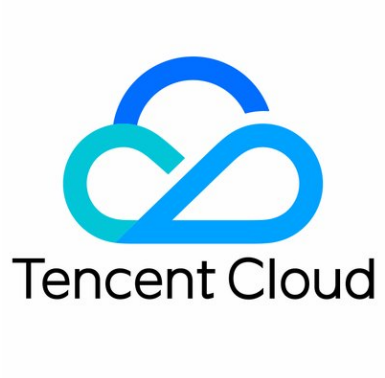
互动白板

开发指南



Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

开发指南
白板推流（推荐）
白板推流概述

Last updated：2024-12-06 14:32:32

功能简介

腾讯云互动白板提供的白板推流服务支持将白板画面推流到实时音视频房间，观看端不需要集成互动白板也能观看到白板画面，再结合实时音视频旁路推流及云端混流的能力，可以把白板画面与实时音视频房间内的其他音视频流按照指定的布局混到一起推到 CDN，观看端通过 CDN 拉流观看。

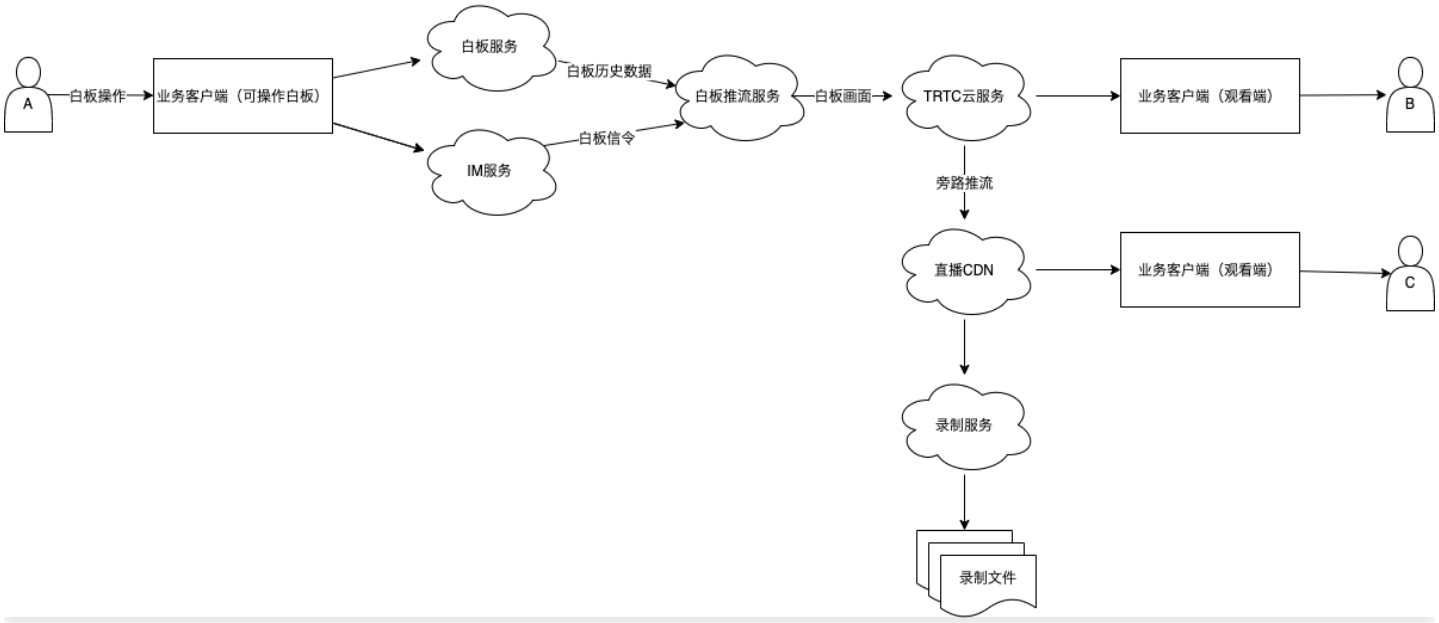
1. 通过白板推流实现观看，具体参考 白板推流观看 。
2. 通过白板推流实现录制，具体参考 白板推流录制 。

白板推流原理

白板推流服务会实时地从白板服务及即时通信（IM）服务获取白板历史数据及白板操作信令，进行白板画面渲染，然后把渲染好的白板画面以主流的方式推送到指定的实时音视频（TRTC）房间内，房间内的成员可以按需对这路白板视频流进行订阅，如果已开启旁路推流，非房间内的用户也可以通过直播 CDN 拉视频流进行观看。

注意：

1. 白板推流（录制）请求成功后，PushUserId（RecordUserId）作为一个推流（录制）用户，会同时加入 RoomId 对应的白板教室、IM 群组、TRTC 房间。
2. 白板房间内操作白板（例如，画笔轨迹、文档翻页、音视频播放等）时，PushUserId 会通过 IM 群组同步接收白板教室内的信令数据，然后渲染成画面推流到 TRTC 房间，推流成功后，在 TRTC 仪表盘上可以看到 PushUserId 对应的数据。
3. RoomId 对应的 IM 群组必须在白板推流开始前创建完成。
4. RoomId 格式必须是数字型参数，对应的 IM 群组需要是数字型字符串（例如 RoomId 为1234，则 IM 群组的 GroupID 为"1234"）。



白板推流工作原理图

白板推流录制

Last updated：2024-09-11 09:08:11

> **注意：**
> 接入白板推流后，白板推流服务会把白板画面推送到指定的音视频房间，如果客户端不需要展示白板推流的画面，请根据白板推流的用户 ID 进行过滤。

交互流程

接入白板推流后，从开始上课到结束上课，一般经过如下几个交互过程：

白板推流

准备 PushUserId 和 PushUserSig

白板推流服务需要通过 IM 服务实时拉取白板操作信令进行白板渲染，同时需要进入到实时音视频房间内把白板画面以主路流的方式推送到房间内，以便其他用户进行流订阅，因此需要您提供一个白板推流服务进房时使用的 `PushUserId` 以及 `PushUserSig` ，生成 `PushUserId` 和 `PushUserSig` 的方法请参考 如何计算 UserSig 。

> **注意：**
> `PushUserSig` 签名请设置一个较长的有效期，至少比课堂时长要长，避免由于签名过期导致白板推流中断的情况。

开始白板推流

在需要进行白板推流时，例如老师学生都已经准备好开始上课，您可以使用 开始白板推流 接口来通知白板推流服务开始白板推流，在请求接口时，需要使用到上一步准备好的 `PushUserId` 和 `PushUserSig` 。
白板推流开始后，会有 推流开始 事件回调，建议提前注册好事件回调监听，具体注册方式参考 事件通知综述 。

> **说明：**
> 由于网络延迟等因素，发送请求后，实际白板推流操作将在2s左右后进行。
> 目前服务端 API 接口只支持区域广州，在调用 API 时，Region 参数请填写 `ap-guangzhou` 。

停止白板推流

在课堂结束或者需要停止白板推流的时候，您可以使用 结束白板推流 接口通知白板推流服务停止当前推流。
白板推流开始后，会有 推流结束 事件回调，建议提前注册好事件回调监听，具体注册方式参考 事件通知综述 。

> **说明：**
> 由于网络延迟等因素，发送请求后，实际白板推流操作将在2s左右后进行。
> 目前服务端 API 接口只支持区域广州，在调用 API 时， `Region` 参数请填写 `ap-guangzhou` 。

白板混流

> **注意：**
> 白板混流依赖实时音视频的云端混流转码功能，使用云端混流功能将产生相应的费用，具体计费规则请参考 云端混流转码计费说明 。

通常情况下，白板推流与房间内的其他音视频流都是独立的音视频流，观看端需要分别订阅各路流，并按照布局来进行视频播放，同时云端录制录下来的视频也是一个个独立的视频，这大大提高了客户端开发以及后续回放视频生成的复杂度。
通过把白板推流与实时音视频云端混流的能力结合起来，可以实现白板视频流与房间内其他音视频流混合成一路，从而达到我们期望的效果，同时降低客户端与录制回放的复杂度。
云端混流 golang 版本代码示例：

```
package main

import (
    "fmt"

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    trtc "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/trtc/v20190722"
```

```
)

func main() {

    // 这里需要填上您的真实SecretID和SecretKey
    credential := common.NewCredential(
        "您的SecretID",
        "您的SecretKey",
    )
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "trtc.tencentcloudapi.com"
    client, _ := trtc.NewClient(credential, "ap-guangzhou", cpf)

    request := trtc.NewStartMCUMixTranscodeRequest()

    // 指定输出参数，决定混流后的视频流去向，如果旁路推流和云端录制选择的是全局自动，则可以忽略这里的参数。
    request.OutputParams = &trtc.OutputParams{
        // 在OutputParams中指定StreamId，表示这路混流视频需要旁路推流，这里的字符串可自行定义。
        StreamId: common.StringPtr("1400000001_880528_mix_stream"),
        // 在OutputParams中指定RecordId，表示这路混流视频需要录制，这里的字符串可自行定义。
        RecordId: common.StringPtr("1400000001_880528_mix"),
    }

    // 指定混流音视频编码参数，可以按需自行调整
    request.EncodeParams = &trtc.EncodeParams{
        AudioSampleRate: common.Uint64Ptr(48000),
        AudioChannels:   common.Uint64Ptr(1),
        AudioBitrate:    common.Uint64Ptr(50),
        VideoWidth:      common.Uint64Ptr(1280),
        VideoHeight:     common.Uint64Ptr(720),
        VideoBitrate:    common.Uint64Ptr(500),
        VideoFramerate:  common.Uint64Ptr(20),
        VideoGop:        common.Uint64Ptr(3),
    }

    // 指定混流布局，这里选择了预定义布局模板2（屏幕分享模板），并设置大屏显示白板推流视频。
    // 这里假设白板推流的用户ID为"tic_push_user_880528_test"，使用的时候请修改为真实的白板推流用户ID
    request.LayoutParams = &trtc.LayoutParams{
        Template:            common.Uint64Ptr(2),
        MainVideoUserId:     common.StringPtr("tic_push_user_880528_test"),
        MainVideoStreamType: common.Uint64Ptr(0),
    }

    // 这里请修改为真实的SdkAppID及RoomID
    request.SdkAppId = common.Uint64Ptr(1400000001)
    request.RoomId = common.Uint64Ptr(880528)

    response, err := client.StartMCUMixTranscode(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        fmt.Printf("An API error has returned: %s", err)
        return
    }
    if err != nil {
        panic(err)
    }
    fmt.Printf("%s", response.ToJsonString())
}
```

实时音视频云端混流的具体使用方法可以参考相关指引文档 云端混流转码 。

观看白板推流

> ⚠ 注意
> 观看白板推流会产生相应的观看费用，具体计费规则可以参考 视频通话计费说明 和 CDN 旁路直播观看费用说明 。

白板推流服务负责把白板画面以主路流的方式推送到指定的音视频房间内，观看端想要对这路流进行观看的话，通常有两种方式：

- **进入房间实时观看**：观看端集成实时音视频 SDK，进入到相同的音视频房间，通过 SDK API 对相应的流进行订阅。具体实现可以参考相关指引文档 导入 SDK 到项目中 。
- **通过 CDN 旁路直播观看**：开通旁路推流，通过直播 CDN 拉流观看。 具体实现可以参考相关指引文档 实现 CDN 直播观看 。

白板推流观看

Last updated：2024-12-06 15:57:22

> **注意**
> 白板推流录制依赖实时音视频的云端录制，开启录制会产生相应的录制费用，具体计费规则参考 云端录制计费说明 。

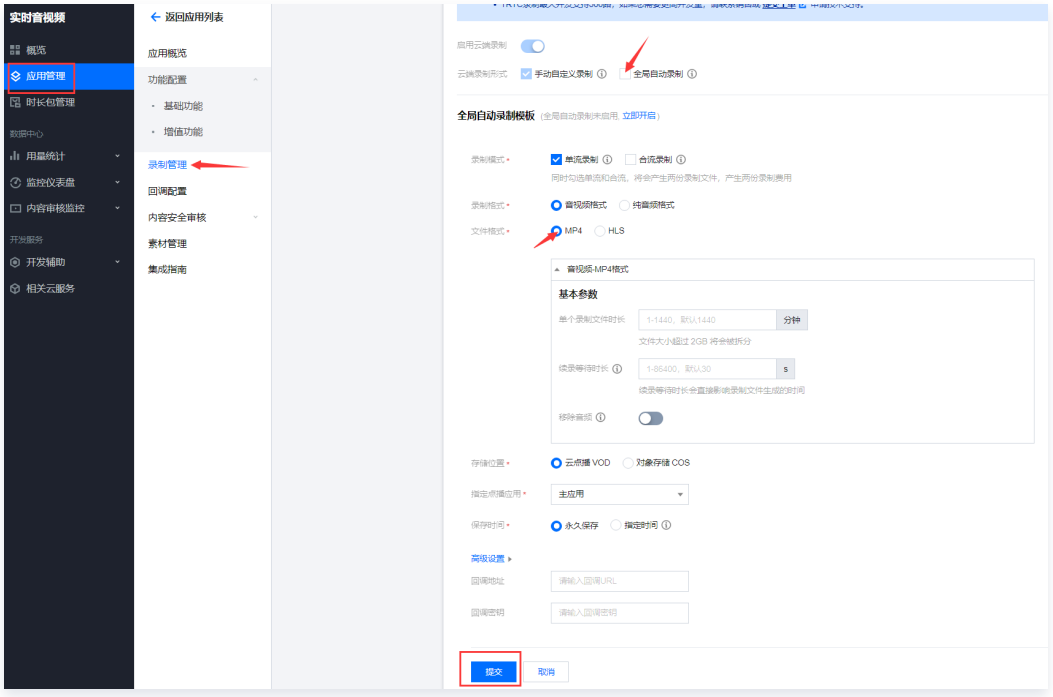
白板推流录制

白板推流服务本身不提供录制功能，不过可以结合实时音视频服务的云端录制功能来进行白板推流录制。
实时音视频云端录制提供了两种录制模式：**全局自动录制**和**指定用户录制**。

全局自动录制（推荐）

在实时音视频云端录制的**全局自动录制**模式下，每一个 TRTC 房间中的每个用户的音视频上行流都会被自动录制下来，录制任务的启动和停止都是自动的，不需要额外操心，比较简单和易用。白板推流作为房间内的一路上行流，同样会被自动录制下来。
全局自动录制配置方法：

1. 登录 实时音视频控制台 ，在左侧导航栏选择 **应用管理**。
2. 单击**应用管理**下的**录制管理**，单击**全局自动录制**，会显示**全局自动录制模板**。
3. 在**全局自动录制模板**设置页面中, 录制文件格式选择 MP4，其他设置按实际情况来选择。
4. 单击下方的**提交**，再次单击勾选**全局自动录制**。



5. 新建的应用：开通实时音视频云端录制的更多设置内容可以参考实时音视频的 实现云端录制与回放 页面。

指定用户录制

如果您需要精细的控制每一路流的录制情况，例如，哪一路流需要录制，哪一路流不需要录制，那么选择**指定用户录制**模式是比较适合这种场景需求的。
指定用户录制 模式的配置方式与**全局自动录制**模式的配置方式类似，只是在云端录制形式选择的时候选择 **指定用户录制** 选项即可。
在选择了**指定用户录制**模式的情况下，那么想要完成录制，需要做一些额外的工作：

1. 如果需要录制用户单流视频，在用户进房的时候，通过客户端 SDK API，指定 userDefineRecordId ，表示这个用户的视频流需要录制。

   Android 端代码示例：

```
mTRTCCloud = TRTCCloud.sharedInstance(getApplicationContext());
```

```
mTRTCCloud.setListener(new TRTCCloudImplListener(this));

// 初始化配置 SDK 参数
TRTCCloudDef.TRTCParams trtcParams = new TRTCCloudDef.TRTCParams();
trtcParams.sdkAppId = GenerateTestUserSig.SDKAPPID;
trtcParams.roomId = Integer.parseInt(mRoomId);

// 这里需要修改为真实的UserID和UserSig
trtcParams.userId = "进房UserID";
trtcParams.userSig = "对应的UserSig";

// 指定录制ID，在用户进房的时候会进行录制，需要修改为实际希望的字符串
trtcParams.userDefineRecordId = "xxx";

// 以主播角色身份进入房间
trtcParams.role = TRTCRoleAnchor;

// 进入通话
mTRTCCloud.enterRoom(trtcParams, TRTC_APP_SCENE_VIDEOCALL);

// 开启本地声音采集并上行
mTRTCCloud.startLocalAudio();
// 开启本地画面采集并上行
mTRTCCloud.startLocalPreview(mIsFrontCamera, mLocalPreviewView);

// 设置上行视频编码参数
TRTCCloudDef.TRTCVideoEncParam encParam = new TRTCCloudDef.TRTCVideoEncParam();
encParam.videoResolution = TRTCCloudDef.TRTC_VIDEO_RESOLUTION_640_360;
encParam.videoFps = Constant.VIDEO_FPS;
encParam.videoBitrate = Constant.RTC_VIDEO_BITRATE;
encParam.videoResolutionMode = TRTCCloudDef.TRTC_VIDEO_RESOLUTION_MODE_PORTRAIT;
mTRTCCloud.setVideoEncoderParam(encParam);
```

2. 如果需要录制白板推流单流视频，在 开始白板推流 的时候，将 `AutoRecord` 参数指定为 `true`，表示白板推流这路视频流需要录制。

golang 版本代码示例：

```
package main

import (
    "fmt"

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    tiw "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tiw/v20190919"
)

func main() {
    // 这里需要填上您的真实SecretID和SecretKey
    credential := common.NewCredential(
        "您的SecretID",
        "您的SecretKey",
    )
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "tiw.tencentcloudapi.com"
```

```
    client, _ := tiw.NewClient(credential, "ap-guangzhou", cpf)

    request := tiw.NewStartWhiteboardPushRequest()

    // 这里请修改为真实的SdkAppID及RoomID
    request.SdkAppId = common.Int64Ptr(1400000001)
    request.RoomId = common.Int64Ptr(880528)

    // 这里请修改为真实的UserID及UserSig
    request.PushUserId = common.StringPtr("tic_push_user_880528_test")
    request.PushUserSig = common.StringPtr("对应的UserSig")

    // 这里配置白板视频的宽高及白板初始化参数
    request.Whiteboard = &tiw.Whiteboard{
        Width:     common.Int64Ptr(1280),
        Height:    common.Int64Ptr(720),
        InitParam: common.StringPtr("{\"ratio\": \"16:9\"}"),
    }

    // 白板推流提供了备份推流能力，如果需要推一个备份流的话，需要在Backup参数中提供另外一个UserID和UserSig
    request.AutoManageBackup = common.BoolPtr(true)
    request.Backup = &tiw.WhiteboardPushBackupParam{
        PushUserId:  common.StringPtr("tic_push_user_880528_test_backup"),
        PushUserSig: common.StringPtr("xxxx"),
    }

    // 设置自动停止推流超时时间，这里设置300，表示超过300s没有人操作白板，则自动停止白板推流
    request.AutoStopTimeout = common.Int64Ptr(300)

    // 设置AutoRecord为true，表示实时音视频的云端录制需要录制白板推流视频
    request.AutoRecord = common.BoolPtr(true)

    response, err := client.StartWhiteboardPush(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        fmt.Printf("An API error has returned: %s", err)
        return
    }
    if err != nil {
        panic(err)
    }
    fmt.Printf("%s", response.ToJsonString())
}
```

3. 如果需要录制 MCU 混流视频，在 启动云端混流 的时候，指定 OutputParams.RecordId，表示这路混流视频需要进行录制。
golang 版本代码示例：

```
package main

import (
    "fmt"

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    trtc "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/trtc/v20190722"
```

```
)

func main() {

    // 这里需要填上您的真实SecretID和SecretKey
    credential := common.NewCredential(
        "您的SecretID",
        "您的SecretKey",
    )
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "trtc.tencentcloudapi.com"
    client, _ := trtc.NewClient(credential, "ap-guangzhou", cpf)

    request := trtc.NewStartMCUMixTranscodeRequest()

    // 在OutputParams中指定RecordId，表示这路混流视频需要录制，这里的字符串可以修改为想要的字符串
    request.OutputParams = &trtc.OutputParams{
        RecordId: common.StringPtr("1400000001_880528_mix"),
    }

    // 指定混流音视频编码参数，可以按需自行调整
    request.EncodeParams = &trtc.EncodeParams{
        AudioSampleRate: common.Uint64Ptr(48000),
        AudioChannels:   common.Uint64Ptr(1),
        AudioBitrate:    common.Uint64Ptr(50),
        VideoWidth:      common.Uint64Ptr(1280),
        VideoHeight:     common.Uint64Ptr(720),
        VideoBitrate:    common.Uint64Ptr(500),
        VideoFramerate:  common.Uint64Ptr(20),
        VideoGop:        common.Uint64Ptr(3),
    }

    // 指定混流布局，这里选择了预定义布局模板2（屏幕分享模板），并设置大屏显示白板推流视频。
    // 这里假设白板推流的用户ID为"tic_push_user_880528_test"，使用的时候请修改为真实的白板推流用户ID
    request.LayoutParams = &trtc.LayoutParams{
        Template:            common.Uint64Ptr(2),
        MainVideoUserId:     common.StringPtr("tic_push_user_880528_test"),
        MainVideoStreamType: common.Uint64Ptr(0),
    }

    // 这里请修改为真实的SdkAppID及RoomID
    request.SdkAppId = common.Uint64Ptr(1400000001)
    request.RoomId = common.Uint64Ptr(880528)

    response, err := client.StartMCUMixTranscode(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        fmt.Printf("An API error has returned: %s", err)
        return
    }
    if err != nil {
        panic(err)
    }
    fmt.Printf("%s", response.ToJsonString())
}
```

实时音视频云端录制的具体配置及使用方法，请参见 实现云端录制与回放 。

文档转码（推荐）
文档转码概述

Last updated：2024-09-11 09:08:12

腾讯云文档转码服务为您提供了将文件 转码为 **HTML5 页面和图片**的能力，将转码后的文档展示于白板，为您提供和线下教育体验高度一致的在线教育服务。

支持转码格式

详细可参考 文档制作规范 。

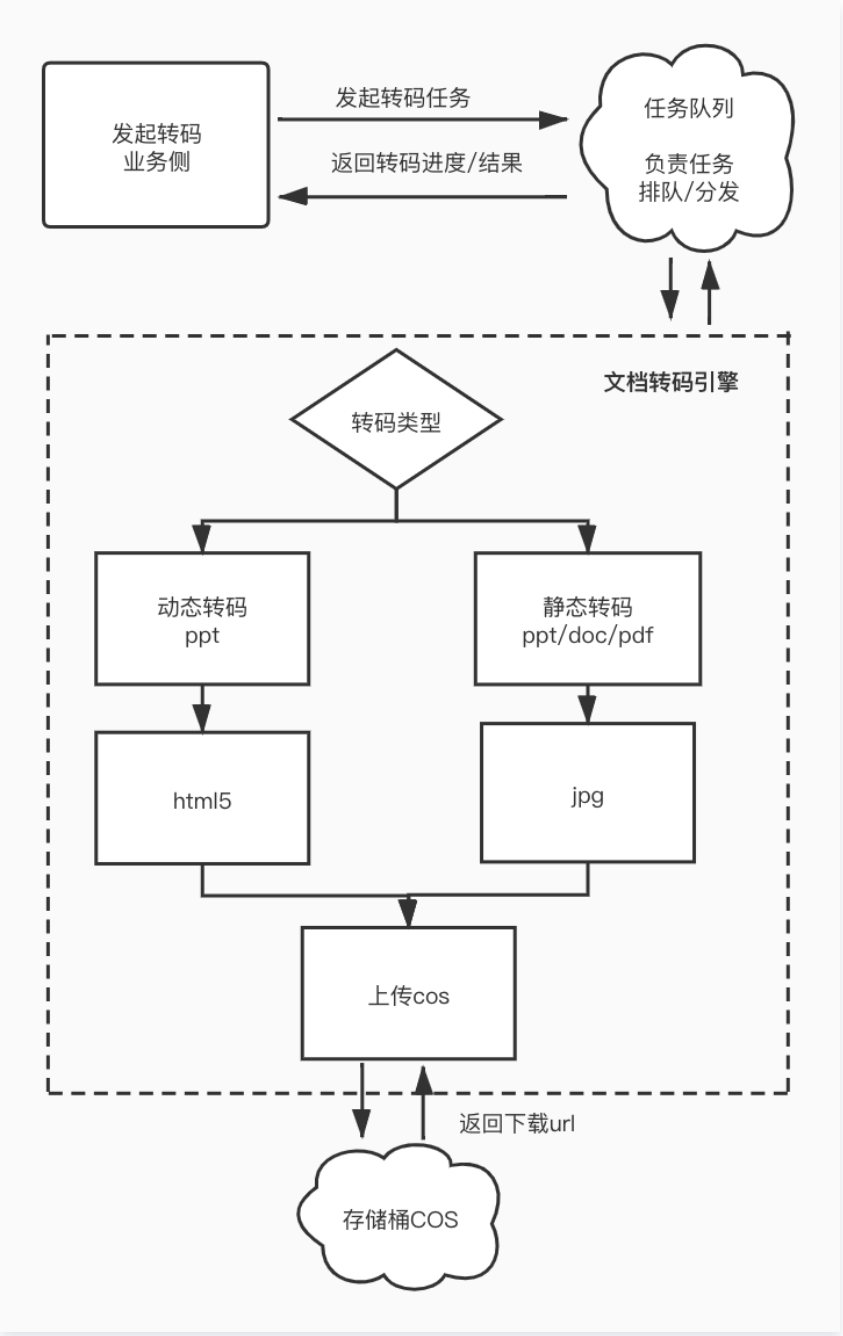
> **说明：**
> 推荐转码方案针对 Office 制作的课件兼容性好，备用转码方案针对 WPS 制作的课件兼容性好。

基本概念

动态转码：PPT/PPTX 转码成 H5，可以保留 PPT/PPTX 中的动效（只支持 PPT/PPTX）。
静态转码：PPT/PPTX/DOC/DOCX/PDF/EXCEL 转码成 静态图片，无动画效果。

架构介绍

文档转码架构图如下所示：



接入流程

Last updated：2024-12-09 10:37:22

文档转码接入有两种方式：

- 注册回调方式（适用在互动白板控制台已经 注册回调 的场景，推荐使用该方式，可以更及时收到转码进度和结果的推送）
- 主动轮询方式（适用于没有在互动白板控制台注册回调的场景）

此文档主要介绍注册回调方式发起文档转码的基本流程。

准备工作

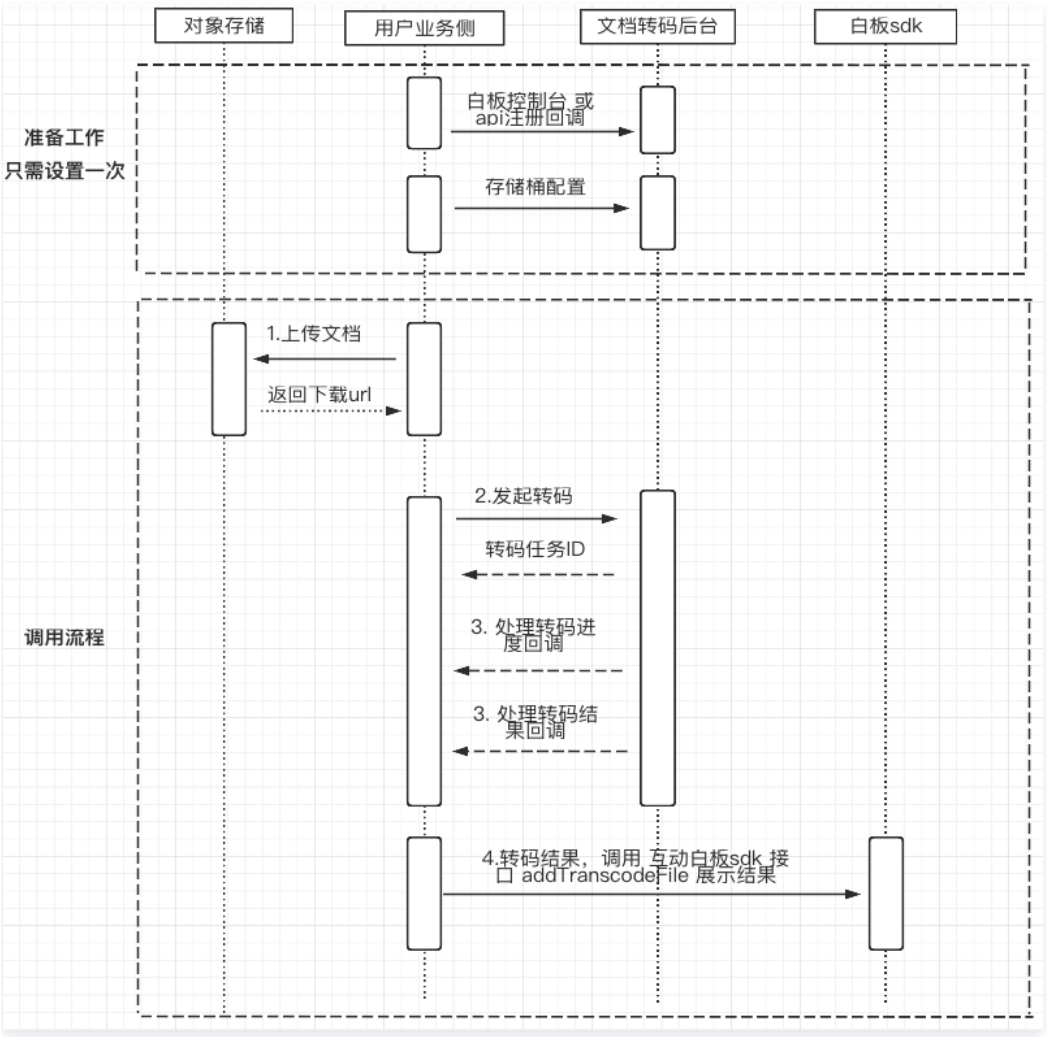
- **存储桶配置**（文档转码后的资源文件存储依赖对象存储 COS，使用文档转码功能前，请先进行 存储桶配置 ）。

> **注意**
> 如果没有进行存储桶配置，转码后的文件存储在互动白板的公共存储桶里，存储有效时间是7天。7天后，公共存储桶里的转码文件自动删除，URL的文件就会失效。建议您进行存储桶配置，将转码文件存储到您自己的存储桶里。

- **注册转码回调**（为了让转码服务器，可以将转码结果实时推送给您，您需要注册回调地址；具体方法请参考 控制台回调设置 ）。

> **注意**
> 由于文档转码存在转码耗时和排队耗时，建议使用服务端 API 提前转码，客户端直接使用转码结果。不建议直接在客户端调用互动白板 SDK 的转码接口 `applyFileTranscode` ，避免长时间的等待，影响产品体验。

交互流程（注册回调方式）



1. 上传文档（上传到 腾讯云 COS 为例）

为了能让腾讯云转码服务器获取到您待转的课件，您需要提供可以供转码服务器下载课件的 URL 地址。

> **说明：**
> 1. 推荐使用腾讯云的 COS 服务来提供下载地址，当然也可以上传到**其他存储服务器**或者**其他方式上传**。
> 2. 此处以上传到腾讯云 COS GOLANG 语言为例，更多语言实现，请参考 腾讯云 COS 。

示例代码：

```
package main

import (
    "context"
    "fmt"
    "log"
    "net/http"
    "net/url"
    "github.com/tencentyun/cos-go-sdk-v5"
    "os"
)

func main() {
    // 初始化cos 资源 TODO：将 examplebucket-1250000000 和 COS_REGION 修改为真实的信息
```

```
    domain := "https://examplebucket-1250000000.cos.COS_REGION.myqcloud.com"
    u, _ := url.Parse(domain)
    b := &cos.BaseURL{BucketURL: u}
    cosClient := cos.NewClient(b, &http.Client{
        Transport: &cos.AuthorizationTransport{
            // TODO: COS_SECRETID 需要替换成用户真实的 SECRETID， COS_SECRETKEY 需要替换成用户真实的
SECRETKEY
            SecretID:  "COS_SECRETID",
            SecretKey: "COS_SECRETKEY",
        },
    })
    // TODO: 上传到 cos 的对象键 名称
    keyName := "test/objectPut.go"
    // 本地文件路径
    localFilePath := "test/objectPut.go"
    // 获取文件大小
    file_info, err := os.Stat(localFilePath)
    if err != nil {
        panic(err)
    }

    opt := &cos.ObjectPutOptions{}
    opt.ObjectPutHeaderOptions = &cos.ObjectPutHeaderOptions{}
    opt.ObjectPutHeaderOptions.ContentLength = int(file_info.Size())

    // 对象键（Key）是对象在存储桶中的唯一标识。
    // 例如，在对象的访问域名 `examplebucket-1250000000.cos.COS_REGION.myqcloud.com/test/objectPut.go`
中，对象键为 test/objectPut.go
    // 开始上传
    _, err = cosClient.Object.PutFromFile(context.Background(), keyName, localFilePath, nil)
    if err != nil {
        // 上传失败
        panic(err)
    }
    // 上传成功，组装 resultUrl
    resultUrl := fmt.Sprintf("%s/%s", domain, keyName)
    log.Printf("upload successful! resultUrl[]%s", resultUrl)
}
```

2. 发起转码

由于动态转码存在转码耗时和排队耗时，建议使用服务端 API 提前转码，客户端直接使用转码结果。不建议直接在客户端调用转码接口 `applyFileTranscode`，避免长时间的等待，影响产品体验。

> **说明：**
> 1. 接口说明，请参考 文档转码相关接口 。 回调事件请参考 文档转码回调事件 。
> 2. SecretId 和 SecretKey 请在 API 密钥管理 中获取。
> 3. 其他开发语言 或 转码接口的示例，请参考 示例代码生成 。

示例代码（GOLANG 为例）：

```
package main

import (
        "fmt"
```

```
        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
        tiw "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tiw/v20190919"
)

func main() {
        // "SECRETID" 和 "SECRETKEY" 需要从控制台的 API 密钥管理 中获取
        credential := common.NewCredential(
                "SECRETID",
                "SECRETKEY",
        )
        cpf := profile.NewClientProfile()
        cpf.HttpProfile.Endpoint = "tiw.tencentcloudapi.com"
        client, _ := tiw.NewClient(credential, "ap-guangzhou", cpf)

        request := tiw.NewCreateTranscodeRequest()
        // SdkAppId 为用户自己的互动白板应用 ID  Url 为上传文档后，得到的下载地址
        params := "{\"SdkAppId\":xxxxxxxxx,\"Url\":\"https://board-sdk-
1259648581.file.myqcloud.com/TIC/1590997573551/欢迎新同学.pptx\"}"
        err := request.FromJsonString(params)
        if err != nil {
                panic(err)
        }
        response, err := client.CreateTranscode(request)
        if _, ok := err.(*errors.TencentCloudSDKError); ok {
                fmt.Printf("An API error has returned: %s", err)
                return
        }
        if err != nil {
                panic(err)
        }
        fmt.Printf("%s", response.ToJsonString())
}
```

3. 处理转码回调

必须要确保在控制台或 API 成功注册了回调，并且回调地址是公网可访问，才能正常收到转码数据回调；如果还没有配置回调，请参考 控制台回调设置。
示例代码（GOLANG 为例）：

```
import (
    "encoding/json"
    "log"
    "testing"
)

const (
    // 转码进度改变
    TranscodeProgressChanged = "TranscodeProgressChanged"
    // 转码结束
    TranscodeFinished = "TranscodeFinished"
)

// 转码错误
```

```
type commonError struct {
    Code    string `json:"Code"`
    Message string `json:"Message"`
}

// 转码的事件数据
type EventData struct {
    Error               *commonError `json:"Error"`               // 错误信息；为空则没有错误
    TaskId              string       `json:"TaskId"`              // 文档转码任务 id
    Progress            int          `json:"Progress"`            // 任务进度
    Resolution          string       `json:"Resolution"`          // 文档分辨率
    Title               string       `json:"Title"`               // 文档标题
    Pages               int          `json:"Pages"`               // 文档总页数
    ResultUrl           string       `json:"ResultUrl"`           // 文档转码结果
    ThumbnailUrl        string       `json:"ThumbnailUrl"`        // 缩略图 Url
    ThumbnailResolution string       `json:"ThumbnailResolution"` // 缩略图分辨率
    CompressFileUrl     string       `json:"CompressFileUrl"`     // 转码结果打包压缩文件的下载 Url
}

// 转码回调结果
type TranscodeResult struct {
    EventType  string    `json:"EventType"`  // 事件类型
    ExpireTime int       `json:"ExpireTime"` // 签名过期时间
    SdkAppId   int       `json:"SdkAppId"`   // 互动白板应用 SdkAppId
    Sign       string    `json:"Sign"`       // 回调签名
    Timestamp  int       `json:"Timestamp"`  // 事件生成的 Unix 时间戳，单位秒
    EventData  EventData `json:"EventData"`  // 事件具体信息
}

// 通过控制台 或 api 注册到 腾讯文档转码后台 的 转码事件回调接口
func TrascoderEventCallback(resultStr []byte) {
    transcodeResult := TranscodeResult{}

    err := json.Unmarshal(resultStr, &transcodeResult)
    if nil != err {
        panic(err)
    }
    // 处理转码结果
    if TranscodeProgressChanged == transcodeResult.EventType {
        // 转码进度改变
        log.Printf("transcode progress [%d]", transcodeResult.EventData.Progress)
    } else if TranscodeFinished == transcodeResult.EventType {
        // 转码结束
        if nil != transcodeResult.EventData.Error {
            // 转码失败
            log.Printf("transcode failed!err[%v]", transcodeResult.EventData.Error)
            return
        }
        // 转码成功
        log.Printf("transcode successful! EventData[%v]", transcodeResult.EventData)
    }
}
```

4. 使用转码结果

客户端将转码结果，设置到互动白板 ，即可实现文档的上/下一步、上/下一页等功能。白板接口请参考 白板接口说明 。
代码示例（JS 为例）

```
// 客户端 根据 文档转码 返回的结果，组装参数
let config = {
      url:    eventData.resultUrl,
      title: eventData.title,
      pages: eventData.pages,
      resolution: eventData.resolution
}
// 将 文档转码 的结果，加入到白板
this.teduBoard.addTranscodeFile(config);
```

效果展示：

快速体验

Last updated：2024-09-11 09:08:12

本文通过 腾讯云 API Explorer 工具，演示发起文档的基本流程。

准备工作

互动白板 SdkAppid，如果您还没有创建应用，请参考文档 应用管理 在 互动白板控制台 里创建。

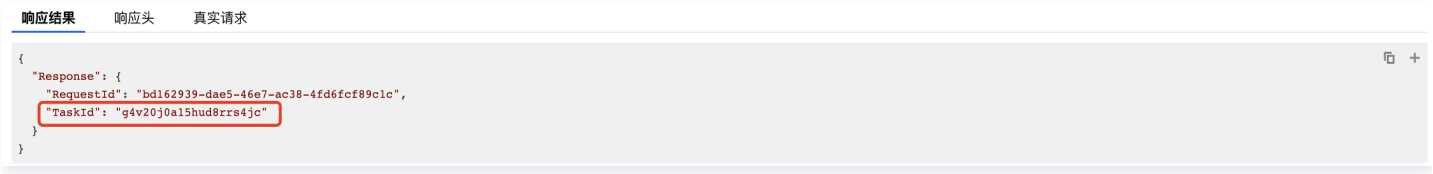
操作步骤

步骤1：上传待转码的 PPT 文件，获取文档的下载链接

如果您使用腾讯云对象存储，可以参考文档 简单上传 。

步骤2：发起转码任务

- 打开 腾讯云 API Explorer，选择**互动白板 > 文档转码相关接口 > 创建文档转码任务**。
- 输入参数里 Region 请选择华南地区(广州)，SdkAppId 填写您的互动白板应用 SdkAppId，如果您还没有创建应用，请参考文档 应用管理 在 互动白板控制台 里创建。
- `Url` 参数里填入步骤一中获取的文档下载 URL，更多参数描述请参考文档 创建文档转码任务 。
- 单击**在线调用 > 发送请求**按钮发送转码请求，获取 TaskId。



- 如图所示获取到的 TaskId 为 g4v20j0a15hud8rrs4jc。

步骤3：查询转码结果

- 打开 腾讯云 API Explorer，选择**互动白板 > 文档转码相关接口 > 查询文档转码任务**。
- 输入参数里 Region 请选择华南地区(广州)，SdkAppId 填写您的互动白板应用 SdkAppId， `TaskId` 填入步骤二中获取的 TaskId。
- 点击在线调用->发送请求按钮查询文档转码结果，可以多次调用等待进度 `Progress=100` 时， `ResultUrl` 参数即为您的文档转码结果。

常见问题

Last updated：2024-09-11 09:08:12

本文记录文档转码使用过程中，遇到的常见问题。

接口使用问题

文档转码的作用是什么呢？为什么互动白板需要配合使用文档转码？

文档转码的作用，是把本地文档，转码成线上可查看的一种方式（如把 PPT 转码成 H5）。
互动白板 使用过程中，老师需要将自己的课件，展示给学生观看，这时就需要通过文档转码，将老师的课件转成线上可查看，然后学生就可以看到老师的课件了。

文档转码支持单独接入使用吗？

不支持。目前文档转码，只能配合 互动白板 使用。若您的互动白板服务未产生时长消耗，却调用了文档转码服务，则您发起的转码任务会被低优先级处理，可能出现长时间排队的情况。

为什么不建议使用 互动白板 SDK 的 applyFileTranscode 发起文档转码呢？

由于文档转码存在转码耗时和排队耗时，建议使用服务端 API 提前转码，客户端直接使用转码结果。不建议直接在客户端调用互动白板 SDK 的转码接口 `addTranscodeFile`，避免长时间的等待，影响产品体验。
后续版本，该接口已废弃，不能使用。

如果出现转码失败，是否需要业务侧重试呢？

要根据具体的错误信息，如果是源文件内容导致的错误，需要先修改源文件后再重试转码。
如果是系统错误，不需要。因为转码服务器内部，已经加了转码失败的重试逻辑，业务侧无需重试（服务器重试逻辑：每台服务器重试3次，尝试2台转码服务器）。
建议 设置转码回调地址，根据具体的错误码来做处理。

服务端 API，Region 支持哪些区域呢？

目前服务端 API 接口只支持区域广州，在调用腾讯云 API 发起转码、查询转码结果时，`Region` 参数请填写 `ap-guangzhou`。

如何在 SDK 中使用转码结果？

请参考 互动白板 各个平台 API 文档中的 `addTranscodeFile` 接口说明。

文档转码后的文件，存储在哪里呢？

- 建议设置 存储桶配置，这样转码后的文件会自动转存到用户配置的 COS 桶里。
- 如果没有设置 存储桶配置，则默认存储在腾讯云转码服务器默认的 COS 桶，存储周期为7天；7天后，转码文件自动删除。

文档转码是怎么收费的呢？

具体收费标准，请参考 购买指南。

事件回调问题

为什么有时会出现长时间接收不到事件回调呢？

1. 转码和录制，回调的超时时间为10s，如果回调失败，会间隔5s，最大重试5次。
2. 如果长时间收不到回调，并且确认回调地址正常可用，可能原因如下：
    - 回调 URL 和服务器的 RTT 超过了10s。
    - 回调因为网络原因丢失了。
3. 所以建议在代码逻辑中加上：如果出现长时间收不到回调时，配合查询任务进度接口确认。

缩略图问题

静态和动态转码都支持缩略图吗？怎么生成？

静态 和 动态转码都支持缩略图。只需要在 发起文档转码 `CreateTranscode` 时，传入参数 `ThumbnailResolution` 参数，并且指定分辨率即可。例如，传入 `"ThumbnailResolution":"640x480"`，转码结果里，即可看到 `"ThumbnailUrl":"http://xxxx/xxx/"` 缩略图 URL 地址。

为什么调用互动白板 `getThumbnailImages` 获取缩略图地址接口，有时返回的缩略图地址无法打开呢？

1. 互动白板 getThumbnailImages 的内部实现，只是根据固定的规则，组装缩略图地址；无法验证缩略图地址是否真实有效；所以如果需要缩略图功能，转码时请务必传入 `ThumbnailResolution` 参数。
2. 静态转码 如果用户没有开通 存储桶配置，getThumbnailImages 会把文档转码结果 ResultUrl 按照固定规则修改成缩略图地址。但是 动态转码 ResultUrl 是 后 H5 链接，无法修成缩略图地址；所以现象就是，静态转码 缩略图正常，但是 动态转码 的缩略图地址打开 http 404 错误。所以如果需要缩略图功能，转码时请务必传入 `ThumbnailResolution` 参数。

转码耗时问题

文档转码，大概会耗时多久呢？

PPT 动态转码成 HTML5 大约为1s/页，静态转码成图片大约为0.5s/页。

为什么有时，转码耗时很长？

1. 文档转码资源有限，存在排队逻辑；当高峰时段，转码量过多时，新的任务会进入排队逻辑 被有序转码；所以就会增加排队时间。
2. 建议老师提前备课，错开高峰期转码；以免上课时转码时间过长，影响上课体验。

为什么发起转码后，长时间收不到转码回调呢？

1. 确认已经设置了回调，并且回调地址可用；设置回调，可以参考 回调设置 。
2. 转码正在排队中。可以通过 查询转码任务 接口确认；如果返回的 Status 为 "QUEUED"；则代表正在排队中。

网络问题

为什么会提示，文档 URL 解析错误呢？URL 具体的要求是什么呢？

1. 目前只支持以 `http://` 或 `https://` 前缀的 URL。
2. 请确保文档下载 URL 的 path 后缀，以 ppt(x)/doc(x)/pdf 结尾；否则下载 URL 会校验失败。

为什么会提示，文档下载失败呢？

1. 请确认文档 URL 的访问权限，并且可以通过浏览器正常下载；
2. 如果是 云服务器 COS 桶，请确认访问权限是否为**公有读私有写**，或者签名是否过期。

为什么会提示，文档下载超时呢？

1. 转码服务器，默认下载超时时间为60s（防止因为下载时间过长，过多占用服务器资源），建议文档最好不要超过200M，防止文档过大，导致下载超时。
2. 如果使用腾讯云 COS 桶存储，下载理论可以走腾讯云内网，所以可以支持更大的文件下载。

文档格式问题

文档转码，支持哪些格式呢？

- 动态演示文件：pptx、ppt；
- 静态演示文件：pptx、ppt；
- 文字文件：doc、docx；
- 图片文件：jpg、jpeg、png、bmp；

- 音频文件：MP3；
- 视频文件：MP4；
- H5 文件：htm、 html；
- 其他格式文件：pdf；
- 推荐转码方案暂时不支持文件： excel 、xlsx、xltx、txt 格式，如需使用，可以使用 备用转码方案 。

ppt 动态转码，格式有哪些需要注意的地方呢？可以汇总一下吗？

1. 避免使用不支持转码的元素，如：" 墨迹 "；可以通过 PowerPoint 打开 PPT，然后**开始 > 选择 > 选择窗格 > 查看是否存在墨迹**。
2. 不支持5种动效；Bold Flash（加粗闪烁）、Underline（下划线）、Grow With Color（颜色渐变）、Bold Reveal（加粗显示）、Wipe（擦除）。
3. 建议使用 Microsoft Office 2007或以上版本（WPS 和 keynote 都有一定的兼容性问题）。
4. 转码文件必须是**可编辑**的。不支持“只读”、“加密”、或其他保护，导致 PPT 无法编辑的文件转码。
5. 请勿插入 flash 动画，不支持 flash 动画播放。
6. 如果使用 WPS，请注意不支持 WPS PPT 中的音视频元素。
7. 如果使用 keynote，请注意插入的音频播放标签，无法展示。
8. 建议使用操作系统默认的中英文字体，请勿使用自己安装的字体，否则可能会出现转码失败或者转码 H5 格式异常，单击查看 PPT 支持的字体列表 。
9. PPT 页数不要过多，页数越多，转换速度越慢（目前最大支持500页转码）。
10. 尽量减小 PPT 体积大小，最好控制在50M以内；因为转码服务器，下载超时时间为1分钟；如果文件过大，可能会因为下载超时导致转码失败。
11. 目前文档转码，对 PPT 中 内嵌音视频 支持不完善；不建议 PPT 中内嵌音视频，否则转码后的 H5 中，可能会遇到音视频播放失败或者本地和远端音视频不同步等问题。
12. 建议不要在 PPT/PPTX 文档中，对文字添加荧光色背景。

转码如何保留动画呢?

1. 使用 PPT/PPTX 文档转码，默认会保留动画；
2. 如果设置了 IsStaticPPT=true，则所有的文档都会转码成 静态图片(包括 PPT 文件)。

提交的文档，会转码成什么格式的文件呢?

1. 默认情况下，PPT/PPTX 文档会转码成 HTML5 页面，能够还原 PPT 原有的动画效果，其他文档转码成静态图片。
2. 如果设置了 IsStaticPPT=true，则所有的文档都会转码成 静态图片(包括 PPT 文件)。

动态转码支持音视频吗?

支持，不过需要音视频文件是正常无损坏的。

转码效果问题

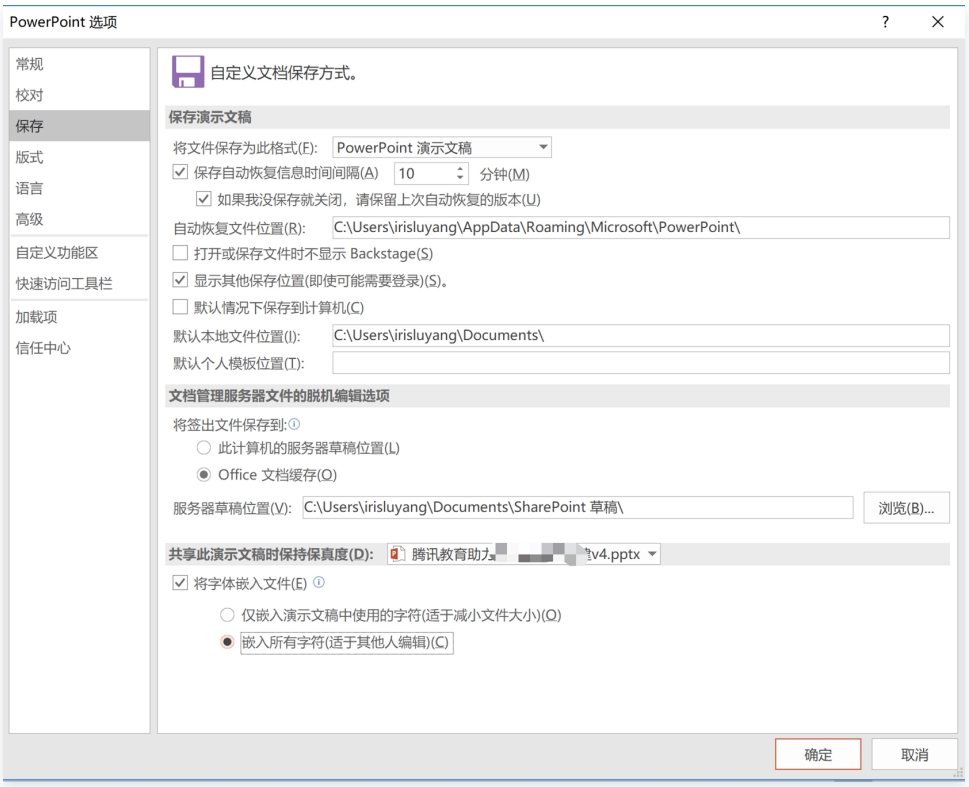
动态转码出现某些字体错位，或大小不一致，如何处理?

1. 可能使用了转码服务器不支持的字体库导致，需要去除或修改不支持的字体 查看转码服务器支持字体 。
2. 对于动态转码和静态转码，如果遇到字体不支持的情况，也可以使用 Office 的将字体嵌入文件的功能。

将字体嵌入文件，操作方法：

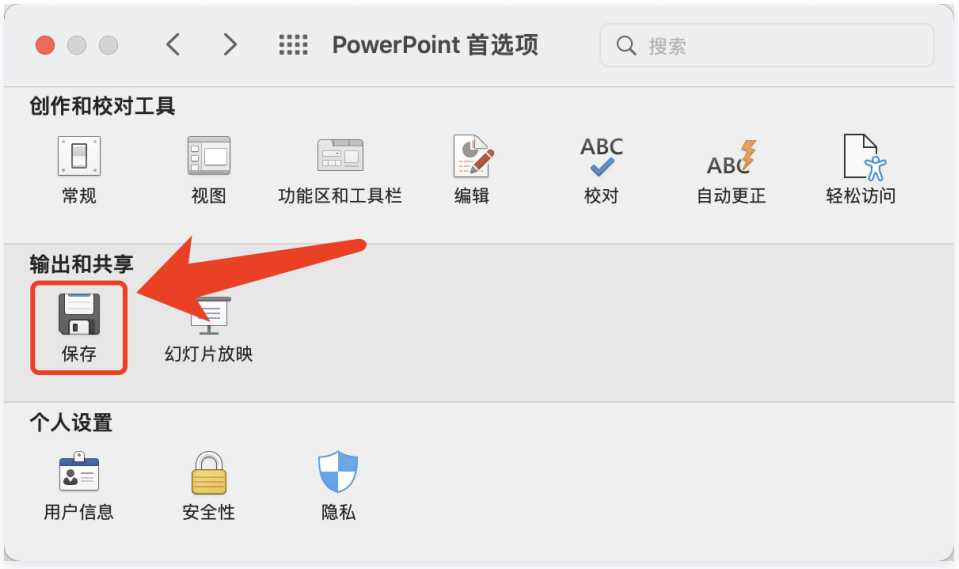
- **Windows 端：**打开文档，选择**文件 > 选项 > 保存**，然后勾选 **将字体嵌入文件 > 嵌入所有字符(适于其他人编辑) > 确定** 保存文档，看到 Office 的底部状态栏，显示**正在嵌入字体**即可。

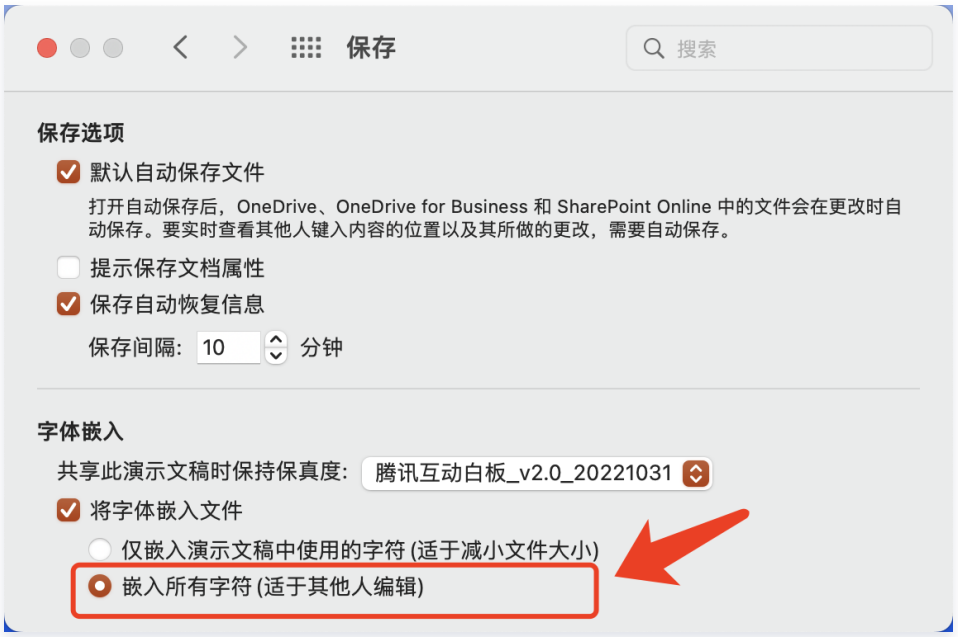
  如图所示：



- **macOS 端：**打开文档，选择 **PowerPoint > 偏好设置 > 保存**，然后 **勾选将字体嵌入文件 > 嵌入所有字符(适于其他人编辑) >** 保存文档，看到 Office 的底部状态栏，显示**正在嵌入字体**即可。

  如图所示：





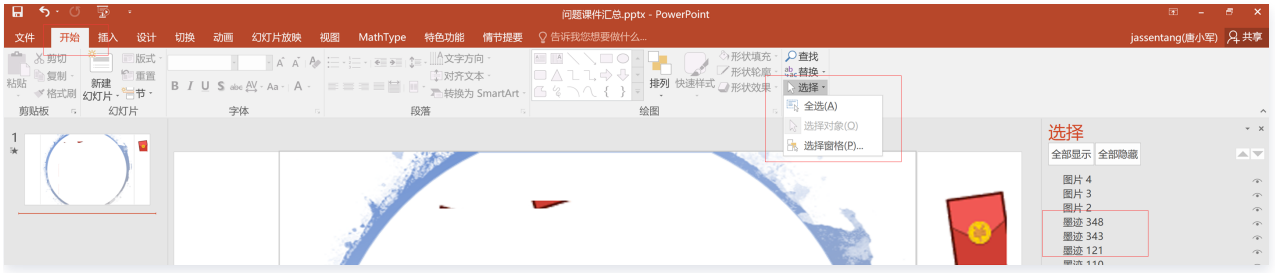
为什么有些 PPT 动效，转换为 H5 后，丢失了呢?

1. 确认没有使用转码不支持的4种动效 Bold Flash(加粗闪烁)、Underline(下划线)、Grow With Color(颜色渐变)、Bold Reveal(加粗显示)。
2. 最好不要使用组合动效，因为组合动效过于复杂，加大了 H5 还原的难度。

转码错误码解析

错误码：-14 Occur issue in convert processing 是什么原因? 该如何解决?

问题原因：PPT 的内容中含有不支持转码的元素
解决方法：需要根据错误信息中提示的具体页数，查看是否使用了 " 墨迹 " 和 4种不支持的动效。
查看**墨迹**方法：PowerPoint 打开 PPT > 依次选择**开始** > **选择** > **选择窗格** > 查看右边框是否存在“墨迹”元素。
如图所示：

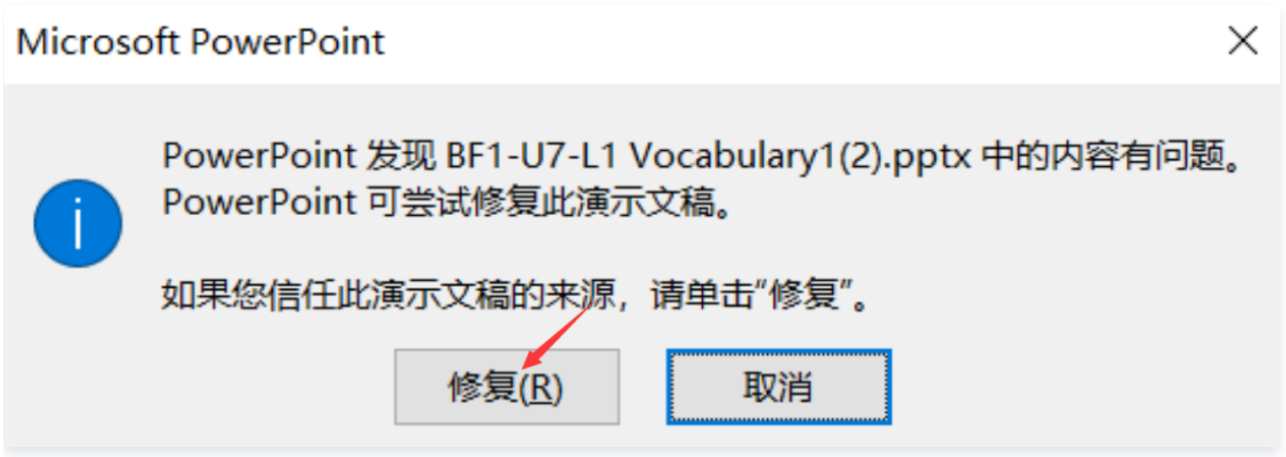


错误码: -9 ppt to pptx failed! 是什么原因呢? 该如何解决?

问题原因：PPT 转换成 PPTX 失败；可能是 PPT 使用的 PowerPoint 版本太老 或者 PPT 中含有某些元素，不支持转换成 PPTX 格式。
解决方法：PowerPoint 打开 **PPT** > **手动转换为 PPTX** > **确认 PPTX 打开不会有任何报错** > 在使用转换后的 PPTX 重新发起转码。

错误码: -20 UnKnow Ppt property! 是什么原因呢? 该如何解决?

问题原因：PPT/PPTX 是 WPS 制作的文件，可能是“只读”、“加密”、或有其他保护，导致 PPT 无法被编辑。
解决方法：使用 Office PowerPoint 打开 **PPT/PPTX** > **修复 PowerPoint 提示的错误** > **确保 PPT/PPTX 已经修改为可编辑** > **保存** > 重新发起文档转码。



错误码：-27 Download ppt data timeout! 是什么原因呢？该如何解决？

问题原因：PPT/PPTX 体积过大 或者 文档 URL 下载的带宽太小，导致转码服务器下载超时了；转码服务器目前下载超时时间为1分钟。
解决方法：

1. 尝试去掉 PPT/PPTX 中占用体积较大的元素，例如音视频文件等。
2. 确认文档 URL 源站的下载带宽。

错误码：-106 download file suffix not support! 是什么原因呢？该如何解决？

问题原因：文档转码 的 URL path，不是以 ppt(x)/doc(x)/pdf 结尾（此处注意，URL 中 ?号后面的是参数，不是 path）。
解决方法：请确保文档下载 URL 的 path 后缀，以 ppt(x)/doc(x)/pdf 结尾；否则下载 URL 会校验失败。

错误码：-116 File request transcode exception!

pdf源文件有问题（超过最大体积，或页码），静态转码支持最大为200M ，2000页。

错误码：-109 File download link connect exception!

是请求转码时的 URL 有问题，可以在浏览器打开看下，URL 是否能正常访问。

返回错误提示 code: LimitExceeded.TaskConcurrency ，message：the number of concurrent tasks exceeds limit, please try again later，Status：ERROR

单个应用的任务数超过并发，只能等前面的任务转码完了 ，才能发起新的请求。
默认情况下，单个白板应用最多20个动态转码任务同时运行，最多40个静态转码任务同时运行。

返回错误提示： UnKnow Ppt property, ppt opened failed! Please use Microsoft office to check the ppt format!

源文件有问题，使用 Office 无法打开。原因是 WPS 制作的课件存在兼容性问题 ，建议使用 Office 制作课件。

其他问题

文档转码返回 URL 的有效期是多长时间?

1. 如果没有开通存储桶配置，转码后的文件存储在互动白板的 COS 桶里，默认存储时间是7天。7天后，存储桶里的转码文件自动删除，URL 的文件就会失效。
2. 强烈建议开通存储桶配置，这样转码后的文件，就可以自动转存到用户自己的 COS 桶下，不用担心转码文件过期的问题存储桶开通指南 。

转码错误码对照表

文档转码 错误描述中的错误码列表

错误码	错误描述	解决方法

0	任务执行成功	无
-1	PPT 下载的 URL 格式错误	请检查 PPT 下载 URL
-2	PPT 打开过程中发生未知错误	请检查 PPT 格式是否正确
-3	PPT 打开超时	请检查 PPT 格式是否正确
-4	PPT 打开无响应	请检查 PPT 格式是否正确
-5	PPT 文件被加密	不支持转换已加密的 PPT
-6	未知 PPT 格式	请检查 PPT 格式是否正确
-7	PPT 打开时发生异常	请检查 PPT 格式是否正确
-8	PPT 为只读格式	不支持只读 PPT，请检查 PPT 格式是否正确
-9	PPT 转码失败	请检查 PPT 格式是否正确
-10	PPTX 格式解析错误	请检查 PPT 格式是否正确
-11	PPT 下载失败，未知错误	请检查 PPT 下载 URL 是否有效
-12	H5 上传失败	请联系客服人员
-13	PPT 转换服务未加载	请联系客服人员
-14	PPT 转码过程中，发生错误	请根据错误信息中的页数，检查该页中的**元素**或**动画组合**是否正确； 如果错误信息中，无具体页数"unknown page"，则代表无法获取某一页转换失败，该 PPT 不支持转换
-15	PPT 转码文件生成失败	请联系客服人员
-16	PPT 转换模式异常	PPT 格式不支持或联系客服人员
-17	PPT 超过最大的转换页数限制（目前为 500）	不支持转换超过500页的 PPT
-18	PPT 转换失败，错误未知	PPT 格式不支持或联系客服人员
-19	PPT 被加密	不支持加密的 PPT
-20	PPT 未知属性错误	请检查 PPT 格式是否正确
-21	PPT 检测属性超时	请检查 PPT 格式是否正确
-22	PPT 转换异常	请检查 PPT 格式是否正确
-23	PPT 下载链接含有非法字符	请检查 PPT 下载链接是否正确
-24	PPT 下载链接打开失败	请检查 PPT 下载地址是否有效
-25	PPT 下载链接打开超时	请检查 PPT 下载链接是否有效或下载服务器网络状况
-26	PPT 下载数据失败	请检查 PPT 下载服务器网络状况
-27	PPT 下载数据超时	请检查 PPT 下载服务器网络状况或是否 PPT 过大，导致下载超时
-30	PPT 转码引擎内部错误	请联系维护人员
-31	PPT 中含有 JPEG 病毒	请根据提示的页码，修改里面的 JPEG 图片
-32	PPT 的幻灯片数量为空	不支持幻灯片为空的 PPT 转换

-33	转码完成后，上传到 COS 桶失败	请在 腾讯云 cam 控制台 检查是否存在 TIW_QCSRole 角色，以及该角色是否具有 QcloudAccessForTIWRoleInWhiteboardResourcesManagement 这个策略 如果没有，请先创建角色 TIW_QCSRole，并给该角色分配 QcloudAccessForTIWRoleInWhiteboardResourcesManagement 策略
-101	task info 文件读取失败	请联系维护人员
-102	task info 格式不正确	请联系维护人员
-103	task info 的 error_code 不等于0	请联系维护人员
-104	task 的 URL 为空	请联系维护人员
-105	task 的 下载 URL 错误	请检查转码文件下载链接
-106	不支持的文件名后缀	请检查文件名后缀名
-107	文件下载失败，未知错误	请检查下载链接地址是否有效
-108	下载链接非法	请检查转码文件下载链接
-109	下载链接打开失败	请检查下载链接地址是否有效
-110	下载链接打开超时	请检查下载链接地址是否有效
-111	下载失败	请检查下载链接地址是否有效
-112	下载超时	请检查下载链接带宽
-113	本地临时文件夹创建失败	请联系维护人员
-114	文件上传转码后台失败	请联系维护人员
-121	超过最大页数	不支持超过500页的文件转码
-122	转码服务器打开本地文件失败	请联系维护人员
-123	转码服务器写本地文件失败	请联系维护人员
-124	转码服务器文件内容为空	请联系维护人员
-125	获取图片分辨率失败	请联系维护人员
-126	PDF 打开失败	PDF 文件格式错误或 PDF 加密，不支持转码
-127	PDF 文件加密	不支持加密 PDF 文件转码
-128	获取 PDF 页数失败	PDF 文件格式不支持
-129	PDF 转码未知错误	PDF 文件格式不支持
-130	Office 转码本地错误；包括转码任务格式，任务完成格式等	请联系维护人员
-131	Office 文件加密	文件格式不支持
-132	Office 文件未知属性错误	文件格式不支持
-133	Office 文件打开超时	文件格式不支持
-134	Office 文件打开异常	文件格式不支持

-135	Office 文件转存异常	文件格式不支持
-136	Office 转换 taskid 对应失败	请联系维护人员
-137	获取 Office 文件分辨率失败	请联系维护人员
-138	转码后的本地文件错误	请联系维护人员
-139	转码丢失了页数	请联系维护人员
-141	PPT 幻灯片为空	不支持幻灯片为空的 PPT 转码
-142	转码丢失了页数	请联系维护人员
-143	转码丢失了页数	请联系维护人员

主动轮询方式

Last updated：2024-12-06 15:57:22

文档转码 接入流程(主动轮询方式)

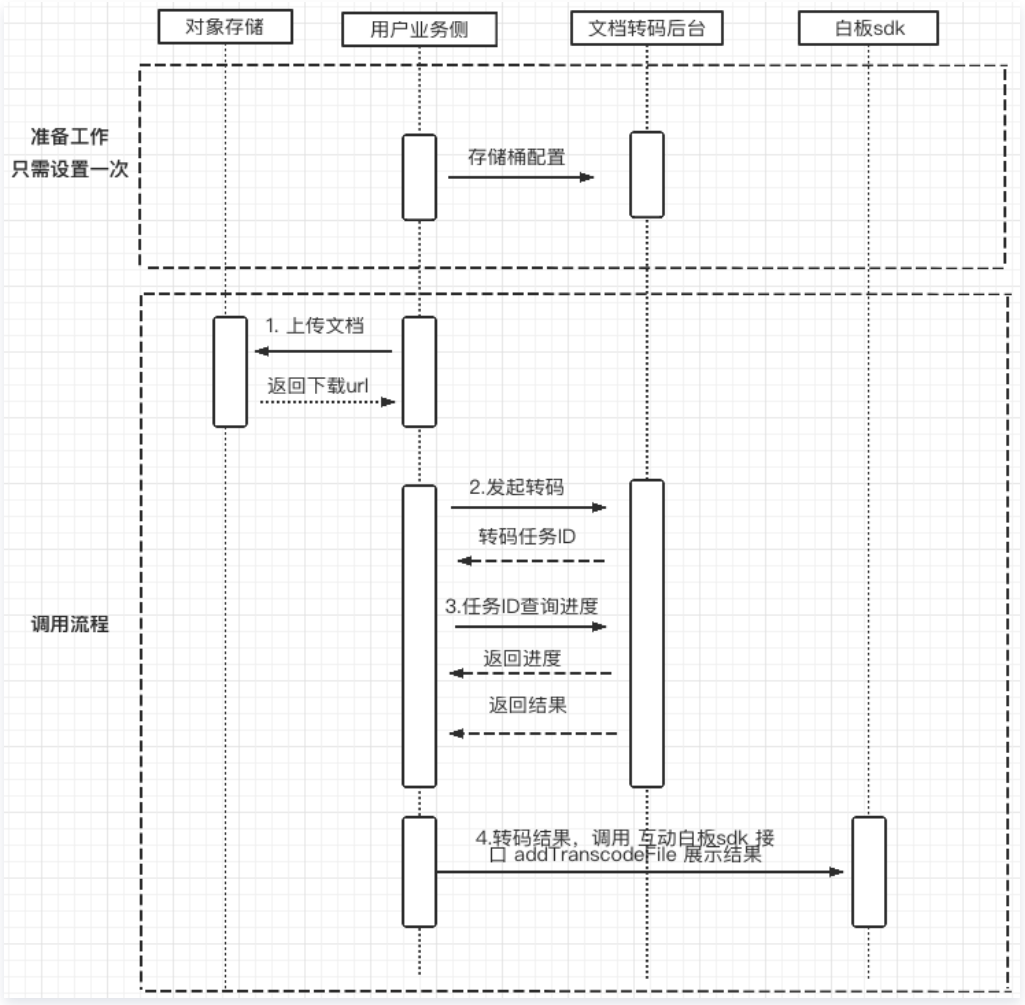
主动轮询方式，因为受轮询间隔时长限制（推荐间隔2s轮询一次转码进度），无法及时获取到转码结果，所以更建议使用 注册回调方式 。

准备工作

- **存储桶配置**（文档转码后的资源文件存储依赖对象存储 COS，使用文档转码功能前，请先进行 存储桶配置 ）

> **注意：**
> 由于文档转码存在转码耗时和排队耗时，建议使用服务端 API 提前转码，客户端直接使用转码结果。**不建议**直接在客户端调用互动白板 SDK 的转码接口 `applyFileTranscode` ，避免长时间的等待，影响产品体验。

- **交互流程（主动轮询方式）：**



接入步骤

1. 上传文档（以上传到 腾讯云 COS 为例）

为了能让腾讯云转码服务器获取到您待转的课件，您需要提供可以供转码服务器下载课件的 URL 地址。

> **说明：**
> 1. 这里推荐使用腾讯云的 COS 服务来提供下载地址，当然也可以上传到其他存储服务器 或者 其他方式上传。

2. 此处以上传到腾讯云 cos golang 语言 为例，更多语言实现，请参考 腾讯云 COS 。

示例代码：

```
package main

import (
    "context"
    "fmt"
    "log"
    "net/http"
    "net/url"
    "github.com/tencentyun/cos-go-sdk-v5"
    "os"
)

func main() {
    // 初始化 cos 资源 TODO：将 examplebucket-1250000000 和 COS_REGION 修改为真实的信息
    domain := "https://examplebucket-1250000000.cos.COS_REGION.myqcloud.com"
    u, _ := url.Parse(domain)
    b := &cos.BaseURL{BucketURL: u}
    cosClient := cos.NewClient(b, &http.Client{
        Transport: &cos.AuthorizationTransport{
            // TODO：COS_SECRETID 需要替换成用户真实的 SECRETID， COS_SECRETKEY 需要替换成用户真实的
SECRETKEY
            SecretID:  "COS_SECRETID",
            SecretKey: "COS_SECRETKEY",
        },
    })
    // TODO：上传到 cos 的对象键 名称
    keyName := "test/objectPut.go"
    // 本地文件路径
    localFilePath := "test/objectPut.go"
    // 获取文件大小
    file_info, err := os.Stat(localFilePath)
    if err != nil {
        panic(err)
    }

    opt := &cos.ObjectPutOptions{}
    opt.ObjectPutHeaderOptions = &cos.ObjectPutHeaderOptions{}
    opt.ObjectPutHeaderOptions.ContentLength = int(file_info.Size())

    // 对象键（Key）是对象在存储桶中的唯一标识。
    // 例如，在对象的访问域名 `examplebucket-1250000000.cos.COS_REGION.myqcloud.com/test/objectPut.go`
中，对象键为 test/objectPut.go
    // 开始上传
    _, err = cosClient.Object.PutFromFile(context.Background(), keyName, localFilePath, nil)
    if err != nil {
        // 上传失败
        panic(err)
    }
    // 上传成功，组装 resultUrl
    resultUrl := fmt.Sprintf("%s/%s", domain, keyName)
    log.Printf("upload successful! resultUrl[]%s", resultUrl)
```

```
}
```

2. 发起转码

由于动态转码存在转码耗时和排队耗时，建议使用服务端 API 提前转码，客户端直接使用转码结果。不建议直接在客户端调用转码接口 `addTranscodeFile` ，避免长时间的等待，影响产品体验。

> **说明：**
> 1. 接口说明，请参考 文档转码相关接口 。
> 2. SecretId 和 SecretKey 请在 API 密钥管理 中获取
> 3. 其他开发语言 或 转码接口的示例，请参考 示例代码生成 。

示例代码（golang 为例）：

```
package main

import (
        "fmt"

        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
        "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
        tiw "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tiw/v20190919"
)

func main() {
         // "SECRETID" 和 "SECRETKEY" 需要从控制台的 API 密钥管理 中获取
         credential := common.NewCredential(
                  "SECRETID",
                  "SECRETKEY",
         )
         cpf := profile.NewClientProfile()
         cpf.HttpProfile.Endpoint = "tiw.tencentcloudapi.com"
         client, _ := tiw.NewClient(credential, "ap-guangzhou", cpf)

         request := tiw.NewCreateTranscodeRequest()
         // SdkAppId 为用户自己的互动白板应用ID  Url 为上传文档后，得到的下载地址
         params := "{\"SdkAppId\":\"xxxxxxxxx\",\"Url\":\"https://board-sdk-
1259648581.file.myqcloud.com/TIC/1590997573551/欢迎新同学.pptx\"}"
         err := request.FromJsonString(params)
         if err != nil {
                  panic(err)
         }
         response, err := client.CreateTranscode(request)
         if _, ok := err.(*errors.TencentCloudSDKError); ok {
                  fmt.Printf("An API error has returned: %s", err)
                  return
         }
         if err != nil {
                  panic(err)
         }
         fmt.Printf("%s", response.ToJsonString())
}
```

3. 任务 ID 查询进度

使用第2步返回的任务 ID，查询当前的任务的进度。

> **说明：**
> 1. 接口说明，请参考 文档转码相关接口 。
> 2. SecretId 和 SecretKey 请在 API 密钥管理 中获取
> 3. 其他开发语言 或 转码接口的示例，请参考 示例代码生成 。

示例代码（golang 为例）：

```
package main

import (
    "fmt"
    "time"

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    tiw "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tiw/v20190919"
)

const (
    // 文档转码状态
    StatusQueued     = "QUEUED"     // 正在排队等待转换
    StatusProcessing = "PROCESSING" // 转换中
    StatusFinished   = "FINISHED"   // 转换完成
)

func main() {
    // "SECRETID" 和 "SECRETKEY" 需要从控制台的 API 密钥管理 中获取
    credential := common.NewCredential(
        "SECRETID",
        "SECRETKEY",
    )
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "tiw.tencentcloudapi.com"
    client, _ := tiw.NewClient(credential, "ap-guangzhou", cpf)
    //
    for {
        request := tiw.NewDescribeTranscodeRequest()
        // SdkAppId 为用户自己的互动白板应用ID  TaskId 为发起转码任务时，转码服务器返回的任务 ID
        params := "{\"SdkAppId\":xxxxxxx,\"TaskId\":\"07lt80iij017uhjp12sb\"}"
        err := request.FromJsonString(params)
        if err != nil {
            panic(err)
        }
        response, err := client.DescribeTranscode(request)
        if _, ok := err.(*errors.TencentCloudSDKError); ok {
            fmt.Printf("An API error has returned: %s", err)
            return
        }
        if err != nil {
            panic(err)
        }
        // 查询成功
```

```
        fmt.Printf("%s", response.ToJsonString())
        if nil == response.Response {
            panic(fmt.Errorf("response.Response is nil!"))
        }
        // 处理查询的结果
        if StatusQueued == *response.Response.Status {
            // 任务在队列中
            fmt.Printf("transcode task status is [%s]", *response.Response.Status)
        } else if StatusProcessing == *response.Response.Status {
            // 任务转码中
            fmt.Printf("transcode task status is [%s] progress [%d]", *response.Response.Status,
*response.Response.Progress)
        } else if StatusFinished == *response.Response.Status {
            resultUrl := *response.Response.ResultUrl
            // 转码成功
            fmt.Printf("transcode successful! resultUrl[%s]", resultUrl)
        }
        time.Sleep(1 * time.Second)
    }
}
```

4. 使用转码结果

客户端将转码结果，设置到互动白板 ，即可实现文档的 上/下一步、上/下一页等功能。

> **说明：**
> 1. 白板接口，请参考 白板接口说明 。
> 2. 此处以 web 端 js 为例，其他语言，需要自行实现。

代码示例（js 为例）：

```
// 客户端 根据 文档转码 返回的结果，组装参数
let config = {
     url:    response.Response.ResultUrl,
     title: response.Response.Title,
     pages: response.Response.Pages,
     resolution: response.Response.Resolution
}
// 将 文档转码 的结果，加入到白板
this.teduBoard.addTranscodeFile(config);
```

效果展示：

转码结果

加入到白板

转码结果 加入到互动白板

文档转码（备用）
接入流程

Last updated：2024-09-11 09:08:12

备用文档转码使用的是对象存储 COS 的文档转码能力，相对于主推的转码方案接入简单，只需要在对象存储 COS 控制台开通文档转码功能即可使用。关于对象存储文档预览更多信息请查看具体的 文档预览概述文档 。

备用转码方案的优势

1. 接入简单，一个 URL 和一个接口即可对接；
2. 备用转码方案针对 WPS 制作的课件兼容性好，主推转码方案针对 Office 制作的课件兼容性好；
3. 支持转码的格式类型多。

支持转码格式

- 动态演示文件：PPTX、PPT；
- 静态演示文件：PPTX、PPT、POT、POTX、PPS、PPSX、DPS、DPT、PPTM、POTM、PPSM；
- 文字文件：DOC、DOCX、DOT、WPS、WPT、DOTX、DOCM、DOTM；
- 表格文件：XLS、XLT、ET、ETT、XLSX、XLTX、CSV、XLSB、XLSM、XLTM、ETS；
- 其他格式文件：PDF、LRC、C、CPP、H、ASM、S、JAVA、ASP、BAT、BAS、PRG、CMD、RTF、TXT、LOG、XML、HTM、HTML。

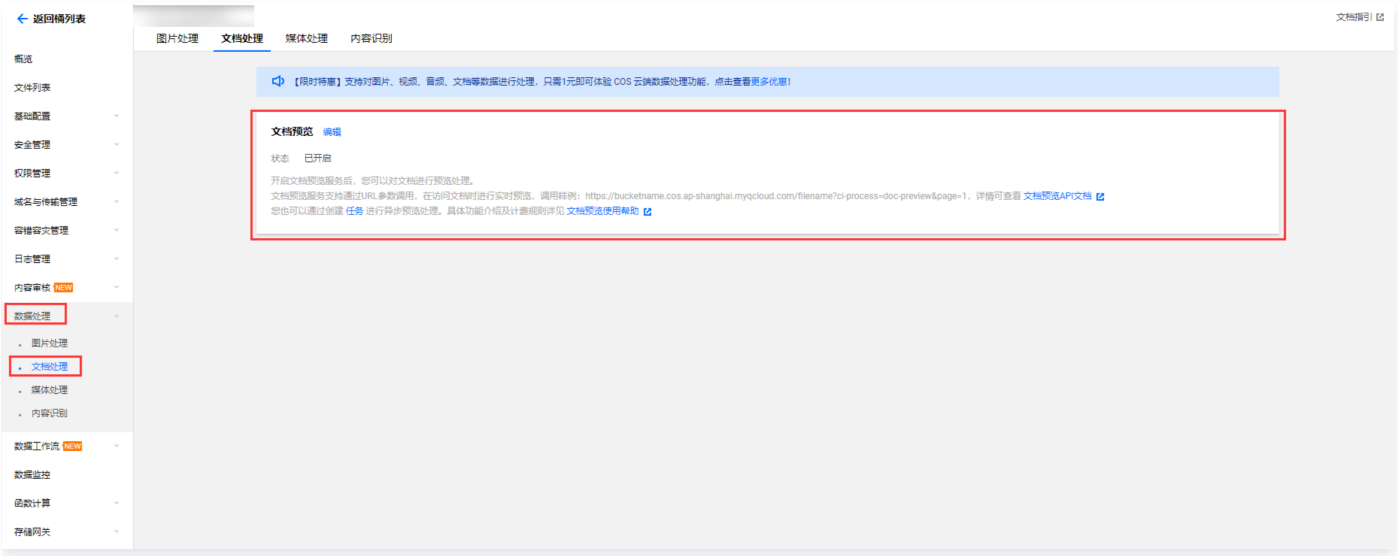
> **说明：**
> 1. 目前支持将上述文件类型转码为 JPG、PNG、PDF、HTML 格式。
> 2. 输入文件大小限制在200MB之内，输入文件页数限制在5000页之内。
> 3. 备用转码方案针对 WPS 制作的课件兼容性好，主推转码方案针对 Office 制作的课件兼容性好。

适用互动白板版本

互动白板从2.6.0的版本开始支持 文档转码（备用） 。

转码服务开通

在腾讯云 对象存储控制台 中，在存储桶中开通转码服务。



在互动白板中使用

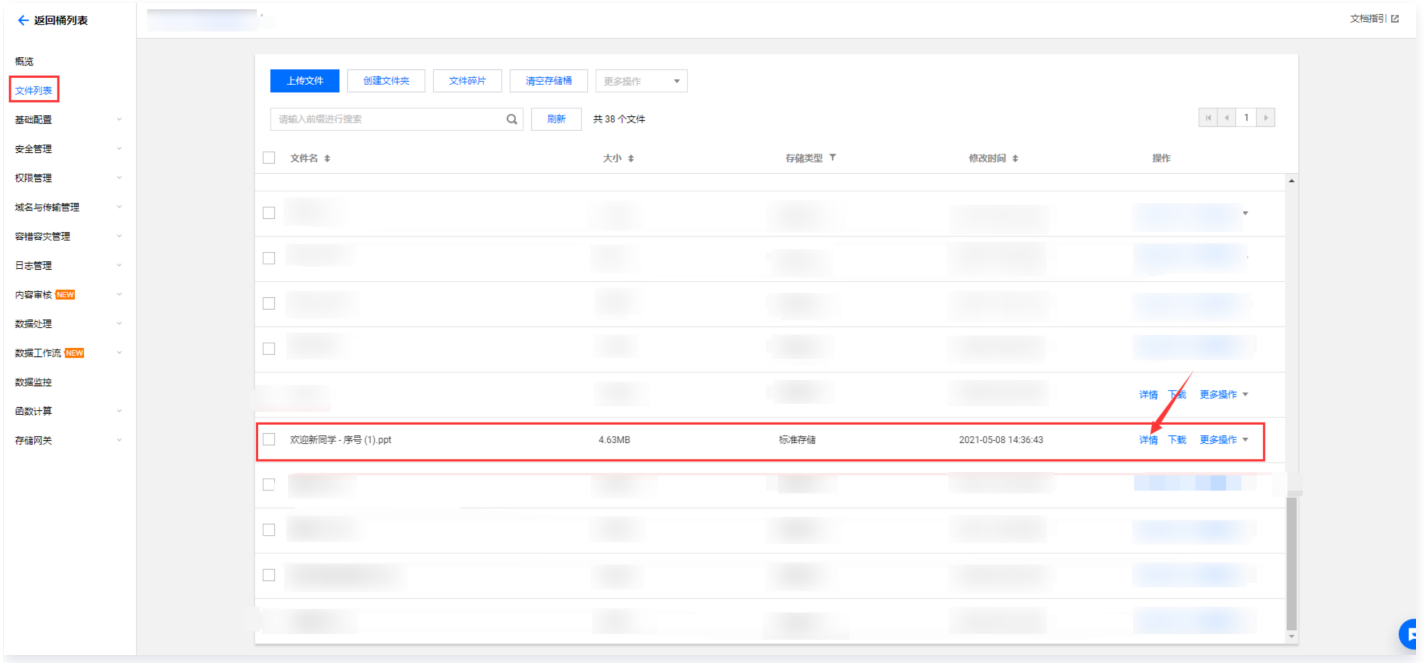
在页面中加载文档转码 SDK

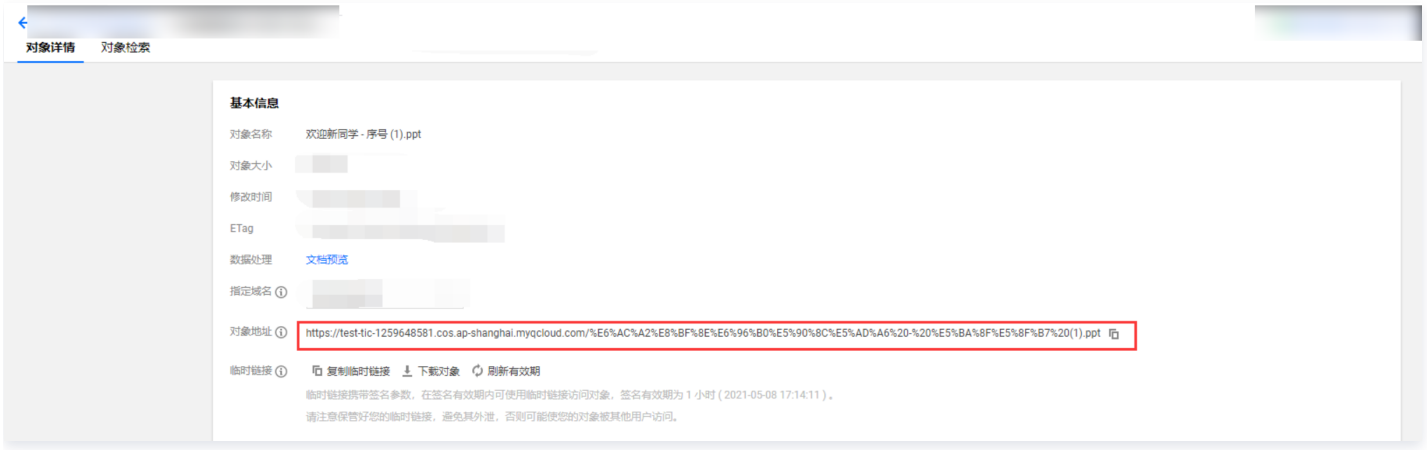
> **注意：**
> 只有 Web 端才需要加载，其他端忽略。

```
<script src="https://res.qcloudtiw.com/board/third/ci/sdk-beta-v1.1.9.js"></script>
```

获取课件的访问链接

将要转码的课件资源上传到 COS 存储桶中，并获取课件的 URL。





> **说明**
> 请确保您使用的 URL 资源是可以跨域访问的。
> - 如果您使用的是 COS 源站地址，请参考 COS 设置跨域访问 进行配置；
> - 如果您使用的是 CDN 地址，请参考 HTTP 响应头配置 进行配置。

调用白板接口，将课件的 URL 设置到白板中

- 在上一步获取到的课件 URL 中，拼接参数 `for_tiw=1` ， `for_tiw=1` 作为识别转码（备用）方案的标识，一定需要加上。
- 调用互动白板接口 `addTranscodeFile` 将 URL 设置到白板中。

> **注意**
> URL 中不需要带 COS 文档预览功能的 `ci-process=doc-preview&dstType=html` 这些参数，只需要保留 `for_tiw=1` 和您的业务参数。

```
// 以 web 端代码为例
// 示例三：2.6.0版本起，支持cos转码
teduBoard.addTranscodeFile({
    title: "欢迎新同学", // 文件标题
    url: "https://test-tic-1259648581.cos.ap-
shanghai.myqcloud.com/%E6%AC%A2%E8%BF%8E%E6%96%B0%E5%90%8C%E5%AD%A6%20-
%20%E5%BA%8F%E5%8F%B7%20(1).ppt?for_tiw=1" // cos资源的url，且携带for_tiw=1的参数
}, true)
```

> **说明**
> 完成以上两步，转码添加到互动白板中的工作就已经完成了。

支持动态转码格式

目前互动白板中只支持：ppt(x) 。

支持的字体和动画

文档转 HTML 功能支持预览多种字体，并可预览演示文件中的动画效果，具体请参考 字体支持表和动画支持表 。

特殊说明

1. 如果您是新接入用户，建议您将所有端的版本都接入至最新的2.6.0+的版本，即可体验最新的转码。
2. 如果您目前线上只有 Web 端，可以将互动白板升级至2.6.0+的版本，即可体验最新的转码。
3. 如果您目前线上还有存量移动端，客户端低于2.6.0的版本，也建议您提前接入2.6.0+版本，在存量用户都升级至2.6.0+的版本后，即可体验最新的转码。
4. 转码（备用）方案中，如果 PPT 中有未支持的动画，会采用默认的动画来代替，保证可以将 PPT 转码成功，这个是符合预期的。
5. 转码（备用）方案中，暂时不支持获取全量和获取每一页对应的步数。
6. 转码（备用）方案中，不支持客户端 snapshot 接口截图功能，支持使用 addSnapshotMark 接口用 云 API 板书生成服务 。

录制回放（备用）
录制回放概述

Last updated：2024-09-11 09:08:12

功能简介

腾讯云互动白板录制回放服务提供了将课堂中各路音视频以及白板画面分别录制为视频的能力，方便您记录每堂课的完整过程，满足课堂质量分析和学生复习回顾等业务场景。

录制模式

录制回放服务提供了两种录制模式：

- 白板实时录制模式（默认）
- 白板推流录制模式（推荐）

这两种录制模式的差异如下：

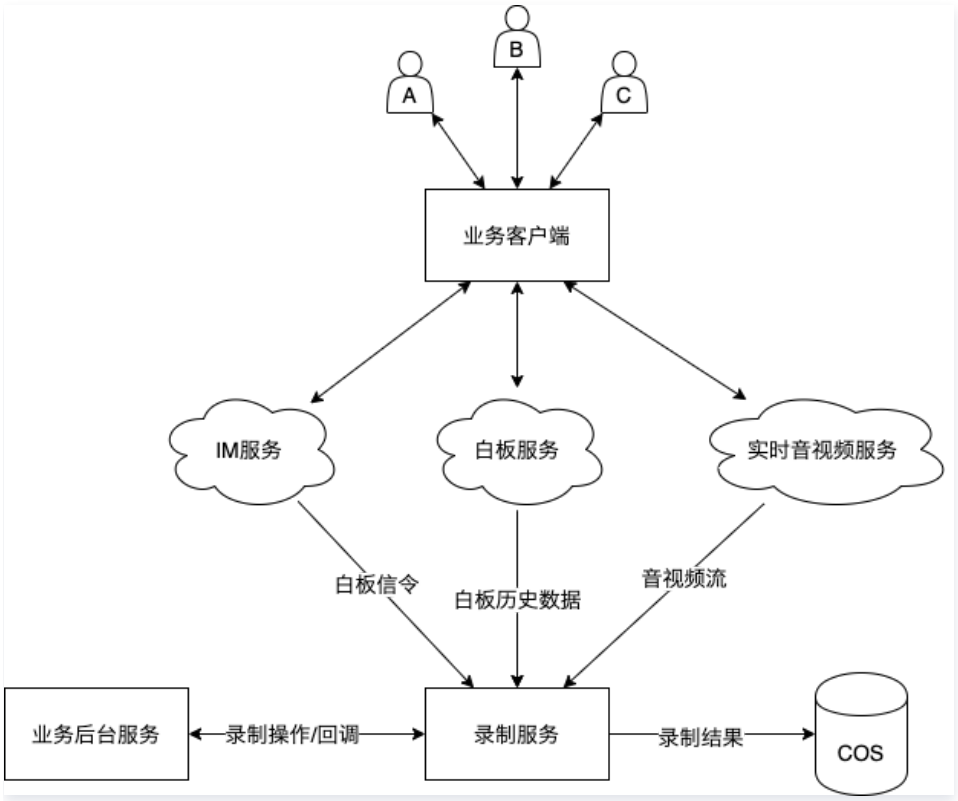
录制模式	录制方式	录制格式	录制耗时	回放方式	成本
白板实时录制模式	上课、下课时调用接口，课后自动生成录制文件	标准 MP4	课后5分钟左右可用	标准播放器	高
白板推流录制模式	上课、下课时调用接口，课后自动生成回放链接，按需发起某路视频，可以把白板流推到 TRTC 房间一起录制	标准 MP4	课后5分钟左右生成	标准播放器	低

注意：
如果没有进行**存储桶配置**，录制后的视频文件存储在**互动白板的公共存储桶**里，存储时间是**3天**。3天后，存储桶里的视频文件自动删除。建议在使用录制功能前进行 存储桶配置 ，或者在获取到录制结果后自行对录制文件进行转存。

录制原理

白板实时录制模式（默认）
实时录制模式下，录制服务会以观众的身份加入到音视频房间实时地从音视频后台拉取音视频流进行录制，同时实时接收白板操作信令在服务器上渲染成画面后进行录制，并在录制结束后快速生成回放视频文件。

此模式支持快速回放，且通常在录制结束后几分钟内就能生成录制文件，但收费较高，适合需要在短时间内获取到回放视频文件的业务。
工作原理如下所示：



白板推流录制模式（推荐）

我们推荐以白板推流的方式进行录制。详见 白板推流录制 部分的文档。

> **说明：**
> 账号 ID 和 AppID 可以在腾讯云控制台 > 账号信息 中查询。

实时录制接入流程

Last updated：2024-09-11 09:08:12

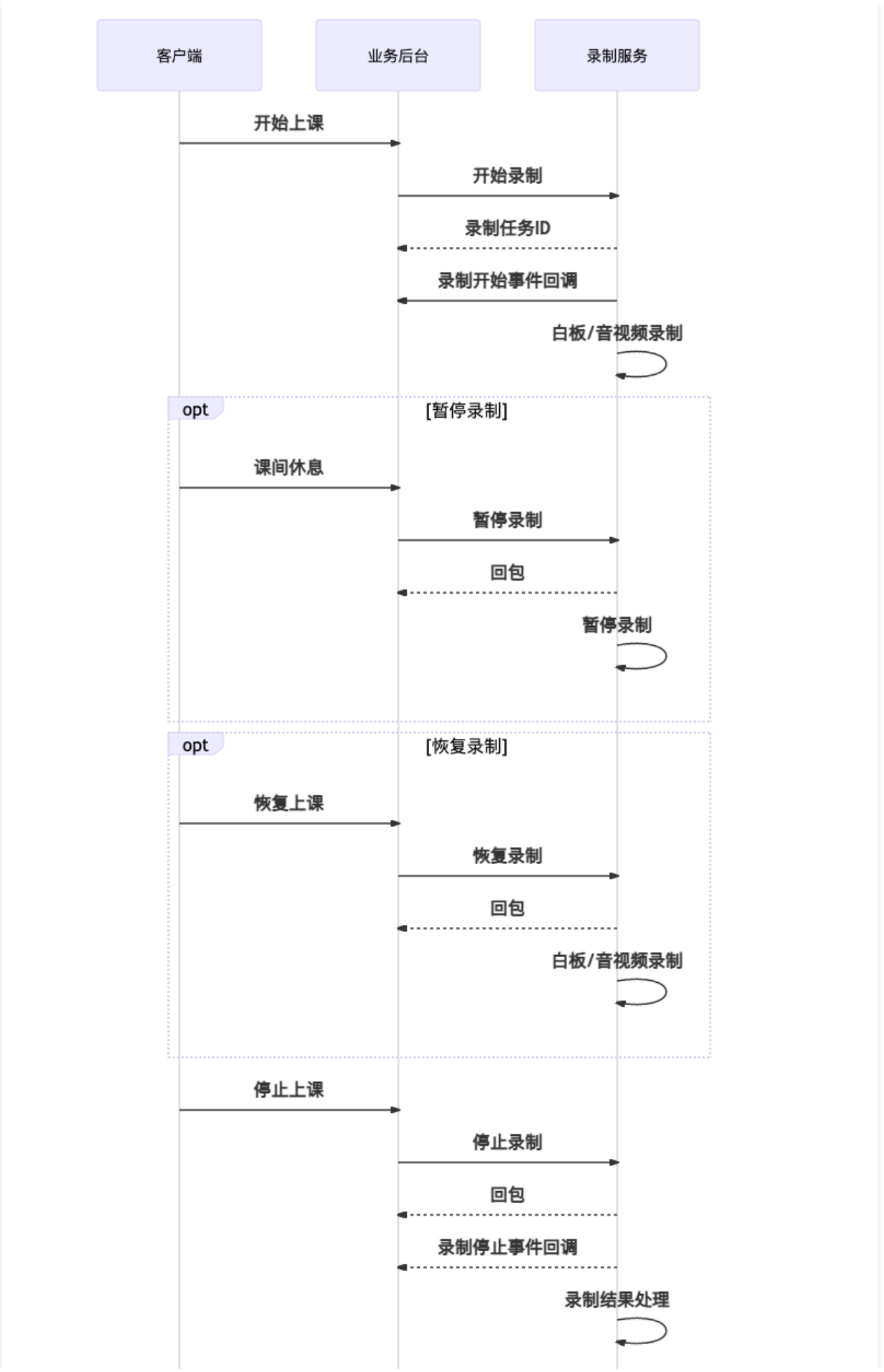
注意

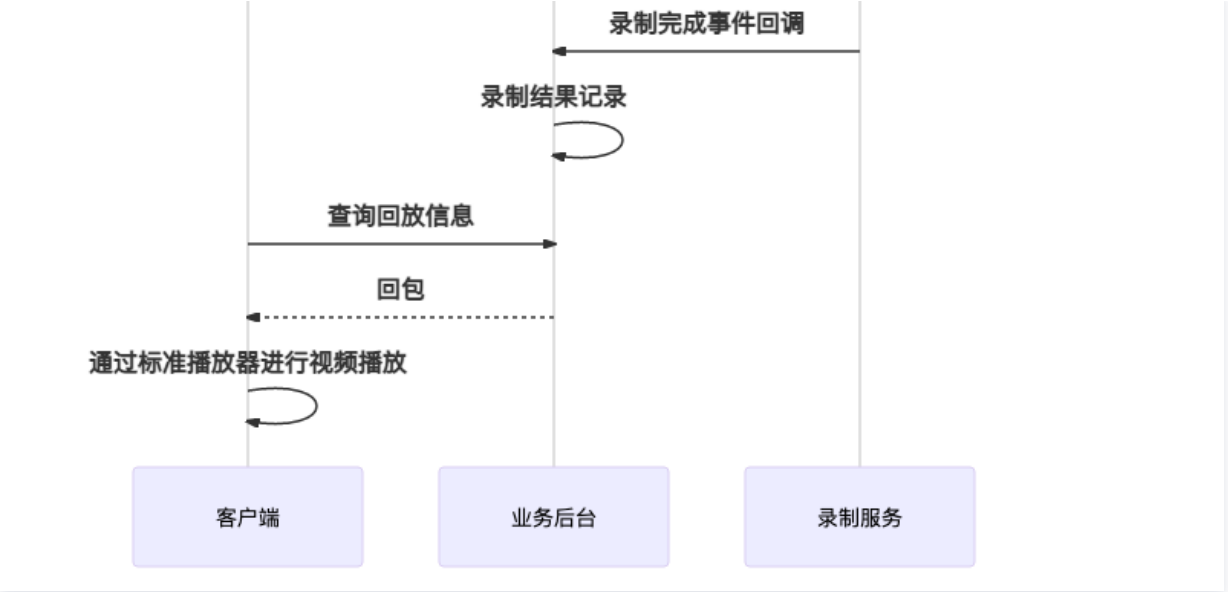
1. 当录制房间内5分钟没有任何音视频上行，且这5分钟内没有任何白板操作的时候，录制任务将自动停止录制。
2. 当录制任务暂停超过90分钟的时候，录制任务将自动停止录制。
3. 当录制任务开始后，超过24小时没有调用结束录制，录制任务将自动停止录制。
4. 录制结果文件只保存**3天**，3天后将被删除，建议在使用录制功能前进行 存储桶配置 ，或在获取到录制结果后自行对录制文件进行转存。

交互流程

实时录制模式

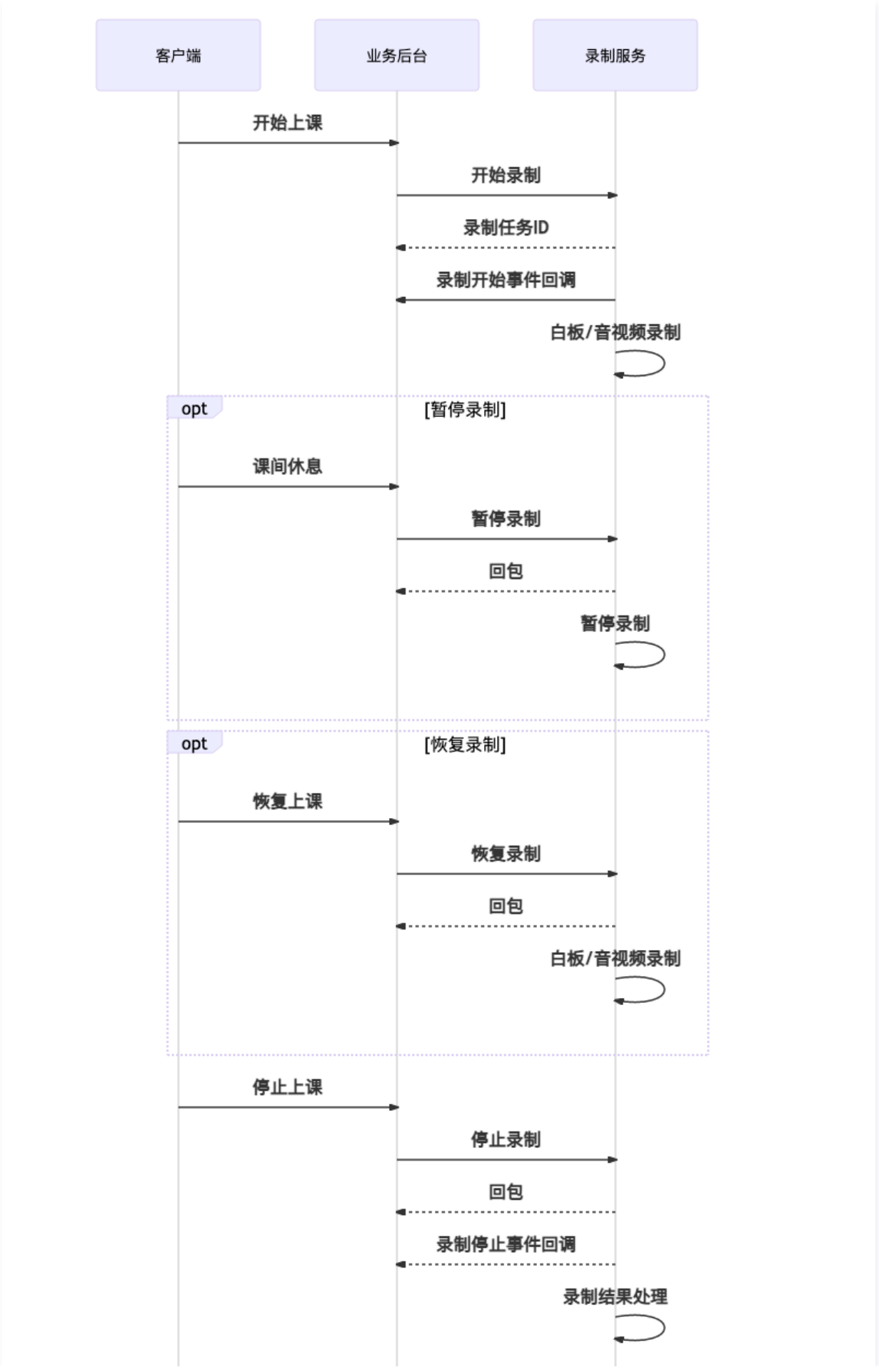
实时录制模式从开始上课到最终进行课堂回放一般经过如下几个交互过程（以配置了事件回调地址为例）：

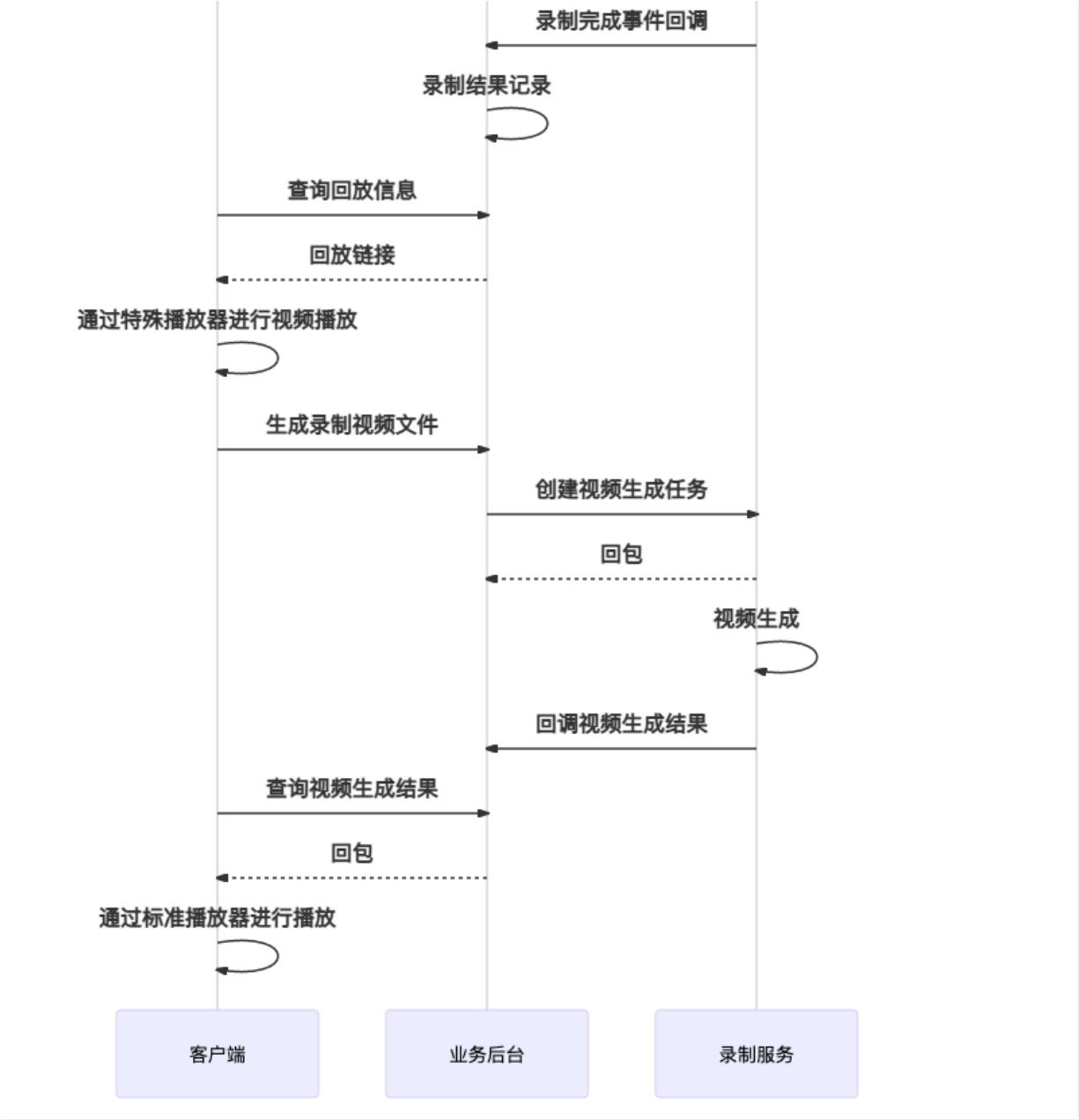




视频生成模式

视频生成模式交互流程与**实时录制模式**大体一致，区别在于回放需要使用特殊播放器，另外需要视频文件的话，可以选择重新生成视频。



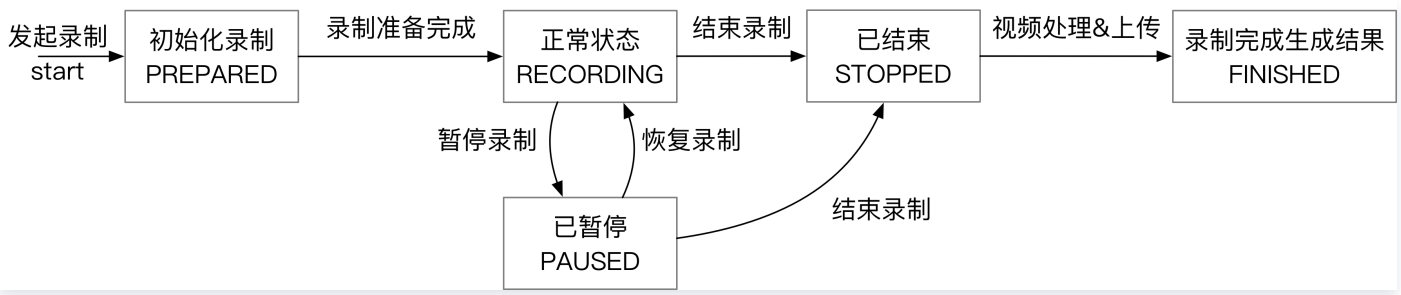


录制任务状态转换

录制任务的生命周期内存在五种状态：

- PREPARED – 表示录制任务创建成功，但录制还没有正式开始。
- RECORDING – 表示录制任务已经正式开始录制。
- PAUSED – 表示录制任务已暂停
- STOPPED – 表示录制任务已经停止录制，正在进行录制结果处理。
- FINISHED – 表示录制任务已结束

状态转换过程如下：



准备工作

存储桶配置

录制的结果文件（视频文件、录制元数据、消息暂存等）的存储依赖 对象存储 COS 服务，因此在使用录制功能前，请**务必**先进行 存储桶配置 。

> ⚠ **注意**
>
> 如果没有进行存储桶配置，则录制服务会将录制结果存储到内部公共桶，且仅保留3天，3天后将被删除。

准备 `RecordUserId` 和 `RecordUserSig`

录制服务需要加入课堂并对课堂的音视频和白板进行录制，因此需要您提供一个录制服务进房时使用的 `RecordUserId` 以及 `RecordUserSig` ，生成 `RecordUserId` 和 `RecordUserSig` 的方法请参考 如何计算 UserSig 。
为了将录制后台的 `RecordUserId` 与普通用户进行区分，我们约定 `RecordUserId` 的格式**必须**如下：

```
tic_record_user_{roomid}_{随机数}
```

其中， `{roomid}` 为您真实的房间号，假如课堂的音视频房间 `100241` ，一个合法的录制 `RecordUserId` 为 `tic_record_user_100241_100` 。
同时您需要提供 `tic_record_user_100241_100` 对应的 `RecordUserSig` 签名。

> ⚠ **注意**
>
> 1. `RecordUserSig` 签名请设置一个较长的有效期，例如1小时，避免由于签名过期导致录制失败。
> 2. 如果同一房间内需要多次发起录制，请使用不同的录制 `RecordUserId` ，否则录制用户会被强制下线而导致录制失败。

开通实时音视频云端自动录制

开通实时音视频云端录制的目的：

- `实时录制模式` 可能会因为不可抗原因导致录制出现异常，如果希望在录制结束后对异常视频进行恢复，请务必在使用 `实时录制模式` 前开通实时音视频云端自动录制。
- `视频生成模式` 依赖实时音视频的云端录制功能，如果要使用 `视频生成模式` ，请务必在使用 `视频生成模式` 前开通实时音视频云端自动录制。

配置方法：

1. 登录 实时音视频控制台 ，在左侧导航栏选择**应用管理** 。
2. 单击目标应用所在行的**功能配置** ，进入功能配置页卡。如果您还没有创建过应用，可以单击**创建应用**，填写应用名称，单击**确定**创建一个新的应用。
3. 单击**启动云端录制**右侧的  ，会弹出云端录制的设置页面。

4. 在弹出的云端录制设置页面中，录制形式选择**全局自动录制**，录制文件格式选择**MP4**。

开通实时音视频云端录制的更多设置内容可以参考实时音视频的 实现云端录制与回放 页面。

在客户端发送对时信息（视频生成模式）

> **注意**
> 1. 此小节仅在使用**视频生成模式**时需要关注，仅使用**实时录制模式**可忽略此内容。
> 2. 如果使用 TIC 进行白板接入的话，可以忽略此内容。

由于视频生成模式的音视频录制与白板录制是由不同服务录制的，需要业务侧在客户端配合调用实时音视频 SDK 接口定时地将白板对时信息写入到视频帧中，从而保证在回放时播放器能够根据对时信息对多路流进行同步播放，视频生成任务可以根据对时信息对多路流进行混流时保持白板与视频之间的音画同步。
对时信息的定义如下：

```
{
    "syncTime": 1600152855000
}
```

其中 `syncTime` 为白板时间，需要通过白板 SDK 提供的接口获取，单位为毫秒(ms)。
要完成对时信息的发送，不同平台的实现方式不一样，下边针对不同平台进行一一说明。

Android

在白板初始化完成后，启动一个定时任务，定时的通过白板 SDK 提供的 getSyncTime 方法获取白板时间，并通过 `TRTC SDK` 提供的 sendSEIMsg 接口把白板时间添加到视频帧里。
这里以 `Handler` 实现的定时任务为例，先定义一个 `Handler` 。

```
static class syncTimeHandler extends Handler {
    WeakReference<TRTCCloud> mTRTCCloud;
    WeakReference<TEduBoardController> mBoard;

    syncTimeHandler(TRTCCloud trtcCloud, TEduBoardController board) {
        mTRTCCloud = new WeakReference<>(trtcCloud);
        mBoard = new WeakReference<>(board);
    }
```

```
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);

        sendSyncTimeBySEI();
        sendEmptyMessageDelayed(0, 5000);
    }

    private void sendSyncTimeBySEI() {
        TRTCCloud trtcCloud = mTRTCCloud.get();
        TEduBoardController board = mBoard.get();

        if (trtcCloud != null && board != null) {
            long time = board.getSyncTime();
            if (time != 0) {
                String result = "";
                JSONObject json = new JSONObject();
                try {
                    json.put("syncTime", time);
                    result = json.toString();
                } catch (Exception e) {
                    e.printStackTrace();
                }
                if (!TextUtils.isEmpty(result)) {
                    trtcCloud.sendSEIMsg(result.getBytes(), 1);
                }
            }
        }
    }
}
```

在白板与 TRTC 实例初始化完成后，通过以上定义的 `Handler` 来触发定时发送对时信息，代码中的 `trtcCloud` 以及 `board` 分别为已经初始化完成的 TRTC 实例及白板控制实例，初始化操作可以参考 白板 SDK 集成文档 。

```
Handler handler = new syncTimeHandler(trtcCloud, board);
handler.sendEmptyMessage(0);
```

iOS/Mac

在白板初始化完成后，启动一个定时任务，定时的通过白板 SDK 提供的 getSyncTime 方法获取白板时间，并通过 `TRTC SDK` 提供的 sendSEIMsg 接口把白板时间添加到视频帧里。
发送对时信息的代码示例如下：

```
void syncRemoteTime:(TRTCCloud *)trtcCloud board:(TEduBoardController *)board {
    // 获取白板时间
    uint64_t syncTime = [board getSyncTime];
    NSMutableDictionary *dataDic = [NSMutableDictionary dictionary];
    [dataDic setObject:[NSNumber numberWithLongLong:syncTime] forKey:@"syncTime"];
    NSData *data = [NSJSONSerialization dataWithJSONObject:dataDic options:0 error:nil];
    [trtcCloud sendSEIMsg:data repeatCount:1];
}
```

在白板与 TRTC 实例初始化完成后，启动一个定时任务来实现定时发送对时信息，其中的 `trtcCloud` 和 `board` 分别为已经初始化完成的 TRTC 实例及白板控制实例，初始化操作可以参考 白板 SDK 集成文档 。

```
NSTimer *syncTimer = [NSTimer scheduledTimerWithTimeInterval:1 repeats:YES block:^(NSTimer *
_Nonnull timer) {
    syncRemoteTime(trtcCloud, board);
}];
```

在退出的时候记得停止已启动的定时器，避免出现内存泄漏。

```
if (syncTimer && [syncTimer isValid]) {
    [syncTimer invalidate];
    syncTimer = nil;
}
```

Windows

在白板初始化完成后，启动一个定时任务，定时的通过白板 SDK 提供的 GetSyncTime 方法获取白板时间，并通过 `TRTC SDK` 提供的 sendSEIMsg 接口把白板时间添加到视频帧里。
下边以 `C++` 代码为例，代码中的 `trtcCloud` 以及 `board` 分别为已经初始化完成的 TRTC 实例及白板控制实例，初始化操作可以参考 白板 SDK 集成文档 。
先实现一个简单的定时器：

```
#include <thread>

class Timer {
public:
    template<typename Function>
    inline void setInterval(Function func, int interval) {
        stopped_ = false;

        std::thread t([=]{
            while(true) {
                if(stopped_) return;
                std::this_thread::sleep_for(std::chrono::milliseconds(interval));
                if(stopped_) return;

                if(func) {
                    func();
                }
            }
        });

        t.detach();
    }

    inline void stop() {
        stopped_ = true;
    }
private:
    bool stopped_;
};
```

最后在白板与 TRTC 实例初始化完成后，启动定时器定时发送对时信息，其中的 `trtcCloud` 和 `board` 分别为已经初始化完成的 TRTC 实例及白板控制实例，初始化操作可以参考 白板 SDK 集成文档 。

```
#include <sstream>
```

```
Timer t;
t.setInterval([trtcCloud, board]{
    std::stringstream ss;
    ss << "{\"syncTime\":" << board->GetSyncTime() << "}";
    auto jsonStr = ss.str();

    trtcCloud->sendSEIMsg((uint8_t*)jsonStr.c_str(), (uint32_t)jsonStr.length(), 1);
}, 5000);
```

Web

web 端的 `TRTC SDK` 不提供 `SendSEIMsg` 接口，所以需要在进房的时候通过设置 `BusinessInfo` 的方法将白板服务校正过的时间戳同步给实时音视频服务的 `WebRTC` 后台服务。
在白板初始化完成后，通过白板 SDK 提供的 getSyncTime 方法获取白板时间。

```
syncTime = this.whiteBoard.getSyncTime()
```

然后在创建 `TRTC Client` 的时候，把 `syncTime` 设置到TRTC初始化参数 `bussinessInfo` 中，可参考 `TRTC SDK` 提供的 createClient 接口。

```
let param = {
    mode: 'live',
    sdkAppId: {您的 sdkAppID},
    userId: {您的用户 ID},
    userSig: {对应用户 ID 的 UserSig},
    bussinessInfo: JSON.stringify({Str_uc_params:{syncTime: syncTime}})
}

// 创建TRTC Client
this.client = TRTC.createClient(param)
```

开始录制

在需要进行录制时，例如老师学生都已经准备好开始上课，您可以使用 开始录制 接口开始录制，在请求接口时，需要使用到上一步准备好的 `RecordUserId` 和 `RecordUserSig` ，当录制开始时，如果您配置了回调地址，您将收到事件为 录制开始 的回调请求通知。

> **说明**
> 由于网络延迟等因素，发送请求后，实际录制操作将在2s左右后进行。
> 目前服务端 API 接口只支持区域广州，在调用 API 时， `Region` 参数请填写 `ap-guangzou`。

暂停录制 和 恢复录制（可选）

在上课过程中，如果您不希望录制中间一段内容（例如课间休息），并且想要将暂停前和暂停后的视频放在一个录制任务结果中，您可以使用暂停录制 接口和 恢复录制 接口实现。

> **说明**
> 由于网络延迟等因素，发送请求后，实际录制操作将在2s左右后进行。
> 目前服务端 API 接口只支持区域广州，在调用 API 时， `Region` 参数请填写 `ap-guangzou`。

停止录制

在课堂结束或者需要停止录制的时候，您可以使用 停止录制 接口通知录制服务停止当前录制，如果您配置了回调地址，录制视频处理完成后，您将收到事件为 录制停止 的回调请求通知。

> **说明**
> 由于网络延迟等因素，发送请求后，实际录制操作将在2s左右后进行。
> 目前服务端 API 接口只支持区域广州，在调用 API 时，`Region` 参数请填写 `ap-guangzou`。

获取录制结果

录制服务提供了两种方式来获取录制结果：

- **设置录制事件回调地址**
  通过 控制台 或者 设置回调地址 接口都可以完成设置录制事件回调地址。在录制任务结束后，录制服务会把录制结果回调到已设置好的地址，您可以在收到回调后根据业务需要对录制结果进行记录或者其他操作。
- **通过 查询录制任务 接口主动查询**
  在录制过程中或者停止录制后，您都可以使用 查询录制任务 接口来查询录制任务的具体信息。

解析录制任务结果

当您主动查询录制进度时收到 `Status` 参数值为 "FINISHED" 或者收到录制完成回调时，您可以拿到录制结果（一个 JSON 串），其格式如下：

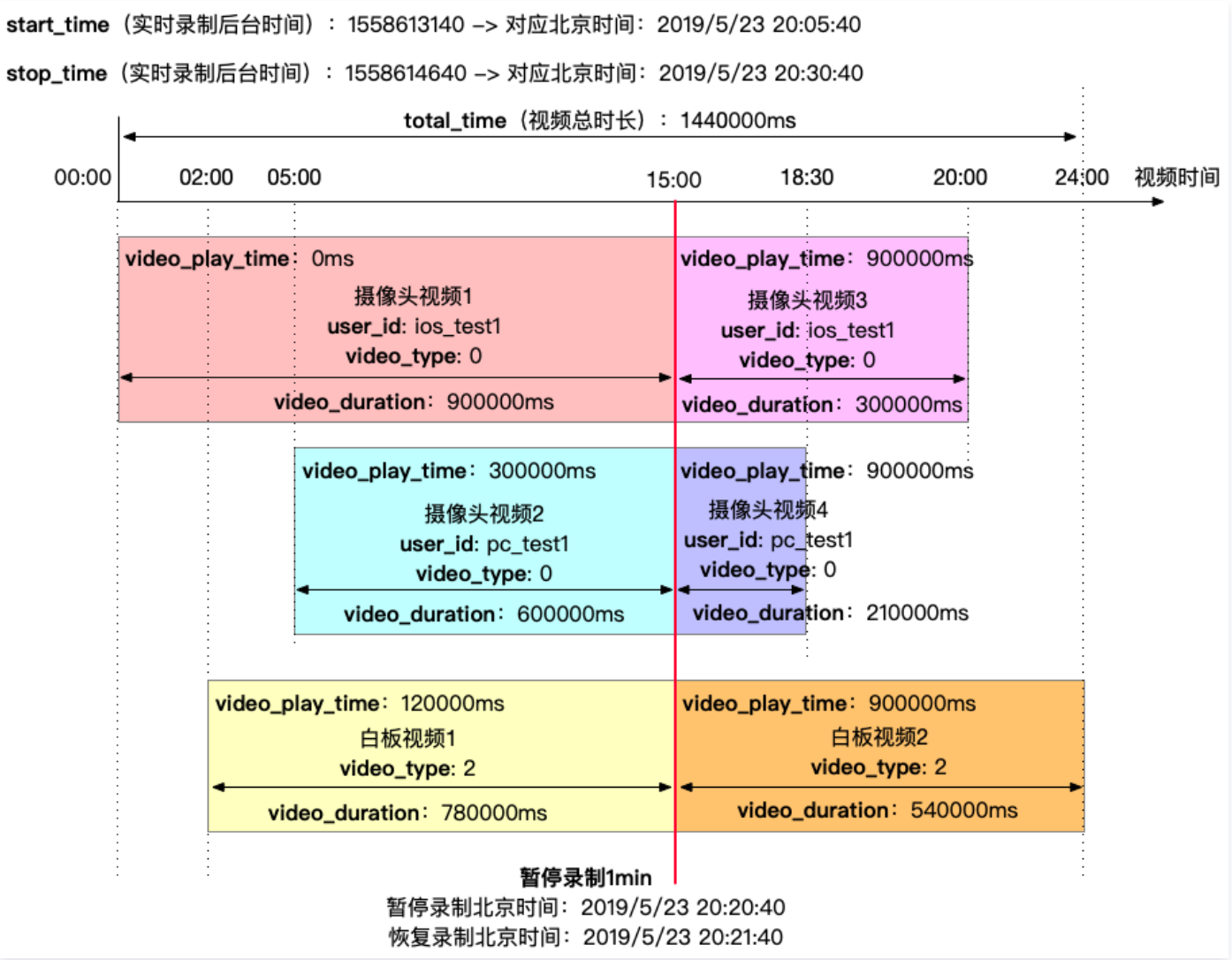
参数名	类型	描述
RoomId	Integer	房间号
GroupId	String	白板的群组 ID
RecordUserId	String	录制所使用的 UserId
RecordStartTime	Integer	录制开始时间，Unix 时间戳，单位秒
RecordStopTime	Integer	录制停止时间，Unix 时间戳，单位秒
TotalTime	Integer	回放视频总时长（单位：毫秒）
ReplayUrl	String	视频回放地址，仅适用于 `视频生成模式`，需要配合信令播放器进行回放。可以参考 视频回放 中的 `视频生成模式` 了解更多的细节
VideoInfos	Array of VideoInfo	录制视频列表

以下为一个录制结果 JSON 串示例：

```
{
    "RoomId":1234,
    "GroupId":"1234",
    "RecordStartTime":1558613140,
    "RecordStopTime":1558614640,
    "TotalTime": 1440000,
    "VideoInfos":[
        {
            "VideoPlayTime":0,
            "VideoSize":13151,
            "VideoFormat":"mp4",
            "VideoDuration":900000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/oM86K7X3Ig8b.mp4",
            "VideoId":"5285890781570653827",
```

```
            "VideoType":0,
            "UserId":"ios_test1"
        },
        {
            "VideoPlayTime":300000,
            "VideoSize":3756,
            "VideoFormat":"mp4",
            "VideoDuration":600000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/oM86K7X3IsdfA.mp4",
            "VideoId":"5285890781570653828",
            "VideoType":0,
            "UserId":"pc_test1"
        },
        {
            "VideoPlayTime":120000,
            "VideoSize":1241,
            "VideoFormat":"mp4",
            "VideoDuration":780000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/52lk3KA0A562.mp4",
            "VideoId":"5285890781570653830",
            "VideoType":2,
            "UserId":""
        },
        {
            "VideoPlayTime":900000,
            "VideoSize":13151,
            "VideoFormat":"mp4",
            "VideoDuration":300000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/oM86K7X3Ig63.mp4",
            "VideoId":"5285890781570653841",
            "VideoType":0,
            "UserId":"ios_test1"
        },
        {
            "VideoPlayTime":900000,
            "VideoSize":3756,
            "VideoFormat":"mp4",
            "VideoDuration":210000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/oM86K7X3Isd15.mp4",
            "VideoId":"5285890781570653842",
            "VideoType":0,
            "UserId":"pc_test1"
        },
        {
            "VideoPlayTime":900000,
            "VideoSize":1241,
            "VideoFormat":"mp4",
            "VideoDuration":540000,
            "VideoUrl":"http://1253488539.vod2.myqcloud.com/52lk3KA0A512.mp4",
            "VideoId":"5285890781570653843",
            "VideoType":2,
            "UserId":""
        }
    ]
}
```

此 JSON 对象表示课堂录制产生了6个视频文件，其中，在过程中暂停录制了1分钟，之后恢复录制，因此最后产生了6段视频，这6个视频文件在时间轴上的排列如下图所示：



视频回放

实时录制模式

实时录制模式的录制结果是标准的mp4视频文件，直接使用标准播放器进行播放即可。

视频生成模式

视频生成模式进行视频回放的时候，有两种方式：

1. 重新生成视频后，使用标准播放器进行播放。
2. 使用特殊播放器对录制结果里的回放链接进行播放（**推荐**）。

下边详细说明第2种方式的使用方法。

在停止录制并拿到回放链接后，您需要使用特殊的播放器才能对回放链接进行回看，我们提供了基于 Web 页面的播放器供您使用，您只需按如下所示拼接 URL，即可在浏览器内观看回放。

```
https://sdk.qcloudtiw.com/web/replay/index.html?url=回放链接&showChatMessages=1
```

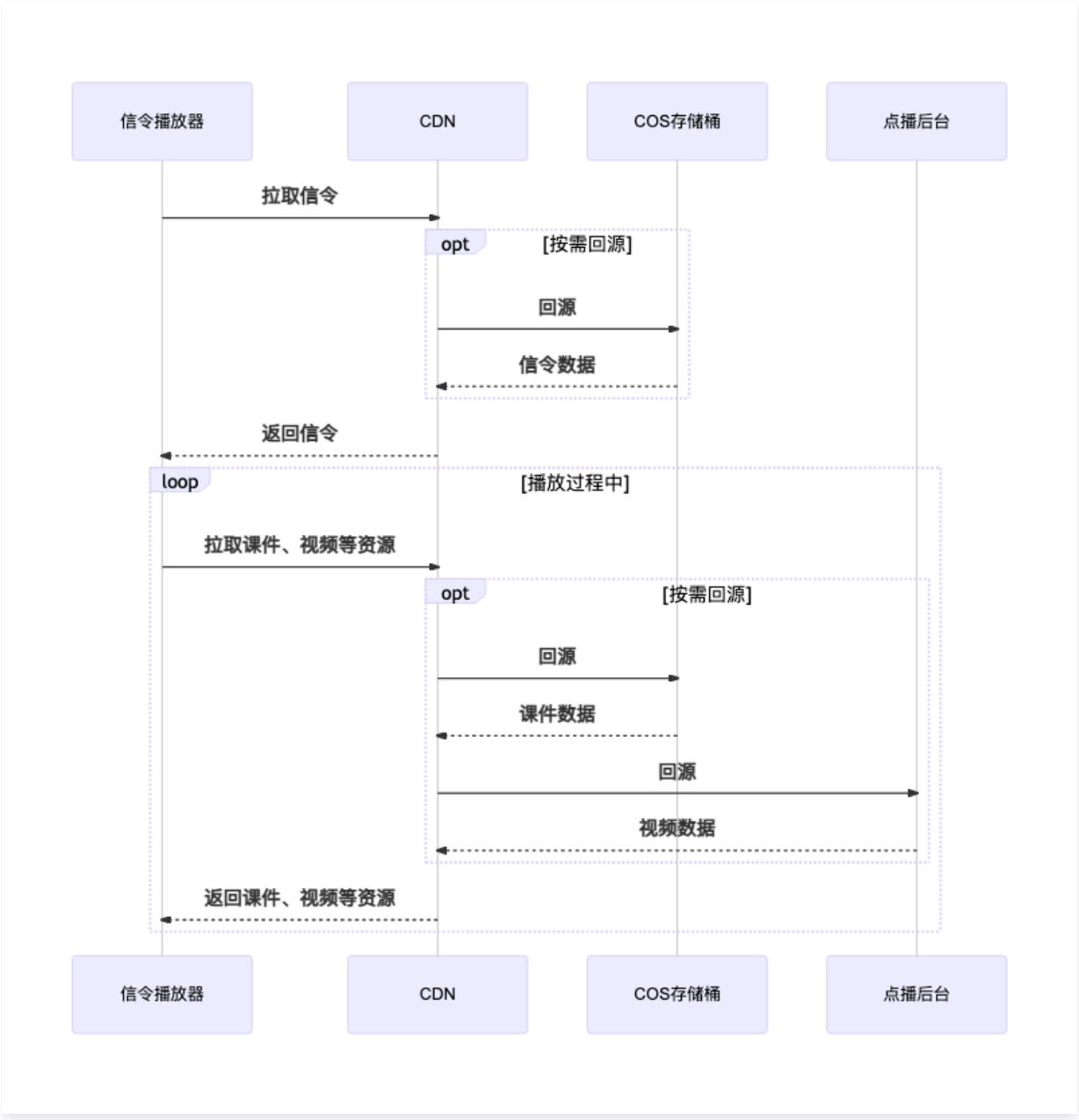
其中，URL 参数指向回放链接，showChatMessages 参数为1表示回放时需要展示 IM 聊天消息。

目前，我们的播放器对各平台浏览器的适配情况如下：

平台	支持的浏览器及其最低版本	已知问题

Windows	Chrome、Microsoft Edge	无
macOS	Chrome、Safari	无
Android	Chrome、系统浏览器、QQ、微信、企业微信	无
iOS	Safari、QQ、微信、企业微信	无

信令播放器回放工作原理如下所示：



重新生成视频（视频生成模式）

> **注意**
> 此小节仅在使用 `视频生成模式` 时需要关注，仅使用 `实时录制模式` 可忽略此内容。

由于视频生成模式在录制的过程中不会自动生成视频文件，如果需要视频文件，可以选择在录制结束后，通过视频生成相关接口来通知录制服务生成相应视频。

创建视频生成任务

> **注意**
> 创建成功的视频生成任务会在视频房间内所有人都退出后才会开始进行视频生成。

您可以使用 创建视频生成任务 接口来通知录制服务进行生成视频，在请求的时候可以选择使用不一样的混流布局来生成不一样的混流视频。

> **说明**
> - 由于网络延迟等因素，发送请求后，实际录制操作将在2s左右后进行。
> - 目前服务端 API 接口只支持区域广州，在调用 API 时，Region 参数请填写 `ap-guangzhou` 。

获取视频生成结果

在视频生成任务完成后，您如果设置了回调地址，录制服务会把视频生成结果回调到指定的地址。另外，您也可以使用 查询视频生成任务 接口来主动查询视频生成结果。
两种方式获取到的视频生成结果数据格式与字段含义与**实时录制模式**的结果相同，具体解析可参考 解析录制任务结果 。

> **说明**
> - 由于网络延迟等因素，发送请求后，实际录制操作将在2s左右后进行。
> - 目前服务端 API 接口只支持区域广州，在调用 API 时， `Region` 参数请填写 `ap-guangzhou` 。

混流录制

Last updated：2024-09-11 09:08:12

功能简介

如果您需要混流录制功能，请按照 模板 提交工单联系腾讯云互动白板客服人员。

> **注意：**
> 1. 录制结果的视频文件存储依赖对象存储 COS，使用录制功能前，请先进行 存储桶配置 。
> 2. 混流录制是在单流录制基础上进行的，混流结束后生成3个类型视频：白板 + 音视频 + 混流视频。混流录制的时长不包含白板、音视频等单流录制视频时长。如果开启了混流录制，视频结果里会包含白板与音视频单流视频，也会包含混流视频，各个类型的视频单独计算时长。

腾讯云互动白板录制服务为您提供了视频混流功能，根据您设定好的混流布局同步的将各路音视频以及白板画面混流成**一个视频**额外附加在录制结果里，方便您记录每堂课的完整过程，满足课堂质量分析和学生复习回顾等业务场景。

如何使用混流录制功能

使用混流功能需要在调用开始录制接口的时候额外提供 `MixStream` 及 `Extra.N` 这两个参数。具体参数定义请查看 开始录制接口文档 。
通过 接入流程 了解如何使用录制功能。

> **说明：**
> 目前服务端 API 接口只支持广州区域，在调用 API 时， `Region` 参数请填写 `ap-guangzou`。

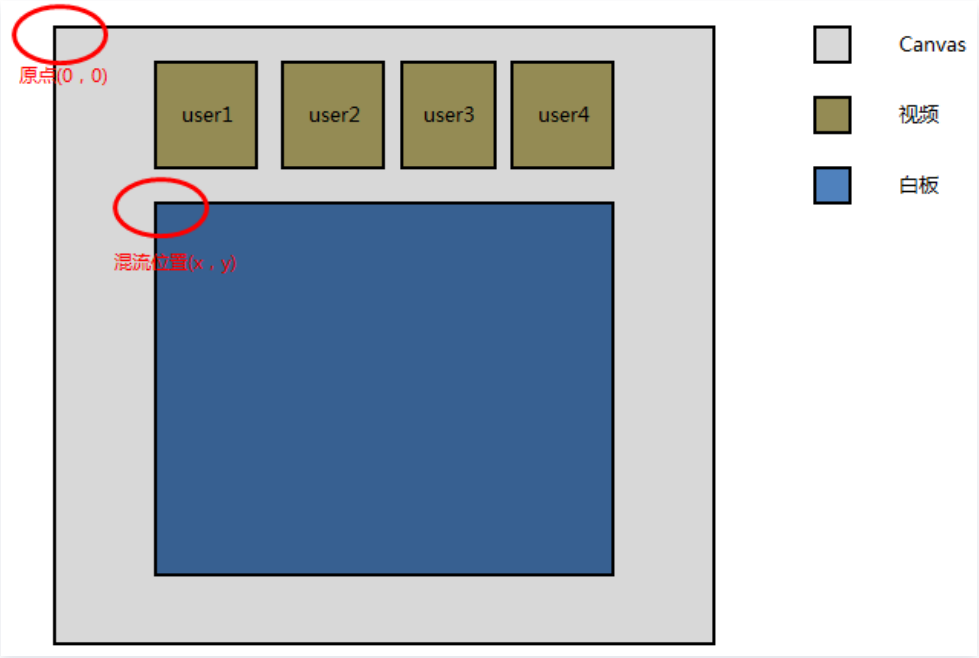
混流录制布局

内置混流布局

内置混流布局支持以下两种模板：

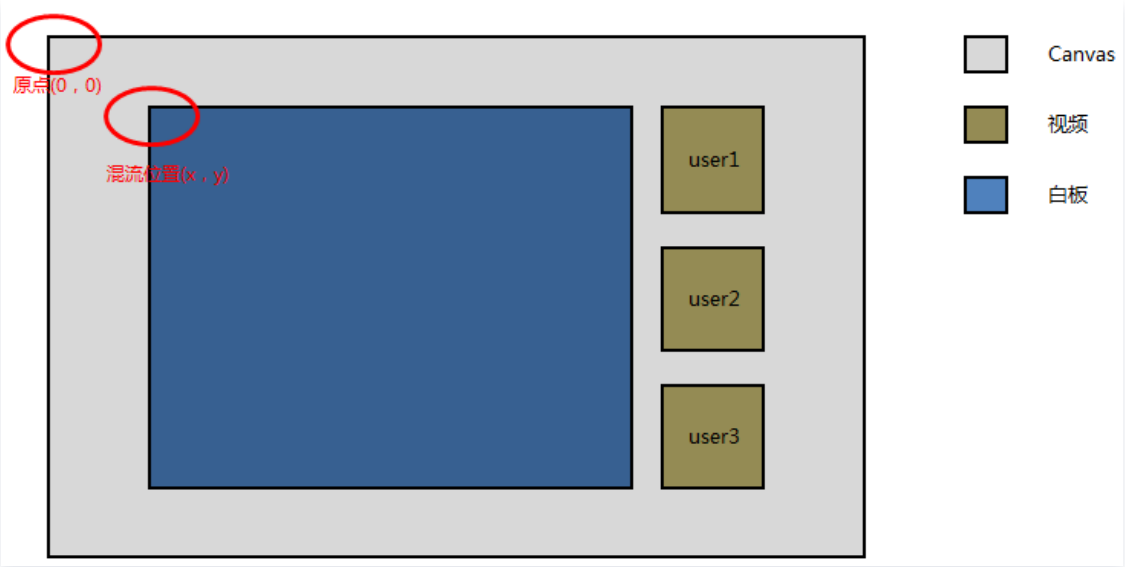
模板1（ModelId = 1）

- canvas：1320 x 1224
- 白板：1280 x 960
- 用户视频：320 x 240
- 视频之间留白：8



模板2（ModelId = 2）

- canvas：1624 x 1000
- 白板：1280 x 960
- 用户视频：320 x 240
- 视频之间留白：8



自定义混流布局：

> **注意**
>
> 1. 参与混流的每一路流布局区域不能超出画布 Canvas 的区域。
> 2. `InputStreamList` 中至少包含一路流的布局位置，否则录制的时候不会进行混流录制。
> 3. `InputStreamList` 中提供的布局位置，如果指定了 `InputStreamId`，则表示这个位置由指定的用户视频独占。
> 4. `InputStreamId` 的匹配规则为前缀匹配，如果房间内存在多个用户 ID 的前缀与指定的 `InputStreamId` 一致，则最后混流结果可能出现多个画面同时抢占一个位置的现象。

自定义混流需要在发起开始录制中输入参数 CustomLayout，以内置混流布局模板2的自定义混流参数为例：

```
{
    "Canvas":{
        "LayoutParams":{
            "Width":1624,
            "Height":1000
        },
        "BackgroundColor":"#34363B"
    },
    "InputStreamList":[
        {
            "InputStreamId":"tic_record_user",
            "LayoutParams":{
                "ZOrder":1,
                "X":8,
                "Y":8,
                "Width":1280,
                "Height":960
            }
        },
```

```
        {
            "InputStreamId":"tic_substream",
            "LayoutParams":{
                "ZOrder":2,
                "X":8,
                "Y":8,
                "Width":1280,
                "Height":960
            }
        },
        {
            "InputStreamId":"",
            "LayoutParams":{
                "ZOrder":3,
                "X":1296,
                "Y":8,
                "Width":320,
                "Height":240
            }
        },
        {
            "InputStreamId":"",
            "LayoutParams":{
                "ZOrder":3,
                "X":1296,
                "Y":256,
                "Width":320,
                "Height":240
            }
        },
        {
            "InputStreamId":"",
            "LayoutParams":{
                "ZOrder":3,
                "X":1296,
                "Y":504,
                "Width":320,
                "Height":240
            }
        },
        {
            "InputStreamId":"",
            "LayoutParams":{
                "ZOrder":3,
                "X":1296,
                "Y":752,
                "Width":320,
                "Height":240
            }
        }
    ]
}
```

申请混流开通模板

公司名：

```
账号 ID：
AppID：
联系人姓名：
联系电话：
行业类型：
预计并发用户数：
预计并发房间数：
预计上线时间：
```

账号 ID 和 AppID 可以在腾讯云控制台 > 账号信息 中查询。

常见问题

Last updated：2024-09-11 09:08:12

互动白板是否支持云端录制？是否支持与实时音视频混流录制？

互动白板支持云端实时录制功能，可以录下白板画面。同时也支持混流录制，将实时音视频与白板画面混流录制。

实时录制用户不合法该怎么解决？

为了将录制后台的 `UserId` 与普通用户进行区分，我们约定 `UserId` 必须为 `tic_record_user_{roomid}_{随机数}` 的形式。假如课堂的音视频房间 `100241` ，一个合法的实时录制 `UserId` 为 `tic_record_user_100241_100` 。请将 `{roomid}` 替换成您真实的音视频房间号，并检查您提供的 UserId 是否符合如上规则。

实时录制为什么会自动结束了？

房间内5分钟没有音视频上行及白板操作，以及暂停超过90分钟会导致实时录制自动停止。如果只是暂时停止推流，请调用暂停接口，如果录制暂停时间超过90分钟，请调用停止录制接口，在需要恢复录制的时候再次开始录制。
可以在发起请求时，通过 AutoStopTimeout 接口修改自动停止录制的时间。

实时录制没有录制到白板操作？

这种情况一般是以下问题导致：

1. 发起录制时提供的录制用户 ID 与其他用户 ID 重复，且在录制期间在其他设备登录了这个录制用户，导致录制服务被踢。
   解决办法：确保录制用户 ID 是唯一的，且不会被其他人使用。
2. 白板群组加群选项不正确（例如，设置了加群需要审批、不允许任何人加群等），导致录制用户无法加入到群组中，从而导致无法录制白板。
   解决办法：在创建群组时，将群组加群选项设置为允许任何人加群。

如何设置只录制白板，而不需要录制其他的音视频？

这个需求可以通过开始录制接口中 `RecordControl` 参数来实现，将全局录制开关设置为禁止，然后再将详细设置中的白板录制打开，可参考如下示例：

```
"RecordControl":{
    "Enabled": true,
    "DisableRecord": true,
    "DisableAudio": false,
    "PullSmallVideo": false,
    "StreamControls":[
        {
            "StreamID":"tic_record_user",
            "DisableRecord": false,
            "DisableAudio": false,
            "PullSmallVideo": false
        }
    ]
}
```

事件通知
事件通知综述

Last updated：2024-09-11 09:08:12

发起文档转码、实时录制等操作需要一段时间才能执行完，您可以调用服务端 API 注册回调接口，互动白板在文档转码进度发生变化、实时录制开始等事件触发时，可以通过回调接口及时通知到 App。
互动白板支持以下几种事件通知：

归类	相关 API 接口	事件通知
文档转码	SetTranscodeCallback SetTranscodeCallbackKey	文档转码进度改变
		文档转码结束
实时录制	SetOnlineRecordCallback SetOnlineRecordCallbackKey	实时录制开始
		实时录制结束
		实时录制长时间暂停告警
		实时录制长时间暂停自动结束
白板推流	SetWhiteboardPushCallback SetWhiteboardPushCallbackKey	推流开始
		推流结束

事件回调鉴权

您的后台在收到实时录制/文档转码的回调的时候，可以通过校验签名是否合法，进而确认回调消息是否确实来自腾讯云后台，签名算法如下：

```
Sign = md5(CallbackKey+ExpireTime)
```

- CallbackKey：鉴权密钥，您可以通过互动白板控制台或者调用实时录制（SetOnlineRecordCallbackKey）或者文档转码（SetTranscodeCallbackKey）的 API 接口设置。
- ExpireTime：签名过期时间，如果一条消息通知中的 expire_time 值所指定的时间已经过期，则可以判定这条通知无效，进而可以防止网络重放攻击。格式为十进制 UNIX 时间戳，即从1970年1月1日（UTC/GMT 的午夜）开始所经过的秒数。

例如：

```
CallbackKey = Xz4ZgayTr7rMgWQrH
ExpireTime = 1588040109

Sign = md5(Xz4ZgayTr7rMgWQrH1588040109) = a2dabb362a9b811c0e26953a6276a41c
```

如果设置了回调密钥，在您收到回调请求时，会携带 `ExpireTime` 和 `Sign` 参数，您可以依据这两个值和您设置的回调密钥进行计算校验请求是否来自于腾讯云。

事件回调协议

请求：互动白板发起的事件回调的形式是 HTTP POST 请求，请求体为 JSON 格式，内容为：

参数名称	类型	描述
Timestam	Intege	事件生成的 Unix 时间戳，单位秒

p	r	
SdkAppId	Integer	触发时间的互动白板应用 SdkAppId
ExpireTime	Integer	签名的过期时间的 Unix 时间戳，单位秒，如果当前时间晚于过期时间，后台可以判断该请求不合法
Sign	String	回调签名，您可以根据 事件回调鉴权 中描述的方法校验签名是否匹配以校验该请求是否来自腾讯云
EventType	String	事件类型
EventData	JSON	事件具体信息，在事件文档中详细描述

例如，文档转码发生进度改变的回调格式为：

```
{
    "EventData": {
        "ResultUrl": "",
        "Pages": 21,
        "Progress": 10,
        "Resolution": "960x540",
        "TaskId": "gaqvbm16jr2q4uhm23rb",
        "Title": "示例.pptx"
    },
    "EventType": "PPT2H5ProgressChanged",
    "ExpireTime": 1588040109,
    "SdkAppId": 1400000001,
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1590045522
}
```

应答：HTTP STATUS CODE = `200`，服务端忽略应答包具体内容，为了协议友好，建议客户应答内容携带 JSON： `{"error_code":0}`
事件通知具备重试能力，重试间隔为5秒，总计重试5次。为了避免重试对您的服务器以及网络带宽造成冲击，请保持正常回包。触发重试条件如下：

- 长时间（10秒）未回包应答。
- 应答 HTTP STATUS 不为 `200`。

白板推流事件

Last updated：2024-09-11 09:08:12

推流开始

事件名称

WhiteboardPushStarted

事件说明

当白板推流服务准备完成，开始推流时通知 App 后台。

EventData 结构定义

参数名称	类型	描述
TaskId	String	推流已开始的白板推流 TaskId
RoomId	Integer	白板的房间 Id
PushUserId	String	当前任务用于白板推流服务进房的用户 Id
PushStartTime	Integer	实际开始推流时间，Unix 时间戳，单位秒

示例

```
{
    "EventData": {
        "PushStartTime": 1610545438,
        "PushUserId": "tic_push_user_1253_01",
        "RoomId": 1253,
        "TaskId": "bj0mt2l23osdj300hl30"
    },
    "EventType": "WhiteboardPushStarted",
    "ExpireTime": 1590046391,
    "SdkAppId": 1400000001,
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1575545412
}
```

推流结束

事件名称

WhiteboardPushStopped

事件说明

当用户主动调用停止推流接口或者由于其他原因自动停止推流时通知 App 后台。

EventData 结构定义

参数名称	类型	描述
TaskId	String	推流结束的白板推流任务 ID
RoomId	Integer	房间号
GroupId	String	白板的群组 ID
PushUserId	String	推流用户 ID
FinishReason	String	推流结束原因 • AUTO：房间内长时间没有音视频上行及白板操作导致自动停止推流

		• USER_CALL：主动调用了停止推流接口 • EXCEPTION：推流异常结束
PushStartTime	Integer	实际开始推流时间，Unix 时间戳，单位秒
PushStopTime	Integer	实际停止推流时间，Unix 时间戳，单位秒
IMSyncTime	Integer	白板推流首帧对应的 IM 时间戳，可用于录制回放时 IM 聊天消息与白板推流视频进行同步对时
ExceptionCnt	Integer	推流过程中出现异常的次数
Error.Code	String	如果白板推流发生错误会有该字段，错误码为 FailedOperation.Record
Error.Message	String	如果白板推流发生错误的具体错误描述

示例


```
{
  "EventData": {
    "TaskId": "ghucnligqtgtvk2624mb",
    "RoomId": 880528,
    "GroupId": "880528",
    "PushUserId": "tic_push_user_880528_test-01",
    "FinishReason": "USER_CALL",
    "PushStartTime": 1568949369,
    "PushStopTime": 1568949392,
    "IMSyncTime": 1568949369,
    "ExceptionCnt": 0
  },
  "EventType": "WhiteboardPushStopped",
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412
}
```

文档转码事件

Last updated：2024-12-13 21:18:52

转码进度

事件名称

TranscodeProgressChanged

事件说明

当文档转码进度发生改变时通知 App 后台。

EventData 结构定义：

参数名称	类型	描述
TaskId	String	发生进度改变的文档 TaskId
Progress	Integer	当前转码进度
Resolution	String	文档分辨率，在文档转码服务解析出文件分辨率之前该值为空字符串
Title	String	文档标题，在文档转码服务解析出文件标题之前该值为空字符串
Pages	Integer	文档总页数，在文档转码服务解析出文件总页数之前该值为0

示例

```
{
  "EventType": "TranscodeProgressChanged",
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "Progress": 24,
    "Resolution": "1024x768",
    "Title": "测试.ppt",
    "Pages": 16
  }
}
```

转码结束

事件名称

TranscodeFinished

事件说明

当文档转码完成时通知 App 后台。

EventData 结构定义：

参数名称	类型	描述
TaskId	String	转码完成的文档 TaskId
Resolution	String	文档分辨率
Title	String	文档标题

Pages	Integer	文档总页数
ResultUrl	String	文档转码结果
ThumbnailUrl	String	缩略图 Url
ThumbnailResolution	String	缩略图分辨率
CompressFileUrl	String	转码结果打包压缩文件的下载 Url
Error.Code	String	如果文档转码发生错误会有该字段，具体错误码描述请参考接口文档 查询文档转码任务
Error.Message	String	如果文档转码发生错误的具体错误描述

示例

示例1 转码成功的回调

```
{
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventType": "TranscodeFinished",
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "Resolution": "1024x768",
    "Title": "测试.ppt",
    "Pages": 16,
    "ResultUrl": "https://transcode-result/0agdnligqtgtvkm65emb/index.html",
    "ThumbnailUrl": "https://transcode-thumbnal/0agdnligqtgtvkm65emb/",
    "ThumbnailResolution": "793x1122",
    "CompressFileUrl": ""
  }
}
```

示例2 转码失败的回调

```
{
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventType": "TranscodeFinished",
  "EventData": {
    "Error": {
      "Code": "InvalidParameter.UrlFormatError",
      "Message": "文档下载 URL 不合法"
    },
    "TaskId": "bj0mt2l23osdj300hl30",
    "Resolution": "",
    "Title": "",
    "Pages": 0,
    "ResultUrl": "",
    "ThumbnailUrl": "",
    "ThumbnailResolution": "",
    "CompressFileUrl": ""
  }
```

```
}
```

PPT 检测结束

事件名称

PPTCheckFinished

事件说明

当 PPT 检测结束完成时通知 App 后台。

EventData 结构定义：

参数名称	类型	描述
TaskId	String	任务 Id
IsOK	bool	PPT 文件是否正常
ResultUrl	String	修复后的 PPT URL，只有创建任务时参数 AutoHandleUnsupportedElement=true，才有返回值
Slides	Array of ErrSlide	错误 PPT 页面列表

ErrSlide 结构定义：

参数名称	类型	描述
Page	String	页面
Errs	Array of ErrInfo	错误元素列表

ErrInfo 结构定义：

参数名称	类型	描述
Name	String	元素名称
Type	int	0：不支持的墨迹类型 1：自动翻页 2：已损坏音视频 3：不可访问资源 4：只读文件 5：不支持的元素编辑锁定状态 6：可能有兼容问题的字体 7：设置了柔化边缘的 GIF 图片 8：存在不兼容的空格下划线 9：存在设置了分段动画的数学公式和文本混合内容 10：存在设置了分段动画的渐变色文本 11：存在不兼容的分散对齐方式 12：存在不兼容的多倍行距设置 13: 存在带有特殊符号内容的datetime类型的a:fld标签元素
Detail	String	错误详情

示例

示例1 检测无异常的回调

```
{
```

```
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventType": "PPTCheckFinished",
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "IsOK": true
  }
}
```

示例2 检测异常的回调

```
{
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventType": "PPTCheckFinished",
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "IsOK": false,
    "ResultUrl": "https://xxx/xxx/测试_fixed.ppt",
    "Slides": [
      {
        "Page": "幻灯片5",
        "Errs": [{
          "Name": "Ink 1",
          "Type": 0,
          "Detail": "wps墨迹"
        }]
      }
    ]
  }
}
```

示例3 检测失败的回调

```
{
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412,
  "EventType": "PPTCheckFinished",
  "EventData": {
    "Error": {
      "Code": "InvalidParameter.UrlFormatError",
      "Message": "文档下载 URL 不合法"
    },
    "TaskId": "bj0mt2l23osdj300hl30"
  }
}
```

实时录制事件

Last updated：2024-09-11 09:08:12

录制开始

事件名称

OnlineRecordStarted

事件说明

当实时录制服务准备完成，开始录制时通知 App 后。

EventData 结构定义

参数名称	类型	描述
TaskId	String	录制已开始的实时录制 TaskId
RoomId	Integer	实时录制的房间 ID
RecordUserId	String	当前任务用于实时录制服务进房的用户 ID

示例

```
{
    "EventData": {
        "RecordUserId": "tic_record_user_1253_01",
        "RoomId": 1253,
        "TaskId": "bj0mt2l23osdj300hl30"
    },
    "EventType": "OnlineRecordStarted",
    "ExpireTime": 1590046391,
    "SdkAppId": 1400000001,
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1575545412
}
```

录制停止

事件名称

OnlineRecordStopped

事件说明

当用户主动调用停止录制接口或者由于其他原因自动停止录制时通知 App 后台。

EventData 结构定义

参数名称	类型	描述
TaskId	String	录制已停止的实时录制 TaskId
RoomId	Integer	实时录制的房间 ID
RecordUserId	String	当前任务用于实时录制服务进房的用户 ID

示例

```
{
    "EventData": {
        "RecordUserId": "tic_record_user_1253_01",
        "RoomId": 1253,
```

```
        "TaskId": "bj0mt2l23osdj300hl30"
    },
    "EventType": "OnlineRecordStopped",
    "ExpireTime": 1590046391,
    "SdkAppId": 1400000001,
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1575545412
}
```

录制结果回调

事件名称

OnlineRecordFinished

事件说明

当实时录制完成时通知 App 后台。

EventData 结构定义

参数名称	类型	描述
TaskId	String	录制结束的实时录制任务 ID
RoomId	Integer	房间号
GroupId	String	白板的群组 ID
RecordUserId	String	录制用户 ID
FinishReason	String	录制结束原因 • AUTO：房间内长时间没有音视频上行及白板操作导致自动停止录制 • USER_CALL：主动调用了停止录制接口 • EXCEPTION：录制异常结束
RecordStartTime	Integer	实际开始录制时间，Unix 时间戳，单位秒
RecordStopTime	Integer	实际停止录制时间，Unix 时间戳，单位秒
TotalTime	Integer	回放视频总时长（单位：毫秒）
ExceptionCnt	Integer	录制过程中出现异常的次数
OmittedDurations	Array of OmittedDuration	拼接视频中被忽略的时间段，只有开启视频拼接功能的时候，这个参数才是有效的
VideoInfos	Array of VideoInfo	录制视频列表
Error.Code	String	如果实时录制发生错误会有该字段，具体错误码描述请参考接口文档 查询实时录制任务
Error.Message	String	如果实时录制发生错误的具体错误描述

示例

- **示例1 实时录制成功的回调**

```
{
  "EventData": {
    "TaskId": "ghucnligqtgtvk2624mb",
    "RoomId": 880528,
    "GroupId": "880528",
```

```
    "RecordUserId": "tic_record_user_880528_test-01",
    "FinishReason": "USER_CALL",
    "RecordStartTime": 1568949369,
    "RecordStopTime": 1568949392,
    "TotalTime": 18317,
    "ExceptionCnt": 0,
    "OmittedDurations": [],
    "VideoInfos": [
      {
        "UserId": "Mac_trtc_04",
        "VideoDuration": 17969,
        "VideoFormat": "mp4",
        "VideoId": "dace3518e865e76a9e36712c629822ba",
        "VideoPlayTime": 0,
        "VideoSize": 593418,
        "VideoType": 0,
        "VideoUrl": "http://online-recording-
1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/d124f518e865e76a9e36712c629822ba.mp4"
      },
      {
        "UserId": "tic_mixstream_880528_101",
        "VideoDuration": 18205,
        "VideoFormat": "mp4",
        "VideoId": "763d1f6b8679c3f17fb118bd37d05c85",
        "VideoPlayTime": 3,
        "VideoSize": 765545,
        "VideoType": 3,
        "VideoUrl": "http://online-recording-
1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/763d1f6b86724f51fb118bd37d05c85.mp4"
      },
      {
        "UserId": "tic_mixstream_880528_3",
        "VideoDuration": 18222,
        "VideoFormat": "mp4",
        "VideoId": "1b9623df0516dc7318df89f6e7fffc1e",
        "VideoPlayTime": 95,
        "VideoSize": 402038,
        "VideoType": 3,
        "VideoUrl": "http://online-recording-
1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/1b9623df05124f51318df89f6e7fffc1e.mp4"
      },
      {
        "UserId": "",
        "VideoDuration": 17605,
        "VideoFormat": "mp4",
        "VideoId": "a8152f8faa2cfe621dc965a066a5813c",
        "VideoPlayTime": 623,
        "VideoSize": 226337,
        "VideoType": 2,
        "VideoUrl": "http://online-recording-
1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/a815224f512cfe621dc965a066a5813c.mp4"
      }
    ]
  },
  "EventType": "OnlineRecordFinished",
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
```

```
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1575545412
}
```

- **示例2 实时录制失败的回调**

```
{
    "EventData": {
        "Error": {
            "Code": "FailedOperation.Record",
            "Message": "code: 40001, msg: 2020-05-21 15:20:06, invalid userid\n"
        },
        "ExceptionCnt": 0,
        "FinishReason": "EXCEPTION",
        "GroupId": "880528",
        "OmittedDurations": [],
        "RecordStartTime": 0,
        "RecordStopTime": 1590045606,
        "RecordUserId": "tic_record_user_880528_test-01",
        "RoomId": 880528,
        "TaskId": "ghucnliggtgtvk2624mb",
        "TotalTime": 0,
        "VideoInfos": []
    },
    "EventType": "OnlineRecordFinished",
    "ExpireTime": 1590046391,
    "SdkAppId": 1400000001,
    "Sign": "a2dabb362a9b811c0e26953a6276a41c",
    "Timestamp": 1575545412
}
```

长时间暂停告警

事件名称

OnlineRecordLongPauseWarning

事件说明

实时录制暂停超过30分钟会自动停止，当暂停了20分钟时，会通知 App 后台处理该任务。

EventData 结构定义

参数名称	类型	描述
TaskId	String	录制已开始的实时录制 TaskId
RoomId	Integer	实时录制的房间 ID
RecordUserId	String	当前任务用于实时录制服务进房的用户 ID
PauseTime	Integer	暂停时间，Unix 时间戳，单位秒
EstimateStopTime	Integer	预计自动停止时间，Unix 时间戳，单位秒

示例

```
{
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "RoomId": 1253,
```

```
    "RecordUserId": "tic_record_user_1253_01",
    "PauseTime": 1575545412,
    "EstimateStopTime": 1575546412
  },
  "EventType": "OnlineRecordLongPauseWarning",
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412
}
```


长时间暂停自动结束

事件名称

OnlineRecordLongPauseForceStopped

事件说明

实时录制暂停超过30分钟会自动停止，会通知 App 后台已经自动停止当前任务。

EventData 结构定义

参数名称	类型	描述
TaskId	String	录制已开始的实时录制 TaskId
RoomId	Integer	实时录制的房间 ID
RecordUserId	String	当前任务用于实时录制服务进房的用户 ID
PauseTime	Integer	暂停时间，Unix 时间戳，单位秒
StopTime	Integer	自动停止时间，Unix 时间戳，单位秒

示例


```
{
  "EventData": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "RoomId": 1253,
    "RecordUserId": "tic_record_user_1253_01",
    "PauseTime": 1575545412,
    "StopTime": 1575546412
  },
  "EventType": "OnlineRecordLongPauseForceStopped",
  "ExpireTime": 1590046391,
  "SdkAppId": 1400000001,
  "Sign": "a2dabb362a9b811c0e26953a6276a41c",
  "Timestamp": 1575545412
}
```

任务告警事件

Last updated：2024-09-11 09:08:12

概述

告警事件回调是通过客户在控制台或调用设置 API 设置的回调地址进行事件回调。其目的是将当前正在运行的存量任务的异常情况以告警事件的方式通知客户，以方便客户感知异常并做出相应的决策。

告警事件说明

任务排队数量超过指定阈值告警事件

事件类型

TaskQueuedAmountLimitWarning

事件说明

当客户发起的任务没有被及时执行处于排队状态，且排队任务数超过指定告警阈值时，发送当前事件通知客户后台。

EventData 数据结构定义

参数名称	类型	描述
TaskType	String	任务类型，目前已有类型如下： • TranscodeH5：动态转码任务，文档转 HTML5 页面 • TranscodeJPG：静态转码任务，文档转图片 • WhiteboardPush：白板推流任务 • OnlineRecord：实时录制任务
Total	Integer	排队任务总数
Tasks	Object	排队任务列表
Tasks.TaskID	String	任务 ID
Tasks.CreateTime	String	任务创建时间
Tasks.CancelTime	String	任务取消时间
Tasks.RoomID	Integer	任务所在房间号，转码任务没有房间号
Tasks.FileURL	String	转码任务指定的转码文件原始URL
Task.SdkAppID	Integer	应用 SDKAppID

示例

以白板推流为例：

```
{
  "EventType": "TaskQueuedAmountLimitWarning",
  "EventData": {
    "TaskType": "WhiteboardPush",
    "Total": 1,
    "Tasks": [
      {
          "TaskID": "bj0mt2l23osdj300hl31",
          "CreateTime": "2023-03-14 15:00:00",
          "CancelTime": "",
```

```
            "RoomID": 12345,
            "FileURL": "",
            "SdkAppID": 1400000001
        }
    ]
  }
}
```

任务排队等待时长超过指定阈值告警事件

事件类型

TaskQueuedTimeLimitWarning

事件说明

当客户发起的任务没有被及时执行处于排队状态，且排队任务等待时间超过指定告警阈值时，发送当前事件通知客户后台。

EventData 数据结构定义

参数名称	类型	描述
TaskType	String	任务类型，目前已有类型如下： • TranscodeH5：动态转码任务，文档转 HTML5 页面 • TranscodeJPG：静态转码任务，文档转图片 • WhiteboardPush：白板推流任务 • OnlineRecord：实时录制任务
Total	Integer	排队任务总数
Tasks	Object	排队任务列表
Tasks.TaskID	String	任务 ID
Tasks.CreateTime	String	任务创建时间
Tasks.CancelTime	String	任务取消时间
Tasks.RoomID	Integer	任务所在房间号，转码任务没有房间号
Tasks.FileURL	String	转码任务指定的转码文件原始 URL
Task.SdkAppID	Integer	应用 SDKAppID

示例

以白板推流为例：

```
{
  "EventType": "TaskQueuedTimeLimitWarning",
  "EventData": {
    "TaskType": "WhiteboardPush",
    "Total": 1,
    "Tasks": [
      {
```

```
            "TaskID": "bj0mt2l23osdj300hl31",
            "CreateTime": "2023-03-14 15:00:00",
            "CancelTime": "",
            "RoomID": 12345,
            "FileURL": "",
            "SdkAppID": 1400000001
        }
    ]
  }
}
```

任务并发数量占比超过指定阈值告警事件

事件类型

TaskConcurrencyLimitWarning

事件说明

当客户发起了大量任务，且当前正在执行的任务数占允许的最大并发数的比例超过指定阈值时，发送当前事件通知客户后台。

EventData 数据结构定义

参数名称	类型	描述
TaskType	String	任务类型，目前已有类型如下： • TranscodeH5：动态转码任务，文档转 HTML5 页面 • TranscodeJPG：静态转码任务，文档转图片 • WhiteboardPush：白板推流任务 • OnlineRecord：实时录制任务
NowRunning	Integer	当前已创建且正在执行的任务数
MaxAllowed	Integer	当前任务类型允许的最大并发数

示例

以白板推流为例：

```
{
  "EventType": "TaskConcurrentWarning",
  "EventData": {
    "TaskType": "WhiteboardPush",
    "NowRunning": 1,
    "MaxAllowed": 300
  }
}
```

任务异常告警事件

事件类型

TaskExceptionWarning

事件说明

当任务出现异常的时候，发送当前事件通知客户后台。

EventData 数据结构定义

参数名称	类型	描述

TaskType	String	任务类型，目前已有类型如下： • TranscodeH5：动态转码任务，文档转 HTML5 页面 • TranscodeJPG：静态转码任务，文档转图片 • WhiteboardPush：白板推流任务 • OnlineRecord：实时录制任务
TaskInfo	Object	异常任务信息
TaskInfo.TaskID	String	任务 ID
TaskInfo.CreateTime	String	任务创建时间
TaskInfo.CancelTime	String	任务取消时间
TaskInfo.RoomID	Integer	任务所在房间号，转码任务没有房间号
TaskInfo.FileURL	String	转码任务指定的转码文件原始 URL
TaskInfo.SdkAppID	Integer	应用 SDKAppID
ExceptionInfos	Object	异常信息列表
ExceptionInfos.Type	String	异常类型，目前已有类型如下： • BoardError：白板发生异常 • IMError： IM 发生异常 • TRTCError：TRTC 发生异常
ExceptionInfos.Time	String	异常时间点
ExceptionInfos.Message	String	异常具体信息

示例

以白板推流为例：

```
{
  "TaskType": "WhiteboardPush",
  "TaskInfo": {
    "CancelTime": "",
    "CreateTime": "2023-08-16 14:58:34",
    "FileURL": "",
    "RoomID": 67042,
    "SdkAppID": 1400000001,
    "TaskID": "08g8hoso9bcobi677ejc"
  },
  "ExceptionInfos": [
    {
        "Type": "BoardError",
        "Time": "2023-08-16 15:42:59",
```

```
        "Message": "OnTEBImageStatusChanged, load image failed after refresh, boardID
web_auto_test_user971_1692169788_3_#1692169788643, url https://example.com/3.jpg, status 4"
    }
  ]
}
```