="5",lifeTime="3",tweenTime="1",strength="4",show="on" 时，拼凑起来去MD5计算的字符串为：

vid=8f8482aaab11dd5f45f183a9192a04c5_8&secretkey=AiDQw1mAmi&username=suki&code=abc&status=1&t=143020010115550947&msg=Errormessage!&fontSize=40&fontColor=0xFFE900&speed=200&filter=on&setting=3&alpha=1&filterAlpha=1&filterColor=0x3914AF&blurX=2&blurY=2&interval=5&lifeTime=3&tweenTime=1&strength=4&show=on 则sign为MD5计算后32位小写的值： 3b07f56f29b7fd728bf20020442338e7

以下为接口返回示例：

```
{
  "status":1,
  "username":"elvis",
  "sign":"6ab63590797e513d1b6c46b407413478",
  "msg":"Errormessage!",
  "fontSize":"40",
  "fontColor":"0xFFE900",
  "speed":"200",
  "filter":"on",
  "setting":"3",
  "alpha":"1",
  "filterAlpha":"1",
  "filterColor":"0x3914AF",
  "blurX":"2",
  "blurY":"2",
  "tweenTime":"1",
  "interval":"5",
  "lifeTime":"3",
  "strength":"4",
  "show":"on"
}
```

三、接口返回参数说明

参数名	类型	必填	默认值	说明
status	Integer	是	/	是否允许播放：1 允许 2 禁止
username	String	是	/	观众名称，也会用于跑马灯显示的文字内容
sign	String	是	/	接口签名，用于校验返回内容是否被篡改
show	String	是	off	当参数值为“on”时表示显示跑马灯，默认不显示
setting	Integer	是	1	跑马灯滚动的样式： 1 从右到左滚动 2 随机位置闪烁 3 从右到左闪烁滚动

参数名	类型	必填	默认值	说明
speed	Integer	是	200	跑马灯文字从右侧移至左侧所需时间，单位：1/10秒，仅对setting(1、3)生效
lifeTime	Integer	是	3	跑马灯文字显示时间，单位：秒，仅对setting(2)生效
interval	Integer	是	5	跑马灯文字隐藏间隔时间，单位：秒，仅对setting(2、3)生效，
tweenTime	Integer	是	1	跑马灯文字渐隐渐现时间，单位：秒 (已废弃)
fontSize	Integer	是	30	跑马灯文字的字体大小
fontColor	String	是	0x000000	跑马灯文字颜色，使用十六进制颜色值表示，如0xFF0000，默认为黑色
alpha	Float	是	1	跑马灯文字透明度，取值范围0.01~1，参数值不能小于0.01
filter	String	是	off	跑马灯文字是否描边，on 描边 off 不描边
filterAlpha	Float	是	1	文字描边透明度，取值范围0~1
filterColor	String	是	0x000000	文字描边颜色，使用十六进制颜色值表示，如0xFF0000，默认为黑色
strength	Integer	是	4	描边强度，取值范围0~255

参数名	类型	必填	默认值	说明
blurX	Integer	是	2	描边水平模糊量，取值范围0~255
blurY	Integer	是	2	描边垂直模糊量，取值范围0~255
msg	String	是	/	自定义错误提示信息

- setting(1): 跑马灯由右向左滚动，`speed/10` 秒完成滚动，间隔2秒后进行下一次滚动
- setting(2): 跑马灯在随机位置交替显示隐藏，每次显示 `lifeTime` 秒后隐藏 `interval` 秒，然后下一次循环开始
- setting(3): 跑马灯由右向左滚动，`speed/10` 秒完成滚动，间隔 `speed/10 + interval` 秒后进行下一次滚动，滚动期间每3秒隐藏一次，隐藏3秒后显示

1. 当接口只做授权验证用途时，只需返回 `status`、`username`和`sign`三个参数，跑马灯相关参数可不返回。
2. 请确保接口返回数据是utf-8编码，注意 `status`参数是整型，不能是字符串类型。
3. 目前跑马灯功能不支持在移动端H5播放器上使用（1.0不支持，建议使用2.0）。
4. 请尽量不要对播放器样式或`<video>`标签进行修改