

视频点播解决方案

1. HLS 标准加密

我们采用的的 HLS 的标准加密，在使用用 HLS 标准加密时，需要申请如下账号。

AccessKey Id: 用于标识用户

AccessKey Secret: 用户用于加密签名字符串和 VOD 用来验证签名字符串的密钥

HLS 标准加密转码: 转码模版

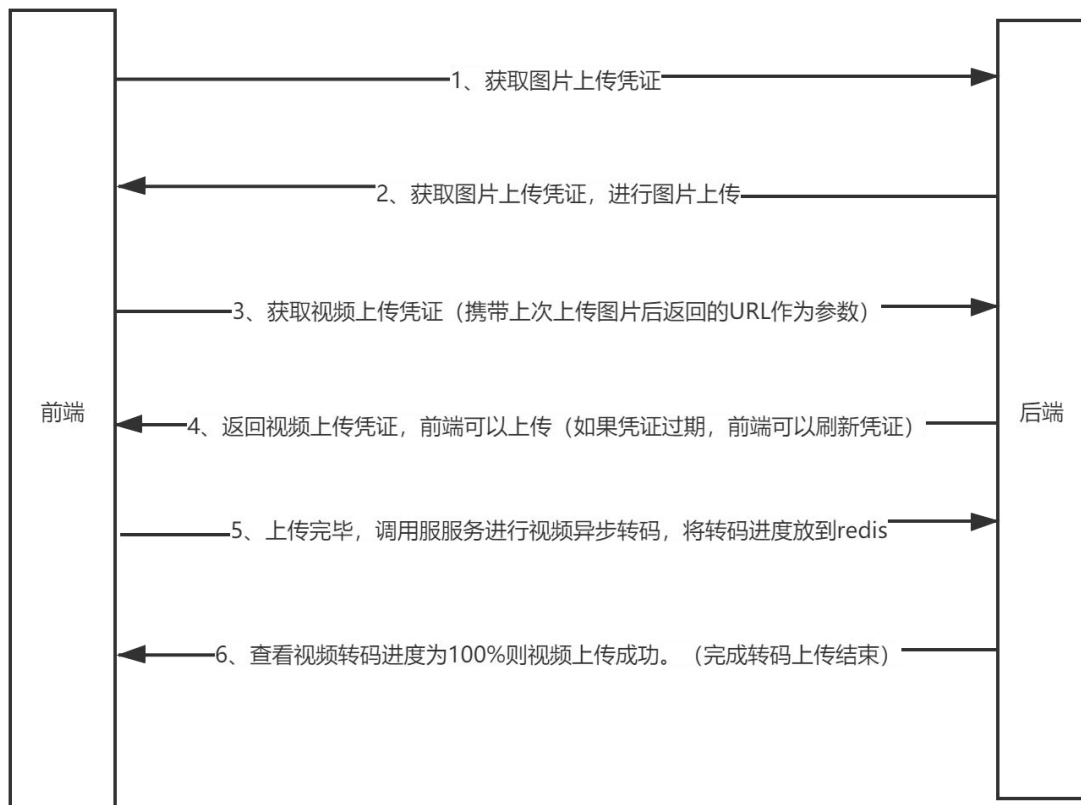
Service key :秘钥管理服务的一种加密主 Key

备注: HLS 标准加密转码和 service key 可以参考

https://help.aliyun.com/document_detail/68612.html?spm=a2c4g.11186623.6.686.438a65ddwgwjSx 来申请

2. 视频上传

- 视频上传整体逻辑



视频转码过程：

首先通过调用“密钥管理服务”获取生成密文密钥，然后构建转码参数进行请求。构建参数的时候可以设置回调地址，这样视频播放的时候，回调我方服务时，我们可以进行权限校验。这个回调是视频播放器自己发起的，所以需要在发起回调之前，给请求中增加对应的token，现在前端增加了拦截器，拦截所有请求，然后发现对应的回调接口将设置对应的token。

转码参数：

https://help.aliyun.com/document_detail/68570.html?spm=a2c4g.11186623.2.65.19ce2334XA1CCZ

EncryptConfig 参数解释：

https://help.aliyun.com/document_detail/86952.html?spm=5176.10695662.1996646101.searchclickresult.ce0b14253wRdiW

视频转码请求代码

```
1.
2. public SubmitTranscodeJobsResponse transcodeVideo(AliyunTask aliyunTask, String
transCode) throws ClientException {
3.     SubmitTranscodeJobsRequest request = new SubmitTranscodeJobsRequest();
4.     //需要转码的视频 ID
5.     request.setVideoId(aliyunTask.getFileId());
6.     //转码模板 ID
7.     request.setTemplateGroupId(transCode);
```

```

8. //构建标准加密配置参数(只有标准加密才需要构建)
9. JSONObject encryptConfig = buildEncryptConfig(aliClient, aliyunTask);
10. //HLS 标准加密配置(只有标准加密才需要传递)
11. request.setEncryptConfig(encryptConfig.toJSONString());
12. return aliClient.getAcsResponse(request);
13. }
14.
15. private JSONObject buildEncryptConfig(DefaultAcsClient client, AliyunTask aliyunTask)
throws ClientException {
16.     //点播给用户在 KMS(密钥管理服务)中的 Service Key，可在用户密钥管理服务对应的
    区域看到描述为 vod 的 service key
17.     String serviceKey = "5ddf9f15-0a94-4e7f-9882-7c6527cfc3c6";
18.     //随机生成一个加密的密钥，返回的 response 包含明文密钥以及密文密钥，
19.     //视频标准加密只需要传递密文密钥即可
20.     GenerateDataKeyResponse response = generateDataKey(client, serviceKey);
21.     JSONObject encryptConfig = new JSONObject();
22.     //解密接口地址，该参数需要将每次生成的密文密钥与接口 URL 拼接生成，表示
    每个视频的解密的密文密钥都不一样
23.     //至于 Ciphertext 这个解密接口参数的名称，用户可自行制定，这里只作为参考参
    数名称
24.     encryptConfig.put("DecryptKeyUri",
"http://edufont.lagou.com/front/course/media/alikey?" + "code=" + response.getCiphertextBlob()
+ "&vid=" + aliyunTask.getFileId());
25. //     encryptConfig.put("DecryptKeyUri", "http://localhost:8080/alikey?" + "code=" +
response.getCiphertextBlob() + "&vid=" + vid);
26.     //密钥服务的类型，目前只支持 KMS
27.     encryptConfig.put("KeyServiceType", "KMS");
28.     //密文密钥
29.     encryptConfig.put("CipherText", response.getCiphertextBlob());
30.     aliyunTask.setFileEdk(response.getCiphertextBlob());
31.     aliyunTask.setFileDk(response.getPlaintext());
32.     return encryptConfig;
33. }

```

3. 视频播放

