

获取视频播放凭证

接口描述

- 1、获取Playsafe Token（播放凭证），用于播放加密视频

2、如果一个token尚未过期，此时使用相同的videoId、viewerId、viewerIp、iswxa参数值请求该接口，则会复用原来的token，并延长原token的有效期

3、接口支持https协议

接口URL

http://hls.videocc.net/service/v1/token

在线API调用

请求方式

POST

接口约束

- 1、接口同时支持HTTP 、HTTPS ， 建议使用HTTPS 确保接口安全，接口调用有频率限制，[详细请查看](#)

请求参数描述

参数名	必选	类型	说明
userid	true	String	保利威点播账户id，可以参考【 获取密钥 】获取，获取路径：官网->登录->点播（API接口）
sign	true	String	签名，为32位大写的MD5值【详见 MD5签名生成规则 】

参数名	必选	类型	说明
videoId	true	String	视频id，例如： e6b23c6f519c5906e54a13b8200d7bb0_e
ts	true	Long	当前13位毫秒级时间戳， 10分钟内有效
viewerId	true	String	观看者id，要求不同的观看者使用不同的id，由数字字母组成；移动端请求需对viewerId进行base64处理
viewerIp	false	String	观看者ip，如果为空，会自动获取调用该接口时的ip
viewerName	false	String	观看者名称
expires	false	Long	token有效时长，单位为秒。为空时默认为10分钟，有效期为最长24小时。
disposable	false	Boolean	token有效期，默认为false true: token仅一次有效（验证一次后，token就失效了） false: 在有效期内可以进行多次验证。
iswxa	false	Integer	是否微信小程序播放，默认为0 1: 是 0: 否
type	false	String	设备端类型，用于限制请求端，参数值：WEB-网页端，NATIVE-原生端。
extraParams	false	String	自定义的其它参数

参数名	必选	类型	说明
securitySeed	false	String	私有加密视频自定义token加密因子, 用于提高视频播放的安全性, 需搭配指定的播放器使用, 详细使用规则请咨询保利威客服

示例

```
http://hls.videocc.net/service/v1/token
```

表单参数：

```
viewerName=ceshi&expires=300&viewerId=ceshiid&viewerIp=127.0.0.1&sign=A5423CC62E20F80AD8812395119E5FA9&videoId=1b448be323b68b2999802799a98dba54_1&iswxa=1&userid=1b448be323&disposable=true&ts=1621851214050
```

响应参数描述

参数名	类型	说明
code	Integer	响应状态码，200为成功返回，非200为失败【详见 全局错误说明 】
status	String	响应状态文本信息
message	String	响应描述信息，当code为400或者500的时候，辅助描述错误原因
data	Object	响应成功时返回token相关数据【详见 data字段说明 】

data字段说明

参数名	类型	说明
token	String	token值

参数名	类型	说明
userId	String	保利威点播账户id, 可以参考【 获取密钥 】获取, 获取路径: 官网->登录->点播 (API接口)
appId	String	账号appId【详见 获取密钥 】
videoId	String	视频id, 例如: e6b23c6f519c5906e54a13b8200d7bb0_e
viewerId	String	观看者id, 要求不同的观看者使用不同的id
viewerIp	String	观看者ip, 如果为空, 会自动获取调用该接口时的ip
viewerName	String	观看者名称
ttl	Long	token有效时长, 单位为毫秒
disposable	Boolean	token有效期 true: token仅一次有效 (验证一次后, token就失效了) false: 在有效期内可以进行多次验证。
iswxa	Integer	是否微信小程序播放 1: 是 0: 否
extraParams	String	自定义的其它参数
createTime	Long	token创建时间, 13位毫秒级时间戳
expiredTime	Long	token过期时间, 13位毫秒级时间戳

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 **VodSignUtil.java** 都包含在下载文件中。

强烈建议您使用[点播Java SDK](#)完成API的功能对接, 点播Java SDK 对API调用逻辑、异常处理、数据

签名、HTTP请求线程池进行了统一封装和优化。

```
private static final Logger log = LoggerFactory.getLogger(VodTokenTest.class);

/**
 * 获取视频播放凭证
 */
@Test
public void testCreateToken() throws Exception, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String secretKey = super.secretKey;
    String userid = super.userId;
    //业务参数
    String url = "http://hls.videocc.net/service/v1/token";
    String videoId = "1b448be323b68b2999802799a98dba54_1";
    String ts = Long.toString(System.currentTimeMillis());
    String viewerId = "ceshiid";
    String viewerIp = "127.0.0.1";
    String viewerName = "ceshi";
    String expires = "300";
    String disposable = "true";
    String iswxa = "1";

    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("userid", userid);
    requestMap.put("videoId", videoId);
    requestMap.put("ts", ts);
    requestMap.put("viewerId", viewerId);
    requestMap.put("viewerIp", viewerIp);
    requestMap.put("viewerName", viewerName);
    requestMap.put("expires", expires);
    requestMap.put("disposable", disposable);
    requestMap.put("iswxa", iswxa);

    //用md5进行签名
    requestMap.put("sign", VodSignUtil.getSignMd5(requestMap, secretKey));

    String response = HttpUtil.postFormBody(url, requestMap);
    log.debug("测试获取视频播放凭证, {}", response);
    //do somethings
}
```

响应示例

系统全局错误说明详见[全局错误说明](#)

成功示例

```
{
  "code": 200,
  "status": "success",
  "message": "",
  "data": {
    "token": "586e4510-d584-40de-b4a6-8cd0d363c9d0-smcdibbk",
    "userId": "1b448be323",
    "appId": null,
    "videoId": "1b448be323b68b2999802799a98dba54_1",
    "viewerIp": "127.0.0.1",
    "viewerId": "ceshiid",
    "viewerName": "ceshi",
    "extraParams": null,
    "ttl": 300000,
    "createdTime": 1617776662954,
    "expiredTime": 1617776962954,
    "iswxa": 1,
    "disposable": true
  }
}
```

异常示例

userid 错误

```
{
  "code": 400,
  "status": "error",
  "message": "user_not_found",
  "data": "user secretKey not found."
}
```

ts过期

```
{
  "code": 403,
  "status": "error",
  "message": "ts_expired",
  "data": "ts parameter is expired."
}
```

签名错误

```
{  
  "code": 403,  
  "status": "error",  
  "message": "sign_invalid",  
  "data": "sign parameter invalid."  
}
```