视频点播

# 最佳实践

| | |
|---|---|
| 文档版本 | 01 |
| 发布日期 | 2024-12-31 |

华为云计算技术有限公司

### 商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

### 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

# 目 录



# 1 上传音视频文件至 VOD

## 场景说明

视频业务在各领域兴起，大量音视频文件需要一个平台对其进行处理和加速分发，华为云视频点播服务就是一个实现视频上传、自动化转码处理、媒资管理、分发加速的一站式媒体服务。在使用点播服务丰富的媒资处理功能前，您需要将这些音视频文件上传到点播服务。为了满足源文件不同的存储方式的场景，华为云点播服务同样也提供了多样的上传方式。

## 上传方式

视频点播提供了的音视频上传方式如表1-1所示

**表 1-1** 上传方式

| 上传方式 | 适用场景 |
| --- | --- |
| 本地上传 | 适用于将存储在本地磁盘的音视频文件迁移至VOD。 |
| OBS转存 | 适用于将OBS桶中的音视频复制迁移至VOD的场景。 |
| URL拉取 | 适用于将网络上的音视频文件拉取存储至VOD的场景。 |

## 本地上传

支持批量上传音视频文件，便于快速将媒资上传到点播服务中，使用浏览器登录控制台即可进行上传。

华为云点播服务提供的本地上传功能有如下限制：

- 控制台由于安全策略，长时间上传可能会由于登录失效导致大文件上传失败，在上传大量文件时，需要操作控制台，从而保证控制台不自动退出登录。
- 支持上传的格式如下所示：

- 视频文件格式：MP4，TS，MOV，MXF，MPG，FLV，WMV，AVI，M4V，F4V，MPEG，3GP，ASF，MKV，WEBM，M3U8、VOB、RM、MTS。其中，M3U8仅支持URL拉取方式上传。
- 音频文件格式：MP3，OGG，WAV，WMA，APE，FLAC，AAC，AC3，MMF，AMR，M4A，M4R，WV，MP2。

**通过控制台本地上传**

**步骤1** 登录**视频点播控制台**。

**步骤2** 在左侧导航树中，选择“上传音视频 > 本地上传”，进入本地上传页面。

**步骤3** 单击“本地上传”，弹出“本地上传”窗口。

**步骤4** 单击“添加音视频”添加本地媒资文件，或者直接将本地文件拖拽至界面区域。

**图 1-1** 本地上传


**步骤5** 您可以根据实际需求在“音视频处理”中选择转码模板或工作流。上传成功后自动对上传的音视频进行处理。

**步骤6** 单击“开始上传”，开始上传音视频文件。

上传时间与文件大小和网络情况都有关。

**----结束**

**调用API完成本地上传**

- 若您上传的音视频文件大小小于20M，则不需要先分段再上传，具体请参见**媒资上传（20M以下）**。
- 若您上传的音视频文件大小大于20M，则您需要先将音视频文件进行分段，分段大小必须小于20M，然后再分别上传各分段，具体请参见**媒资上传（20M以上）**。

## OBS 转存

若在开通点播服务前，已在华为云OBS桶中存储了大量的音视频文件，希望使用视频点播的转码、截图等功能对这些音视频进行处理。您可以使用该功能将OBS桶中的音视频文件复制转存到点播服务中，然后使用点播服务的相关功能。

华为云点播服务提供的OBS转存功能有如下限制：

- 不支持跨区域转存，如“华北-北京四”OBS桶中的音视频只能转存到“华北-北京四”点播服务中。

- 转存是指将OBS桶中的音视频文件复制一份到点播服务中，因此，若OBS桶中的音视频不删除，则OBS和点播服务中都将会有相关的存储费用产生。
- 支持转存的格式如下所示：
  - 视频文件格式：MP4，TS，MOV，MXF，MPG，FLV，WMV，AVI，M4V，F4V，MPEG，3GP，ASF，MKV，WEBM，M3U8、VOB、RM、MTS。其中，M3U8仅支持URL拉取方式上传。
  - 音频文件格式：MP3，OGG，WAV，WMA，APE，FLAC，AAC，AC3，MMF，AMR，M4A，M4R，WV，MP2。

OBS转存方式暂只支持调用API实现，您可以调用**创建媒资：OBS转存**接口实现该功能。

## URL 拉取

支持将其它网络音视频拉取上传到点播服务，华为云点播服务提供的URL提取功能具有如下限制：

- 控制台一次最多支持拉取100条音视频，API一次最多支持拉取16条音视频。
- 拉取的URL需要直接指向音视频文件，不能是某个网站的页面链接，且后缀必须是音视频格式，当前支持拉取的URL后缀如下所示：
  - 视频文件格式：MP4，TS，MOV，MXF，MPG，FLV，WMV，AVI，M4V，F4V，MPEG，3GP，ASF，MKV，WEBM，M3U8、VOB、RM、MTS。其中，M3U8仅支持URL拉取方式上传。
  - 音频文件格式：MP3，OGG，WAV，WMA，APE，FLAC，AAC，AC3，MMF，AMR，M4A，M4R，WV，MP2。
- 拉取的URL协议暂只支持HTTPS和HTTP协议，其中，HTTP协议存在安全风险，建议使用HTTPS协议。
- 从其它云服务商拉取过来的音视频将会生成新的媒资ID，无法继承原有媒资ID。

**通过控制台拉取上传**

**步骤1** 登录**视频点播控制台**。

**步骤2** 在左侧导航栏中，选择“上传音视频 > URL拉取”，进入URL拉取页面。

**步骤3** 单击“URL拉取”，进入“URL拉取”页面，如**图1-2**所示。

输入拉取文件信息。还可以根据实际需求在“音视频处理”中选择转码模板或工作流。拉取成功后会自动对拉取的音视频进行处理。

**图 1-2** URL 拉取设置


**步骤4** 单击“确定”，可在URL拉取列表中查看到任务执行情况。

**----结束**

**调用点播API拉取上传**

1. 获取调用API的用户Token，具体请参见**构造请求**。
2. 调用**创建媒资：URL拉取注入**接口，在请求参数中设置批量拉取的URL及音视频处理参数。
3. 调用**查询媒资列表**接口，查看拉取结果。

   建议您等待一两分钟后（具体根据视频文件的大小有所差别），再查询拉取结果。

# 2 自定义视频封面

## 场景说明

随着视频点播存储视频文件量的增加，设置视频封面不仅能提升展示的美观性，还能方便通过封面查找相关视频。同时，上传的视频封面也将会生成对应的封面地址，实现加速分发，可以直接将封面与视频文件一同引用到网页中。

上传视频时，点播服务会默认截取视频第一秒的首帧作为封面图片。您也可以通过上传图片或截图封面来更新视频的封面。

- 上传封面：适用于需要通过封面表达视频的大概内容及重点的场景。在上传前，您需要提前线下设计一张JPG或PNG格式的封面图片。
- 截图封面：适用于希望将视频中的某个瞬间画面来设置成封面的场景。无需要额外准备工作，您可以直接使用视频点播的截图功能生成。

您可以通过以下方式自定义视频封面：

- **控制台设置视频封面**
- **调用API设置视频封面**

## 控制台设置视频封面

在视频点播控制台的音视频管理中上传封面图片。

- 暂只支持对视频格式为FLV、MP4、TS、MOV、MXF、MPG、WMV、AVI、M4V、F4V、MPEG、ASF、MKV、3GP、WEBM、VOB、RM、MTS的视频文件进行截图。
- 暂只支持截图生成JPG格式的图片文件。

1. 登录**视频点播控制台**。
2. 在左侧导航栏中，选择“音视频管理”，进入音视频管理页面。
3. 在需要上传封面的视频行单击“管理”，在“基本信息”页签单击“编辑”。
4. 单击添加框，上传本地的封面图片，上传完成后单击“保存”即可。

在视频点播控制台的“视频处理 > 视频截图”中设置封面图片。

1. 登录**视频点播控制台**。
2. 在左侧导航栏中，选择“视频处理 > 视频截图”，进入视频截图页面。
3. 勾选需要截图的视频，单击“截图”，在弹出的截图设置框中设置截图参数。

**图 2-1** 截图设置


- 按时间间隔：根据设置的时间间隔，从视频首帧开始截图，以最后一帧截图结束，最大间隔不能超过12秒。可选择是否将首帧截图作为视频封面。
- 指定时间点：根据设置指定时间从视频中截取图片，最多可设置10个时间点。可选择是否将某张截图作为视频封面。

4. 单击“确定”，视频开始截图。
5. 截图完成后，单击“详情”，可以在详情页面中选择某一张截图作为封面。

## 调用 API 设置视频封面

视频点播API支持通过媒资上传、视频更新、视频处理三种方式设置视频的封面图片。

- 视频上传时设置封面

  视频点播提供了上传、OBS转存、URL拉取三种创建媒资的方式，其中上传方式创建媒资支持上传或截图封面，其它二种方式仅支持截图封面。具体如下所示：

  - 上传方式创建媒资

    **上传封面**：调用**创建媒资：上传方式**接口，在请求参数中设置“cover_type”，即上传封面的图片类型，然后在请求的返回参数中获取“cover_upload_url”，通过“cover_upload_url”上传封面图片即可。

    **截图封面**：调用**创建媒资：上传方式**接口，在请求参数中设置“thumbnail”，设置截图类型，指定某张截图作为封面。

  - OBS转存和URL拉取方式创建媒资

    分别调用创建媒资：**OBS转存方式**接口或**创建媒资：URL拉取注入**接口，在请求参数中设置“thumbnail”，设置截图类型，指定某张截图作为封面。

- 视频更新时设置封面

  调用**视频更新**接口，在请求参数中设置“cover_type”，即上传封面的图片类型，然后在请求的返回参数中获取“cover_upload_url”，通过“cover_upload_url”上传封面图片即可。

- 视频处理时设置封面

  调用**视频处理**接口生成截图，然后指定某张截图作为封面。若您需要在已生成的截图中更换封面，则可以先调用查询媒资详细信息接口，获取该视频的截图URL，然后调用**设置封面**接口修改截图封面。

# 3 通过防盗链控制音视频的播放权限

## 场景说明

为对分发的音视频进行播放权限控制，点播服务提供了防盗链功能。开启后，CDN会对所有播放请求中携带的关键信息进行校验，仅校验通过的请求会予以响应，其它非法的访问将直接返回403。防盗链方案中包含Referer防盗链和Key防盗链。

Referer防盗链是基于HTTP协议支持的Referer机制实现的，通过播放请求中携带的Referer字段识别请求来源。配置黑名单或白名单，CDN将根据名单对请求来源进行过滤，从而达到最基本的访问控制的目的。Referer防盗链具有配置便捷，无需额外开发，快速生效等优势，适用于音视频主要在web端引用的场景。

由于HTTP Header的内容可伪造，Referer防盗链只能达到最基本的保护，安全性不高。所以还可以采用Key防盗链方案，通过鉴权播放URL保障点播资源的安全。由于鉴权的Key值具有时效性，所以适用于对媒资安全要求比较高的场景。

本示例中，开启Referer防盗链，只允许白名单中的域名访问视频文件，然后通过Key防盗链生成鉴权播放地址。

## 实现原理

Referer防盗链的实现原理比较简单，在点播控制台配置了白名单或黑名单后，点播服务会将这份名单分发到CDN中。当CDN接收到资源请求时，会根据这个名单来识别请求是否合法，若合法，则访问请求的资源，否则拒绝并返回403。

Key防盗链是视频点播的加速节点与点播源站联合实现的，比Referer防盗链更为安全可靠的一种防盗播方案。Key防盗链的实现过程如**图3-1**所示。

**图 3-1** Key 防盗链实现原理


流程说明如下所示：

1. 租户在点播控制台开启Key防盗链功能，并配置误差允许时间、算法等。
2. 点播服务将配置的密钥值等下发到CDN节点中。
3. 租户通过点播服务获取到点播媒资的鉴权URL。
4. 观众通过租户提供的鉴权播放URL向CDN请求视频播放。
5. CDN根据播放URL中携带的鉴权信息校验请求的合法性，仅校验通过的请求会被允许。

## 配置 Referer 防盗链

先配置Referer防盗链，限制请求来源，使点播资源得到最基本的安全保障。

**说明**

当前Referer防盗链配置不支持带端口。

**步骤1** 登录**视频点播控制台**。

**步骤2** 在左侧导航栏中，选择“域名管理”，进入域名管理界面。

**步骤3** 单击域名右侧“配置 ”，在“防盗链”页签单击“Referer防盗链”。

**步骤4** 在弹出的配置框中打开“开关”，并配置相关参数。

**图 3-2** Referer 防盗链配置


- 类型：支持黑名单和白名单模式。
  - Referer黑名单：表示黑名单内的域名不允许访问点播资源，其它可以访问。若同时勾选了“包含空Referer”，则表示不允许HTTP Header中Referer为空的请求。
  - Referer白名单：表示白名单内的域名允许访问点播资源，其它不可以访问。若同时勾选了“包含空Referer”，则表示允许HTTP Header中Referer为空的请求。
- 规则：名单详情，最多支持4级域名，最多支持100条，以英文“;”分隔。域名、IP地址可混合输入，支持泛域名添加。域名前不能带协议名（http://和https://）。

  示例：www.example.com;*.test.com;192.168.0.0

**步骤5** 单击“确定”，完成配置。

大约需要3-5分钟，Referer防盗链才生效。

**----结束**

## 配置 Key 防盗链

为点播资源配置Key防盗链，通过鉴权URL的时效性来进一步加强点播资源的安全。

**步骤1** 登录视频点播控制台。

**步骤2** 在左侧导航栏中，选择“域名管理”，进入域名管理界面。

**步骤3** 单击域名右侧“配置 ”，在“防盗链”页签选择“Key防盗链”。

**步骤4** 在弹出的配置框中打开“防盗链开关”，并配置相关参数。

**图 3-3** 配置 Key 防盗链


**表 3-1** 参数说明

| 参数 | 描述说明 |
| --- | --- |
| 密钥值 | 即Key值，单击“生成”，可自动生成符合格式的密钥值。 |
| 误差允许时间 | 表示当前防盗链有效时间，默认为120分钟。<br>示例：若鉴权URL生成时间为1573806090（2019/11/15 16:21:30），配置的“误差允许时间”为120分钟，则鉴权URL的失效时间为2019/11/15 18:21:30。 |
| 旧key的失效时间 | 采用新Key时，旧Key值的失效时间，从新Key生效时开始计算，默认60分钟后失效。<br>示例：若新Key生效时间为2019/11/15 16:21:30，配置的“旧Key的失效时间”为60分钟，则旧鉴权URL的真正失效时间为2019/11/15 17:21:30。 |
| 算法 | 加密算法，支持ABCD四种算法，默认为算法D。选择的算法不同，生成的鉴权URL也是不同的，具体的生成规则可参考**Key防盗链**。<br>**说明**<br>算法ABC暂不支持HLS和DASH播放场景，建议使用算法D。 |

**步骤5** 单击“确定”，完成参数配置。

**步骤6** 提交工单申请审核，提交的信息需要包含配置的域名，及表3-1中的信息。

审核通过后Key防盗链功能才会生效。若修改了Key防盗链的配置，也需重新提交工单审核。

**----结束**

## 验证防盗链功能

- 验证Referer防盗链

  在Referer防盗链中配置的referer白名单为“www.huaweicloud.com”，且不勾选“不包含空Referer”。在“http://www.example.com/test/test.html”网页中引用点播服务中的视频文件“https://1280.cdn-vod.huaweicloud.com/input/1.mp4”，访问该网页并播放视频，若播放失败则表示Referer防盗链生效。

- 验证Key防盗链

  a. 登录视频点播控制台，在左侧导航树中选择“音视频管理”。

  b. 在某个音视频行单击“管理”，选择“播放地址”页签，获取播放地址。

  **图 3-4** 播放地址

  其中“地址”列为原始播放地址，单击可获取鉴权播放地址。

  c. 在播放器端分别播放原始播放地址和鉴权播放地址，若原始播放地址播放失败，鉴权播放地址播放成功，则表示Key防盗链生效。

# 4 通过 HLS 加密防止视频泄露

## 环境准备

已下载SDK。

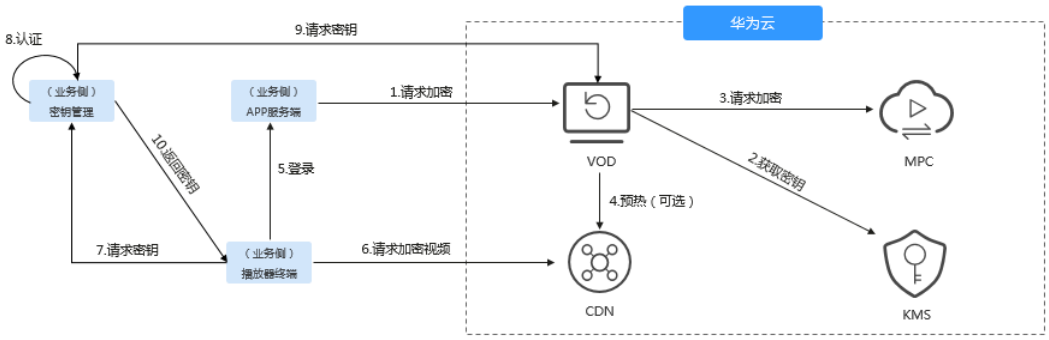
## 场景说明

使用防盗链机制可以控制播放行为，避免非授权用户通过播放URL下载或播放点播视频，但无法阻止恶意的付费用户将视频下载到本地后进行二次分发。

为了有效防止视频泄露和盗链问题，华为云视频点播提供了对HLS视频内容进行加密的能力。加密后的视频，即使恶意用户下载也无法分发给其他人观看。HLS加密涉及到业务侧的密钥服务和Token生成服务的搭建，所以本方案主要适用于能自行搭建一套完整的鉴权及密钥管理服务的业务侧。

## 实现原理

华为云视频点播提供的HLS加密使用的HLS规范中的通用加密方案，通过指定的AES-128加密算法来加密每一个TS，并在生成的m3u8文件中描述播放器如何解密TS文件的方法，支持所有的HLS播放器。




本方案中，点播服务集成了华为云的KMS，向HLS加密提供密钥。

- **加密流程**
  a. 业务侧将视频上传到点播服务（VOD）后，请求HLS加密。
  b. 点播服务收到加密请求后，向KMS请求加密密钥，并将获取的密钥ID和密钥密文存储在点播服务中。

c. 点播服务向媒体处理服务请求HLS加密，媒体处理服务通过转码功能将对应的视频进行加密。

转码加密后生成的m3u8文件带有“#EXT-X-KEY”标签，该标签包含了“METHOD”和“URI”属性，其中“URI”即为业务侧搭建的密钥管理服务的地址，示例如下所示。

**说明**

若加密后使用的播放地址是**https**，则密钥管理服务的地址也需要配置为https协议，否则无法在点播控制台预览播放。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="https://domain-sample/encrypt/get-key?
asset_id=6aee80009c4ca6970f508d6334194794",IV=0x80a3ff24ccd788042ca7f2237e74c59d
#EXTINF:5.000000,  6aee80009c4ca6970f508d6334194794_1_1920X1080_3000_0_0.ts
#EXTINF:5.000000,  6aee80009c4ca6970f508d6334194794_1_1920X1080_3000_0_1.ts
#EXT-X-ENDLIST
```

d. 加密后，点播服务通过CDN将加密的HLS视频文件进行加速分发。

- **解密流程**

a. 终端用户登录播放器终端，业务侧会对终端用户进行身份校验，校验通过后，会为播放终端分配一个Token，并将带Token的播放地址返回给播放器端。

示例：若转码加密后的HLS视频播放地址为：https://1280.cdn-vod.huaweicloud.com/input/test.m3u8，则播放器终端获取的播放地址为：https://1280.cdn-vod.huaweicloud.com/input/test.m3u8?token={**token**}

b. 播放器终端通过带Token的播放URL向CDN请求播放。由于Token是动态的，所以CDN收到请求后，会直接回源到点播服务。点播服务会将请求URL中的Token写入请求的m3u8文件的“URI”中。

点播服务返回给CDN的m3u8文件中会携带播放终端的Token值，示例如下所示。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="https://domain-sample/encrypt/get-key?
asset_id=6aee80009c4ca6970f508d6334194794&token={token}",IV=0x80a3ff24ccd788042ca7f22
37e74c59d
#EXTINF:5.000000,  6aee80009c4ca6970f508d6334194794_1_1920X1080_3000_0_0.ts
#EXTINF:5.000000,  6aee80009c4ca6970f508d6334194794_1_1920X1080_3000_0_1.ts
#EXT-X-ENDLIST
```

c. 播放终端解析返回的m3u8文件，得到EXT-X-KEY标签中的“URI”内容，向“URI”请求密钥。

d. 业务侧的密钥管理服务收到请求后，先验证Token的合法性，若Token合法，则通过调用点播服务的API**查询密钥**。

密钥管理服务可以选择将查询到的密钥缓存在本地，当下次有其它播放终端请求时，可以直接返回，无需每次都向点播服务获取。

e. 密钥管理服务将点播服务返回的密钥返回给播放终端。播放终端通过获取的密钥解密播放m3u8文件。

## 搭建相关服务

若需要使用HLS加密功能，并实现解密播放功能，您需要在您的业务端服务器搭建密钥管理服务和Token生成服务。

- 密钥管理服务，搭建的密钥管理服务需要具备如下功能，密钥管理服务示例代码请参见**示例代码**。
  - 支持身份鉴定：如**实现原理**中描述，密钥管理服务收到密钥请求时，需要验证请求的Token是否合法。
  - 支持向点播服务获取密钥：HLS加密的原始密钥是存储在点播服务中的，因此，密钥管理服务需要调用点播服务API，获取对应媒资的密钥。
  - 支持缓存获取的密钥：为避免每次都从点播服务获取密钥，密钥管理服务应该具备缓存功能，将获取的密钥进行缓存。
- Token生成服务：当终端用户登录您的播放终端时，您的业务侧服务应该对终端用户的合法性进行校验，生成对应的Token，将带Token的播放地址返回播放器端。Token生成示例代码请参见**示例代码**。

  生成的Token需要包含大写字母、小写字母和数字，长度可自行限制。每一次登录分配一个唯一的Token，且具备时效性。遵循权限最小化原则，建议仅将该Token用于HLS加密视频的场景。

## 视频加密

**步骤1** 上传待加密的视频文件。

若待加密的视频还未上传，可以通过**控制台上传**等方式上传到点播服务中。

**步骤2** 配置获取解密密钥URL。

加密前，需要将**搭建相关服务**中搭建的密钥管理服务的地址配置到点播服务中，加密时，将把该地址写入转码生成的m3u8文件中。

1. 登录视频点播控制台，在左侧导航树中选择“全局设置 > 安全设置”，进入安全设置页面。
2. 单击“HLS加密设置”，在弹出框中输入搭建的密钥管理服务的URL。

   示例：https://domain-sample/encrypt/get-key
3. 单击“确定”，完成配置。

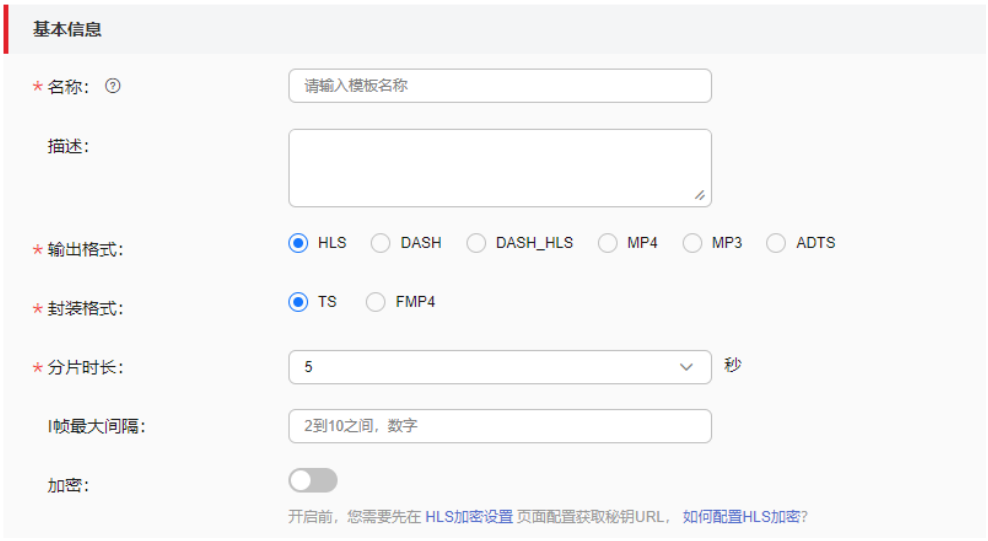
**步骤3** 创建转码模板。

HLS加密是通过转码来实现的，所以在转码前，需要创建开启加密的转码模板。

1. 在左侧导航树中选择“全局设置 > 转码设置”，进入转码设置页面。
2. 单击“自定义转码模板组”，在新建议转码模板页面配置相关参数。

**图 4-1** 设置基本信息




基本信息中“输出格式”选择“HLS”，打开“加密”开关，其它参数可以根据实际需求配置，具体可参考**转码设置**。

3. 单击“确定”，完成HLS转码模板的配置。

**步骤4** 视频加密。

1. 在左侧导航树中，选择“音视频管理”，进入音视频管理页面。
2. 勾选需要HLS加密的视频，单击“转码”。
3. 在弹出框中选择**步骤3**中创建的转码模板，单击“确定”。
   视频开始转码，当转码状态为“转码成功”时表示转码完成，即HLS加密完成。

**----结束**

## 视频播放

由于本方案采用的HLS标准加密，对于支持HLS协议播放的播放器都可以进行解密播放。

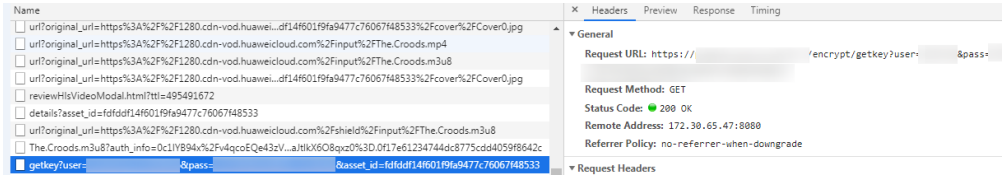
1. 登录视频点播控制台，在左侧导航树中选择“音视频管理”，进入音视频管理页面。
2. 在已经加密的视频行单击“管理”，选择“播放地址”页签。
3. 在对应的HLS格式行单击，播放HLS视频。

**图 4-2** 播放地址


4. 打开浏览器的开发者模式，可以看到控制台在预览播放时有自动通过配置的获取密钥URL去请求密钥，并解密播放。

**图 4-3** 浏览器开发者模式




## 示例代码

- 密钥管理服务示例代码

  示例中采用UUID方式生成Token，您也可以自行选择生成方式。此外，示例代码中未包含登录终端用户的合法性校验，若有需要，您也可以自行实现。

  密钥管理服务收到密钥请求时，会先查看缓存中是否保存该媒资的解密密钥，若没有，则调用点播服务端SDK查询密钥。

```
import java.util.Base64;
import java.util.UUID;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang3.StringUtils;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.huawei.kms.initvodclient.VodClientFactory;
import com.huawei.kms.util.CacheUtils;
import com.huaweicloud.sdk.vod.v1.model.ShowAssetCipherRequest;

import retrofit2.http.Header;

@RestController
public class KeyManagerController {
    /**
     * 给合法客户端分配token并返回带token的播放url
     *
     * @param accessToken 终端携带的鉴权信息，这里需要校验身份信息是否合法
     * @param playUrl     播放url
     * @return 返回带token的播放url
     */
    @GetMapping("/get-url")
    public String getTokenPlayUrl(@Header("access-token") String accessToken,
            @RequestParam(value = "play_url", required = true) String playUrl) {
        // 为合法终端分配token, 这里"*****"需要客户侧代码生成
```

```
        String token = "*****";
        // 构造带token的play_url并返回, http://{domain}/asset/{asset_id}/play_video/index.m3u8?
token={token}
        return playUrl.substring(0, playUrl.lastIndexOf("/") + 1) +
                playUrl.substring(playUrl.lastIndexOf("/") + 1) + "?token=" + token;
    }


    /**
     * @param asset_id 媒资id
     * @param token    给终端分配的token，这里需要校验token是否合法，只给校验通过的终端返回密钥
     * @param response
     * @return 返回字节数组类型的密钥
     */
    @GetMapping(value = "/get-key",headers = "Accept=application/octet-stream")
    public byte[] getKey(@RequestParam(value = "asset_id", required = true) String asset_id,
            @RequestParam(value = "token", required = true) String token, HttpServletResponse
response) {
        // 获取密钥，先从缓存中获取，假如不存在，再从点播服务获取。这里以本地缓存为例，用户可以自
行选择缓存方式，比如存在缓存数据库
        String key = CacheUtils.getCipherFromCache(asset_id);
        if (StringUtils.isEmpty(key)) {
            ShowAssetCipherRequest request = new ShowAssetCipherRequest();
            request.withAssetId(asset_id);
            key = VodClientFactory.getClient().showAssetCipher(request).getDk();
            // 跨域放通，填写实际的站点或填写“*”
            response.setHeader("Access-Control-Allow-Origin", "*");
            response.setHeader("Content-Length", "16");
            // 设置返回密钥的数据类型
            response.setHeader("Content-Type", "application/octet-stream");
            // 更新缓存
            CacheUtils.updateCipherFromCache(asset_id, key);
        }
        return Base64.getDecoder().decode(key);
    }
}
```

- 获取VodClient示例代码

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.vod.v1.VodClient;
import com.huaweicloud.sdk.vod.v1.region.VodRegion;

public class VodClientFactory {

    private final static String AK = System.getenv("CLOUD_SDK_AK");
    private final static String SK = System.getenv("CLOUD_SDK_SK");
    private final static String REGION="cn-north-4"; // 服务实际节点，如cn-north-1，cn-east-2

    private static volatile VodClient vodClient = null;

    public static VodClient getClient() {
        if (vodClient == null) {
            synchronized (VodClient.class) {
                if (vodClient == null) {
                    ICredential auth = new BasicCredentials()
                            .withAk(AK)
                            .withSk(SK);
                    vodClient = VodClient.newBuilder().withCredential(auth)
                            .withRegion(VodRegion.valueOf(REGION))
                            .build();
                }
            }
        }
        return vodClient;
    }
}
```

- 缓存示例代码

密钥管理服务从VOD中获取到解密密钥后，需要将密钥缓存下来，避免同一媒资重复请求VOD获取。示例中采用本地缓存方式，您也可以选择数据库方式缓存。

```
import com.google.common.cache.Cache;
import com.google.common.cache.CacheBuilder;

import java.util.concurrent.TimeUnit;

public class CacheUtils {

    private static Cache<String, String> cipherCache = CacheBuilder.newBuilder()
            .maximumSize(100) // 设置缓存的最大容量
            .expireAfterWrite(10, TimeUnit.MINUTES) // 设置缓存在写入一分钟后失效
            .concurrencyLevel(10) // 设置并发级别为10
            .recordStats() // 开启缓存统计
            .build();

    public static String getCipherFromCache(String key) {
        return cipherCache.getIfPresent(key);
    }

    public static void updateCipherFromCache(String key, String value) {
        cipherCache.put(key, value);
    }
}
```

- 上述示例代码所需的Maven依赖，如下所示：

  注：以下使用的jar包版本非固定，以JAVA项目以及jar包的实际情况为准。

```
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.3.12.RELEASE</version>
    <relativePath/>
</parent>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-lang3</artifactId>
        <version>3.7</version>
    </dependency>
    <dependency>
        <groupId>com.huaweicloud.sdk</groupId>
        <artifactId>huaweicloud-sdk-vod</artifactId>
        <version>3.1.72</version>
    </dependency>
    <dependency>
        <groupId>com.squareup.retrofit2</groupId>
        <artifactId>retrofit</artifactId>
        <version>2.5.0</version>
    </dependency>
    <dependency>
        <groupId>com.google.guava</groupId>
        <artifactId>guava</artifactId>
        <version>27.0.1-jre</version>
    </dependency>
</dependencies>
```

# 5 修改记录

| 发布日期 | 修订记录 |
| --- | --- |
| 2020-08-30 | 第二次正式发布。<br>本次更新说明如下：<br>● 新增“上传音视频文件至VOD”章节。 |
| 2019-03-30 | 第一次正式发布。 |