

视频处理 用户指南 产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

用户指南

控制台操作指南

DEMO使用指南

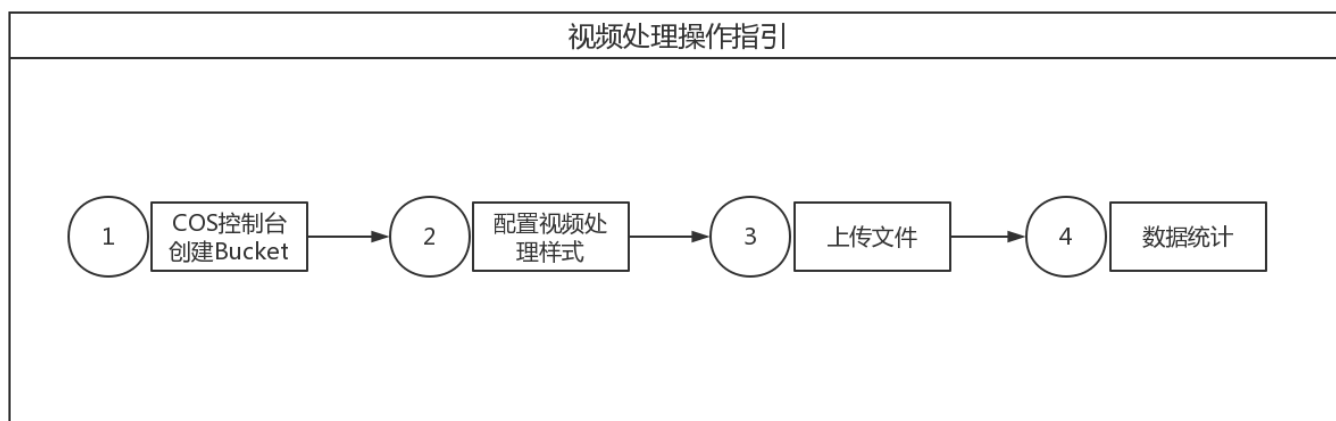
用户指南

控制台操作指南

最近更新时间：2018-05-28 15:55:43

1 操作指引

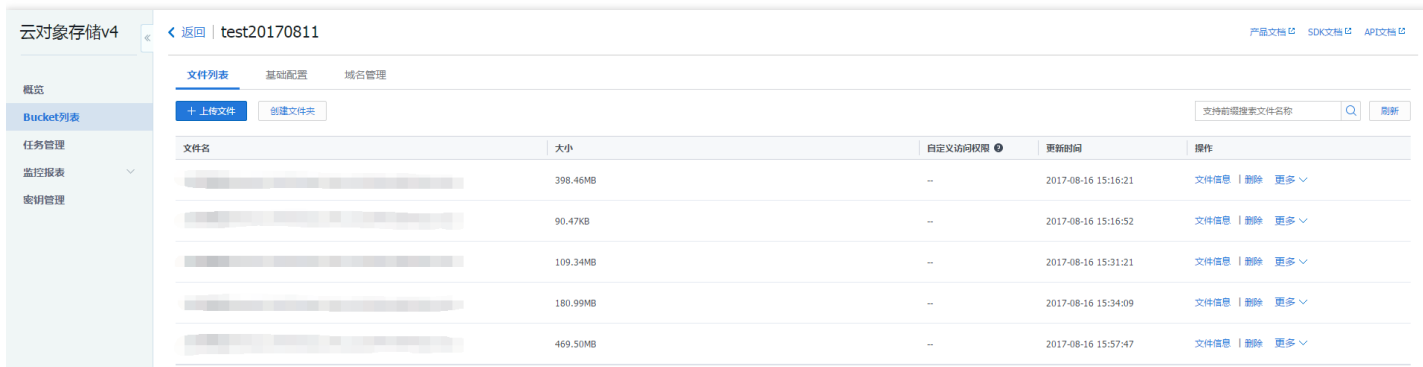
开发者可按照视频处理的控制台指引，通过[COS 控制台](#)创建存储桶（Bucket）后，在视频处理控制台配置好相应处理规则，使用COS 提供的SDK 上传管理相应文件即可。上传后，视频类文件将通过预置规则进行处理。



2 存储桶（Bucket）

存储桶（Bucket）是视频资源的组织管理单元。开发者可以为一个项目创建多个空间（上限200个），用以存储不同类型的视频，同时一个项目下的多个空间共用一套鉴权服务。

用户可以在[COS 控制台-Bucket列表](#)，对上传的资源进行管理：



2.1 Bucket 管理

在[视频处理控制台-Bucket 管理](#)页面中，您可以对视频处理规则进行一系列的设置。



2.1.1 视频处理状态

视频处理状态开关打开时，上传到COS 的文件会按照预设的所有输出规则进行转码。



2.1.2 视频转码

视频处理服务提供以下三种转码服务，设置完毕后上传的视频会做相应的转码处理，转码时长和原视频的时长成正比。

输出规则

输出格式 MP4

- 当前规则
- 高清720：设定分辨率为720p，宽等比缩放，码率1024k，帧率30
 - 标清480：设定分辨率为480p，宽等比缩放，码率512k，帧率25
 - 低清240：设定分辨率为240p，宽等比缩放，码率256k，帧率25

保存

取消

支持的视频转码格式包括：（如有特殊需求，请联系客服或提交工单）

flv , m4v , mov , mp4 , m4a , 3gp , 3g2 , mpegts , rm , wav , avi , rmvb , mpeg , mjpeg

2.1.3 水印设置

视频处理提供为视频打水印的功能，开发者可以自己上传水印图片，指定好水印在视频中的相对位置，保存成功后，视频会在上传后进行打水印操作。

注意：视频处理的水印暂时是全局的，针对一个项目只能设置一个，同一项目下不同bucket设置，只以最后一次的

设置为准，此处的优化工作我们会尽快完成，给您带来的不便敬请谅解。

水印设置

状态

水印类型 图片水印 动态水印

水印图片



X轴偏移百分比
填写小数范围【0~100】，代表水平偏移量与左上角位置的比率。

Y轴偏移百分比
填写小数范围【0~100】，代表垂直偏移量与左上角位置的比率。

开启后，视频转码时会自动按设置规则添加水印。

2.1.4 输出格式

视频处理现在暂时只支持mp4格式的输出。

2.1.5 封面设置

当开启本开关时，视频在上传到视频空间后，取视频非黑屏的第一帧作为视频封面。

如果用户不希望系统自动生成封面，而是自己指定封面，则需要使用restful api 或视频处理SDK 设置。需要注意的是，默认封面功能依赖转码功能，需要用户在控制台上，同时开启清晰度转码选项。

用SDK自定义封面，则不受转码影响。当用户在开启了默认封面的同时，指定了自定义封面，则系统不会为视频生成默认封面。

系统会为用户自动创建一个系统Bucket：systemcover，用来存储为视频生成的默认封面的图片。为了不影响功能，建议用户不要更改systemcover 的设置。systemcover 产生的存储和流量，和视频同等标准计费。

封面设置

状态

1. 输出规则至少选择一项，封面设置才会生效
2. 开启后当前Bucket视频使用非黑屏的第一帧做封面

保存

取消

2.1.6 回调设置

开发者设置转码完成回调url，如下图：

通知回调URL 编辑

状态

回调URL

1. 视频转码完成后，会将处理结果状态提交到通知回调URL的网址
2. 回调URL需以http开头且默认返回200正确码方可使用

保存

取消

设置成功，在视频转码完成后，视频处理会向该url发送以下格式的http请求。

HTTP包信息如下表：

参数名称	类型	必选	描述
playurl	String	是	原视频的播放url
vid	String	否	保留字段，现在传空值
status	int	否	转码状态，0：初始化；1：转码中；2：转码成功；3：转码失败
type	int	否	保留字段
up_time	uint	是	视频上传时间，UNIX时间戳
cover_url	String	否	视频封面url地址
desc	String	否	视频描述

参数名称	类型	必选	描述
title	String	否	视频标题
magic_context	String	否	上传视频时携带的透传参数，如果业务有较复杂的参数，建议整理成json格式

接口样例如下：

+POST http://xx.yy.com/

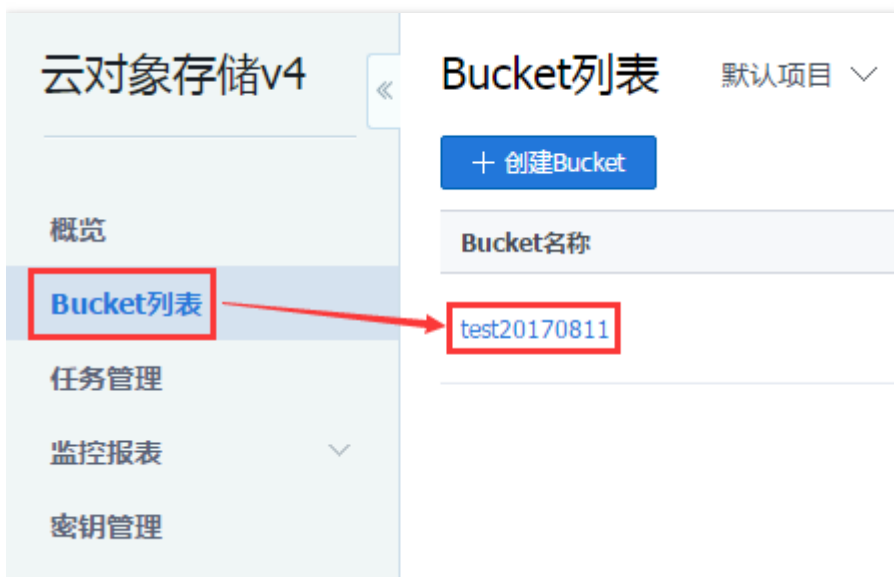
playurl=playurl&vid=vid&status=status&type=type&up_time=up_time&cover_url=cover_url&desc=desc&title=title&magic_context=magic_context

3 使用统计

腾讯云·视频处理提供针对单个应用其存储量、下载流量、上传下载删除的次数等信息的使用量统计数据，数据会有一定延迟，延迟时间在20分钟以内。如需使用请到[视频处理-使用统计](#)中查看。

4 其他

域名设置、防盗链设置、目录管理相关设置、服务质量监控相关请前往[COS 控制台](#)进行配置：



项目密钥配置相关，请前往官网[云API 密钥](#)页面进行配置。

DEMO使用指南

最近更新时间：2018-10-08 17:17:20

以下 Demo 均为旧版微视频 Demo，只适用于从微视频升级至视频处理的用户！

新用户请直接参考 [COS iOS Demo](#) 和 [Android Demo](#)。

1 视频处理 DemoAPP 简介及下载

1.1 视频处理 DemoAPP 简介

视频处理 DemoAPP，是腾讯云提供给开发者的一个测试 APP，包括图片（数据万象）和视频（视频处理）两个体验模块。图片体验模块包括图片的上传、暂停、删除、查询和复制等功能；视频体验模块包括视频的上传、暂停、查询和删除等功能。开发者可以使用 DemoAPP 体验腾讯云视频处理服务，也可以在 DemoAPP 上注册自己的 APP 来验证其视频处理服务是否正常开通。

1.2 视频处理 DemoAPP 下载

Android 和 iOS 均参考 [SDK 下载](#)。

2 视频处理 DemoAPP 使用说明

2.1 注册 APP

DemoAPP 本身自带了官网内置的 APP 应用信息，可以直接使用上传、暂停、删除、查询、复制等操作来体验视频处理服务。如果开发者希望验证自己的视频处理服务是否开通，可以在 DemoAPP 上注册自己的 APP，如下图所示，其中

- (1). APPID 为开发者在 [移动服务控制台——应用管理](#) 创建应用时获取的 AppID；
- (2). USERID 是开发者自己业务体系中的 userid，用户可以自己设置也可以使用 DemoAPP 自带的 123456；
- (3). SIGN 为签名。签名是开发者客户端向其业务服务器请求得到的，验证自己的移动是否开通时，开发者可以参考 [服务器端 SDK](#)，选择熟悉的语言 SDK，根据其提供的接口生成签名。



填写注册信息：



2.2 使用上传、暂停、删除、查询、复制等操作

图片服务页面如下：



视频服务页面如下：



2.3 iOS 返回码说明

```

/#!/
@enum TXYUploadRequestResultCode enum
@abstract 上传返回错误码
*/
typedef enum
{
    /*! 上传成功 */
    TXYUploadResult_Success = 0,
    /*! 无效session */
    TXYUploadResult_InvalidSession = -290,
    /*! 解包失败 */
    TXYUploadResult_DecodeError = -20101,
    /*! 服务器连接出错 */
    TXYUploadResult_ConnectError = -20102,
    /*! 数据收发超时 */
    TXYUploadResult_Timeout = -20103,
    /*! 控制包收取超时 */
    TXYUploadResult_Ctrl_Timeout = -20104,
    /*! 命令字不支持 */
    TXYUploadResult_Cmd_Not_Found = -20105,
    /*! 服务器返回偏移错误 */
    TXYUploadResult_Invalid_Offset = -20106,
    /*! 文件为空 */
    TXYUploadResult_File_Empty = -20107,
} TXYUploadRequestResultCode;

```

```

/**
 * Session执行结果
 * [-20001, -20100]
 */
typedef enum {
    USR_SUCCESS = 0, // 成功
    USR_ERROR_CONN_TIMEOUT = -20001, // 链接超时
    USR_ERROR_HOST_NOTREACH = -20002, // 地址不可达
    USR_ERROR_SOCKET_RESET = -20003, // socket被reset
    USR_ERROR_OVERFLOW = -20005,
    USR_ERROR_UNDERFLOW = -20006,
    USR_ERROR_SENDError = -20007,
    USR_ERROR_ADDR = -20008, // 地址有误
    USR_ERROR_NET = -20009, // 网络出错
    USR_ERROR_HS_PACK = -20010, // (打包、解包)握手包有误
    USR_ERROR_HS_TIMEOUT = -20011, // 握手超时
    USR_ERROR_SOCKET_NOMEM = -20013, // Cannot allocate memory
    USR_ERROR_NET_UNREACHABLE = -20014, // Network unreachable
} UpSessionResult;

```

```

//返回错误码分类
typedef enum _DLMNetworkErrorType {
    //文件下载完成校验错误
    DLMWebpDecodeErrorType = -4,           //WEBP解码错误
    DLMInvalidFileLengthType = -3,        //长度不一致
    DLMPicNoNeedCacheType = -2,          //无需缓存的图片
    DLMInvalidPicDataType = -1,          //无效图片格式

    //网络相关错误
    DLMConnectionFailureErrorType = 1,    //连接失败
    DLMRequestTimeoutErrorType = 2,      //请求超时
    DLMAuthenticationErrorType = 3,
    DLMRequestCancelledErrorType = 4,     //请求取消
    DLMUnableToCreateRequestErrorType = 5,
    DLMInternalErrorWhileBuildingRequestType = 6,
    DLMInternalErrorWhileApplyingCredentialsType = 7,
    DLMFileManagementError = 8,
    DLMTooMuchRedirectionErrorType = 9,
    DLMUnhandledExceptionError = 10,
    DLMCompressionError = 11,
    DLMStreamErrorInnerNotRetry = 12,    //遇到'socket not connected', 'broken pipe' or 'connection lost' error类型错误ASI内部不要删除本地临时文件重
        试由外部来处理

    //断点续传校验(以下错误建议业务层把临时数据删除或重置)
    DLMContentTypeNotEqualType = 80001,  //前后两次请求返回文件类型不一致
    DLMRangePositionNotEqualType = 80002, //返回与请求起始点range不匹配
    DLMRangePositionIsZeroType = 80003,  //返回位置为0
    DLMRangeFileLengthNotEqualType = 80004, //起始点加上剩余长度与总长度不匹配
    DLMRangeInvalidStatusCodeType = 80005, //断点续传返回非206状态码
    DLMRangeModifyTimeNotEqualType = 80006, //前后两次请求修改时间不匹配

    //NSURLSession操作相关失败
    DLMUSDownloadLocationInvalid = 90001, //获取下载临时文件失败
    DLMUSCheckFileDataFail = 90002,     //校验文件失败
} DLMErrorType;
    
```

2.4 Android 返回码说明


```
public enum RetCode
{
    SUCCEED(0, "成功"),
    FAST_SUCCEED(1, "秒传成功"),
    CANCELED(2, "任务取消"),
    PAUSED(3, "任务暂停"),
    FILE_NOT_EXIST(4, "文件不存在"),
    FILE_LENGTH_INVALID(5, "文件长度不合法"),
    NETWORK_NOT_AVAILABLE(6, "网络不可用"),
    RESPONSE_IS_NULL(7, "服务器回包为空"),
    REQUEST_TIMEOUT(8, "Request timeout"),

    NO_SESSION(11, "没有可用Session"),
    SESSION_DISCONNECT(12, "Session连接断开"),
    SESSION_STATE_INVALID(13, "Session状态非法"),
    SESSION_DETECT_ERROR(14, "网络探测失败"),

    SESSION_ALL_ROUTE_FAILED(101, "网络通道建立失败"),
    SESSION_CONN_SEND_FAILED(102, "Connection Send Failed"),
    SESSION_WITHOUT_CONN(103, "Connection为空"),
    SESSION_DIVIDE_PACKET_ERROR(104, "接受数据异常, 分包失败"),
    SESSION_SEND_DATA_ENCODE_ERROR(105, "Request encode error"),

    OTHER(1000, "其他");
}
```

```

public class ErrorCode {
    public static final int LOCAL_RET_CODE_UNKNOW = -99999; // 未知错误
    public static final int LOCAL_RET_CODE_TRY_FAILED = -99998; // 超过尝试次数
    public static final int LOCAL_RET_CODE_NETWORK_FAIL = -99997; // 网络请求失败
    public static final int LOCAL_RET_CODE_TYPE_MISMATCH = -99996; // 下载内容并非所需要类型
    public static final int LOCAL_RET_CODE_DECODE_FAILED = -4; // 解码失败
    public static final int LOCAL_RET_CODE_LENGTH_MISMATCH = -3; // 长度与content-length不匹配
    public static final int LOCAL_RET_CODE_NO_CACHE = -2; // no-cache
    public static final int LOCAL_RET_CODE_ILLEGAL_FORMAT = -1; // 非法格式
    public static final int LOCAL_RET_CODE_FNF_EXP = 1; // 本地FileNotFoundException
    // exception
    public static final int LOCAL_RET_CODE_IO_EXP = 2; // 本地IO exception
    public static final int LOCAL_RET_CODE_OOM_ERR = 3; // 本地OutOfMemory error
    public static final int LOCAL_RET_CODE_EXP = 4; // 本地未知exception
    public static final int LOCAL_RET_CODE_ILL_EXP = 5; // 本地IllegalStateException
    public static final int LOCAL_RET_CODE_SOCKET_EXP = 6; // 本地Socket exception
    public static final int LOCAL_RET_CODE_SOCKET_TO_EXP = 7; // 本地Socket
    // Timeout
    // exception
    public static final int LOCAL_RET_CODE_CLIENT_PROTOCOL_EXP = 8; // 本地ClientProtocolException
    public static final int LOCAL_RET_CODE_UNKNOW_HOST_EXP = 9; // 本地UnknowHostException
    public static final int LOCAL_RET_CODE_CONNECT_TIMEOUT_EXP = 10; // 本地ConnectTimeoutException
    public static final int LOCAL_RET_CODE_NO_HTTP_RSP_EXP = 11; // 本地NoHttpResponseException
    public static final int LOCAL_RET_CODE_SSL_PEER_UNVERIFIED_EXP = 12; // 本地SSLPeerUnverifiedException
    public static final int LOCAL_RET_CODE_CONNECT_POOL_TIMEOUT_EXP = 13; // 本地ConnectionPoolTimeoutException
    public static final int LOCAL_RET_CODE_CONNECT_CLOSED_EXP = 14; // ConnectionClosedException
    public static final int LOCAL_RET_CODE_NETWORK_UNAVAILABLE = 15; // 网路不可用
}
    
```