实时互动-教育版

高级功能

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

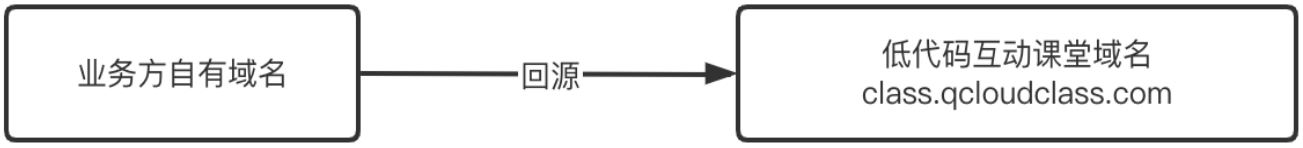
高级功能
自定义业务域名
移动端（平板）屏幕共享
移动端预加载

高级功能
自定义业务域名

最近更新时间：2023-10-24 14:43:44

想使用自有域名完成实时互动-教育版 Web 端的访达：
从地址 `https://yourdomain.com/` **转为** `https://class.qcloudclass.com/`

方案

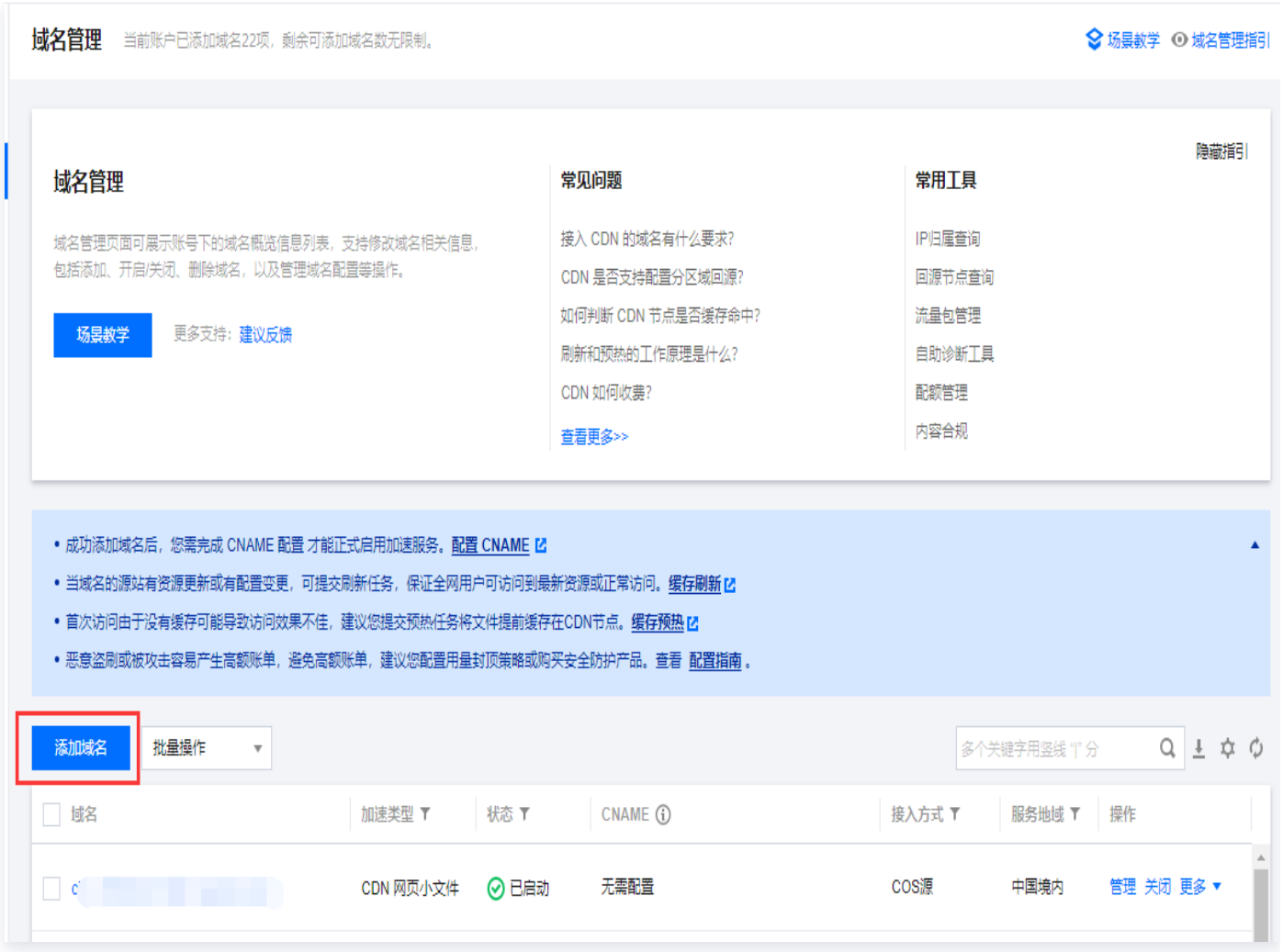


操作步骤

> **说明：**
> 以 腾讯云 CDN（内容分发网络）为例。
> 我们假设业务层拥有一个课堂专属三级域名： `class.yourdomain.com`

一. 配置 CDN

1. 前往 内容分发网络 控制台，单击**添加域名**进行添加。

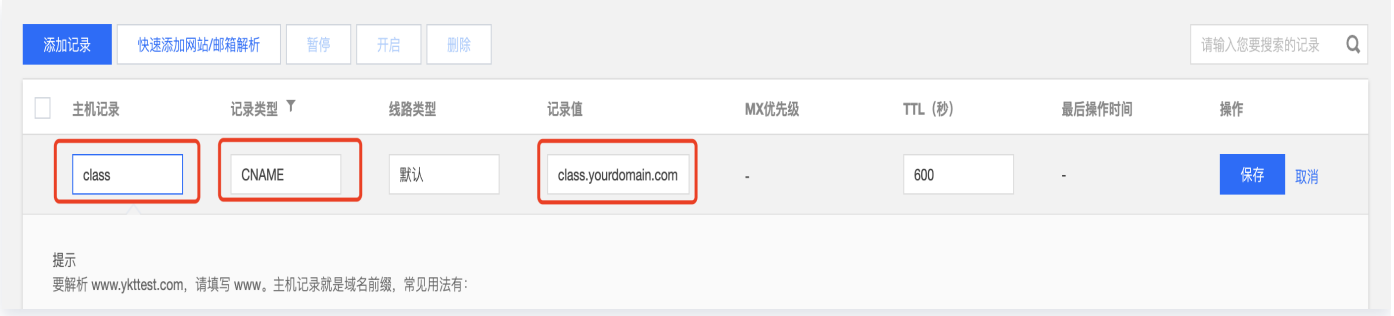


2. 配置回源参数。填写相关域名配置和源站配置。

3. 添加成功后，单击域名查看基础配置信息，复制 cname。

二. 配置域名解析

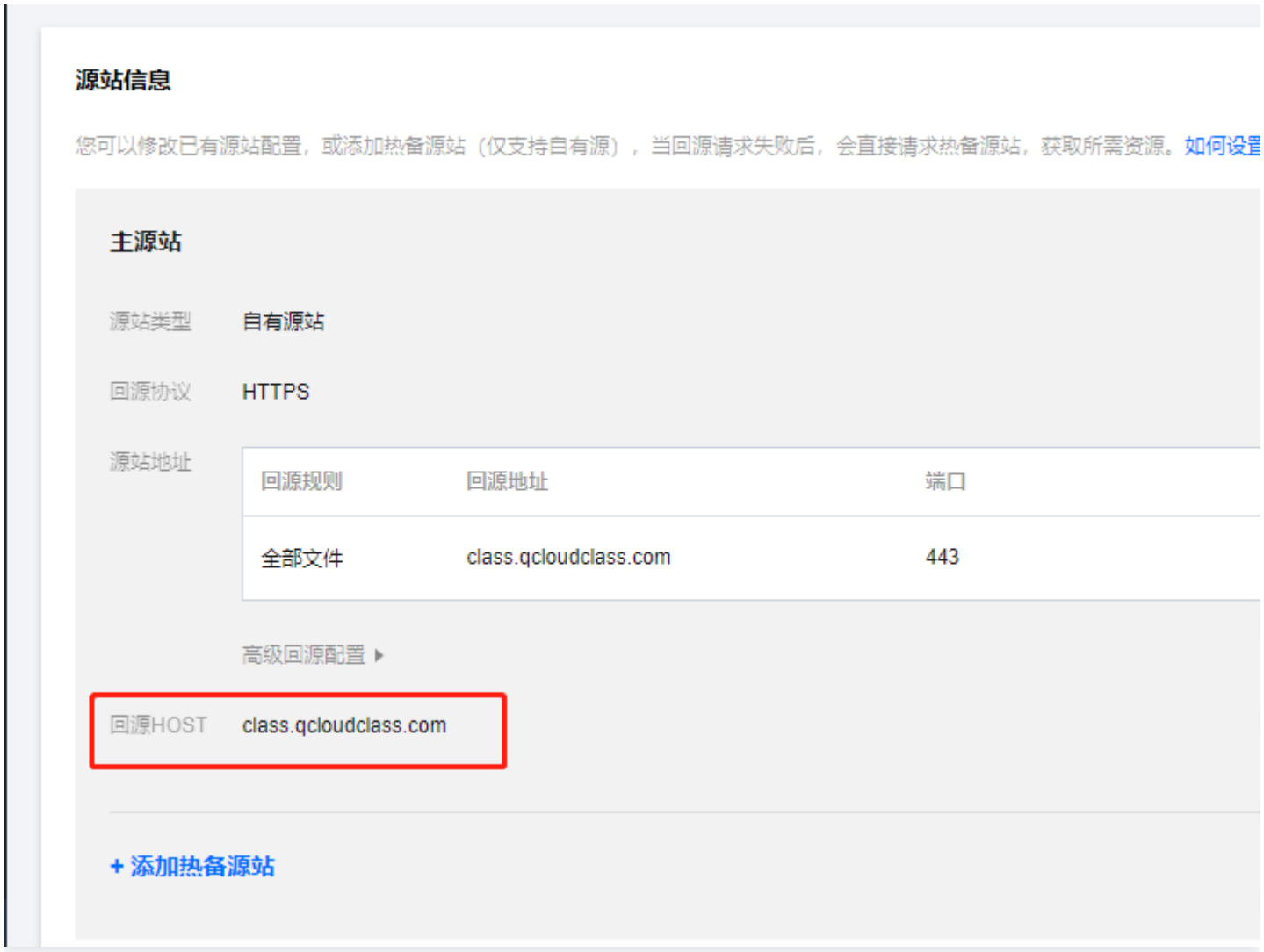
1. 前往域名解析配置 class.yourdomian.com 的域名指向。
2. 配置域名解析指向，以 云解析 DNS 为例。类型选择 CNAME，记录值为前面步骤获取的 CNAME 值。



3. 单击**保存**。

三. 配置回源 HOST

进入**回源配置**页面，修改 `回源HOST` => `class.qcloudclass.com` 。



优化配置

为了让自定义业务域名能达到更好的访问效果，还需要进行以下调整。

一. 完成 https 配置

在 域名管理 页面，单击域名后选择 HTTPS配置去进行配置。

二. 完成节点缓存过期配置

在**缓存配置**页面进行节点缓存过期配置。

三. 完成 gzip 配置

在**高级配置**页面，配置智能压缩规则。

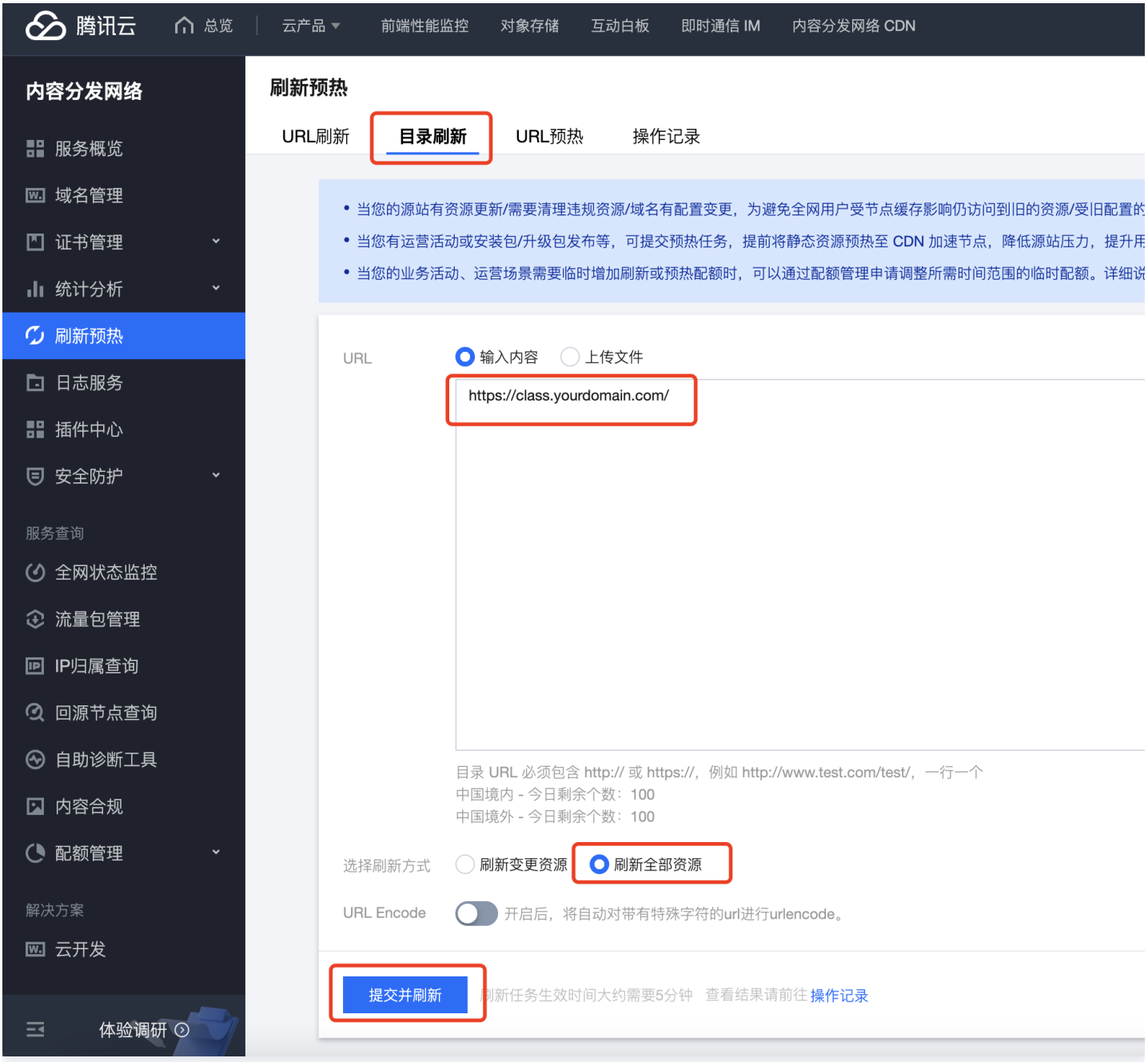
四. 缓存键规则配置

在**缓存配置**页面，配置缓存键规则。

资源刷新

请参见 CDN 刷新预热 。

> **为何需要做资源刷新?**
> 根据实际的运营需求/Bug 修复/体验优化，实时互动-教育版会对线上版本不定期进行热更新。
> CDN 的缓存策略决定可能出现业务自有域名与实时互动-教育版官方域名表现不一致的情况。
> 我们建议业务侧定期进行资源的刷新操作，或按需进行立即刷新。



移动端（平板）屏幕共享

最近更新时间：2023-10-07 16:44:02

iOS 端

1. 创建 `App Group`，参见实时音视频文档中 步骤 1：创建 App Group。
2. 创建 `Broadcast Upload Extension`，参见实时音视频文档中 步骤 2：创建 Broadcast Upload Extension。
3. 为新创建的 `Target`，依赖 `TCICSDK_ReplayKit`，如下代码，之后重新 `pod install` 即可。

```
target '新target名' do
# Comment the next line if you don't want to use dynamic frameworks
# use_frameworks!
pod 'TCICSDK_ReplayKit'
end
```

4. 添加下列代码复制到 `SampleHandler.m` 中，将 `APPGROUP` 改为第 1 步创建的 `App Group`。

```
#import "SampleHandler.h"
#import <TXLiteAVSDK_ReplayKitExt/TXLiteAVSDK_ReplayKitExt.h>
#import <TCICScreenKit/TCICScreenKit.h>
// 注意：此处的 APPGROUP 需要改成上文中的创建的 App Group Identifier。
#define APPGROUP ""

@interface SampleHandler() <TXReplayKitExtDelegate>
@end
@implementation SampleHandler

- (void)broadcastStartedWithSetupInfo:(NSDictionary<NSString
*,NSObject *> *)setupInfo {
    [[TXReplayKitExt sharedInstance] setupWithAppGroup:APPGROUP
delegate:self];

    [[TCICScreenKit sharedScreenKit] onScreenKitStarted];
}
- (void)broadcastPaused {
    // User has requested to pause the broadcast. Samples will stop
being delivered.
    [[TCICScreenKit sharedScreenKit] onScreenKitPaused];
}
```

```
- (void)broadcastResumed {
    // User has requested to resume the broadcast. Samples delivery
will resume.
    [[TCICScreenKit sharedScreenKit] onScreenKitResumed];
}
- (void)broadcastFinished {
    [[TXReplayKitExt sharedInstance] finishBroadcast];
    // User has requested to finish the broadcast.
    [[TCICScreenKit sharedScreenKit] onScreenKitFinished];
}
#pragma mark - TXReplayKitExtDelegate
- (void)broadcastFinished:(TXReplayKitExt *)broadcast reason:
(TXReplayKitExtReason)reason
{
    NSString *tip = @"";
    switch (reason) {
        case TXReplayKitExtReasonRequestedByMain:
            tip = @"屏幕共享已结束";
            break;
        case TXReplayKitExtReasonDisconnected:
            tip = @"应用断开";
            break;
        case TXReplayKitExtReasonVersionMismatch:
            tip = @"集成错误（SDK 版本号不相符合）";
            break;
    }
    NSError *error = [NSError
errorWithDomain:NSStringFromClass(self.class) code:0 userInfo:@{
        NSLocalizedFailureReasonErrorKey:tip
    }];
    [self finishBroadcastWithError:error];
}
- (void)processSampleBuffer:(CMSampleBufferRef)sampleBuffer withType:
   (RPSampleBufferType)sampleBufferType {
    switch (sampleBufferType) {
        case RPSampleBufferTypeVideo:
            [[TXReplayKitExt sharedInstance]
sendVideoSampleBuffer:sampleBuffer];
            break;
        case RPSampleBufferTypeAudioApp:
            // Handle audio sample buffer for app audio
            break;
        case RPSampleBufferTypeAudioMic:
```

```
            // Handle audio sample buffer for mic audio
            break;

        default:
            break;
    }
}
@end
```

5. 对接主 App 端的接收逻辑：目前主 App 中的使用 TCICSDK，已支持系统屏幕分享相关逻辑，只需要业务方配置好 App Group ，在进入课堂前，设置 AppGroup 即可。

```
TCICClassConfig *roomConfig = [[TCICClassConfig alloc] init];
roomConfig.userId = "test";
roomConfig.token = "test_token";
roomConfig.classId = 123454;
roomConfig.schoolId = xxxxx;

// 通过KVC方式设置AppGroup
[roomConfig setValue:@"group.com.xx.xxxx" forKey:@"appGroup"];
```

注意事项

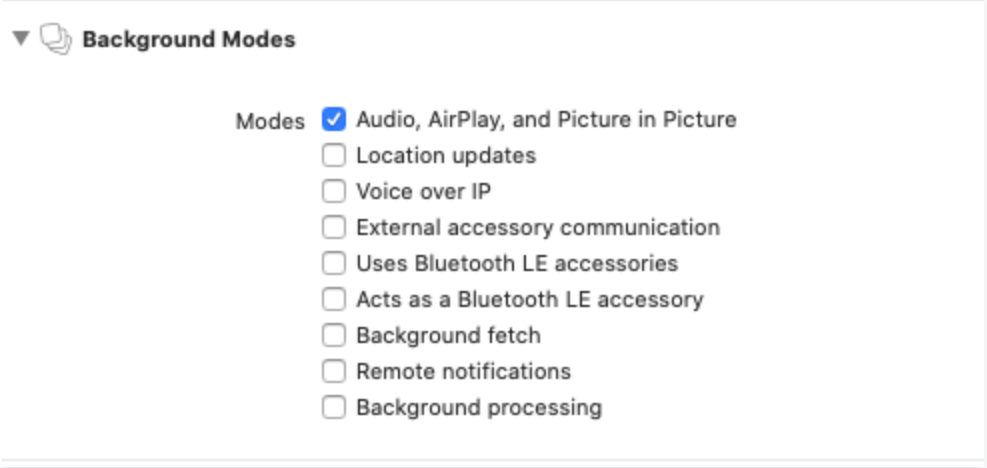
1. TCICSDK 中已支持屏幕分享的触发按钮，具体可参见实时音视频文档中 步骤 4：增加屏幕分享的触发按钮（可选），但该功能有限制条件：
   1.1 屏幕分享的触发按钮 只支持 iOS12 以上，同时需要创建的工程，不依赖 `Scene` 生命周期 ，如果代码中已支持 `Scenedelegate` ，可参见 Xcode 11 删除 Scenedelegate ，进行移除。以 Demo 为例，弹

出效果如下，单击开始直播即可。

1.2 对于 iOS11 的机型，需要业务侧引导用户从远程控制中 `长按录屏` 进行触发，并选择业务自创建的 `Broadcast Upload Extension` 进行触发，下图以 `腾讯会议` 为例：

长按“录屏” 按钮　　点击“开始直播”

2. 创建的 `Upload Extension` 的 `Deployment target` 配置在 `iOS 11.0` （ `Replay Kit` 于 `iOS11` 才开始支持），调试时真机也尽量在 `iOS11` 之后。
3. 主 App 要支持系统级屏幕分享，需要添加 `Background Modes` 。



Android 端

> **说明：**
> Android的屏幕分享对版本的要求： Android 5.0 +

除版本要求外，没有其他配置及特殊要求。

版权所有：腾讯云计算（北京）有限责任公司

移动端预加载

最近更新时间：2024-09-06 15:52:51

移动端进入课堂时，需要加载课中资源，在一般性能一般的机器上，可能会出现短暂黑屏过渡页面，影响体验。为此，我们提供了提前加载课中资源的能力，以此达到优化进课堂体验的目标。建议在进入应用时调用，提前加载课中资源。

> **说明：**
> 预加载只需首次调用，后续退出课堂无须再调用，SDK 内部会自动预加载。

Android 预加载接口

预加载课中资源 preloadClass

参数说明

参数	类型	描述
context	Context	上下文
env	String	环境（填空即可）

示例

```
TCICWebViewManager.getInstance().preloadClass(context, '');
```

iOS 预加载接口

参数说明

参数	类型	描述
env	String	环境（填空即可）

示例

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    // 开启预加载逻辑
    [TCICClassController preloadClass:nil];
```

```
    return YES;
}
```