

低代码互动课堂

客户端集成指引

产品文档



腾讯云

【版权声明】

©2013-2024 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默示的承诺或保证。

文档目录

客户端集成指引

Web 和 H5

Android

iOS

Windows 和 macOS

客户端集成指引

Web 和 H5

最近更新时间：2024-06-28 10:05:05

本文主要介绍如何快速将腾讯云实时互动-教育版 LCIC Web/H5 应用快速集成到您的项目中。

前提条件

1. 您已经 [注册腾讯云](#) 账号，并完成 [实名认证](#)。
2. 您使用的桌面或手机浏览器能够支持音视频服务。详细要求可参考 [Web 端常见问题](#)。

步骤一：创建新的应用

1. 登录 [实时互动-教育版控制台](#)，在左侧导航栏选择 **快速跑通应用**。
2. 默认进入“创建应用”界面，应用类型可选择“**创建新应用**”，输入应用名称，例如 TestLCIC。
若您已创建应用，应用类型项可单击“选择已有应用”。

说明：

每个账号可免费领用一个试用版应用，若需创建商用版应用，可根据业务需求在 [购买页](#) 选择并购买对应版本。

3. 添加或编辑标签（可选），单击 **创建**。

快速跑通应用 体验零代码快速开发

1 创建应用 > 2 生成课堂参数 > 3 跑通应用成功

创建/选择应用

应用类型 创建新应用 选择已有应用

应用名称 0/15

添加标签 (可选填)

标签用于从不同维度对资源分类管理。如现有标签不符合您的要求，请前往 [标签管理](#)

标签更新后，请 [点击此处刷新标签](#)

标签

+添加标签

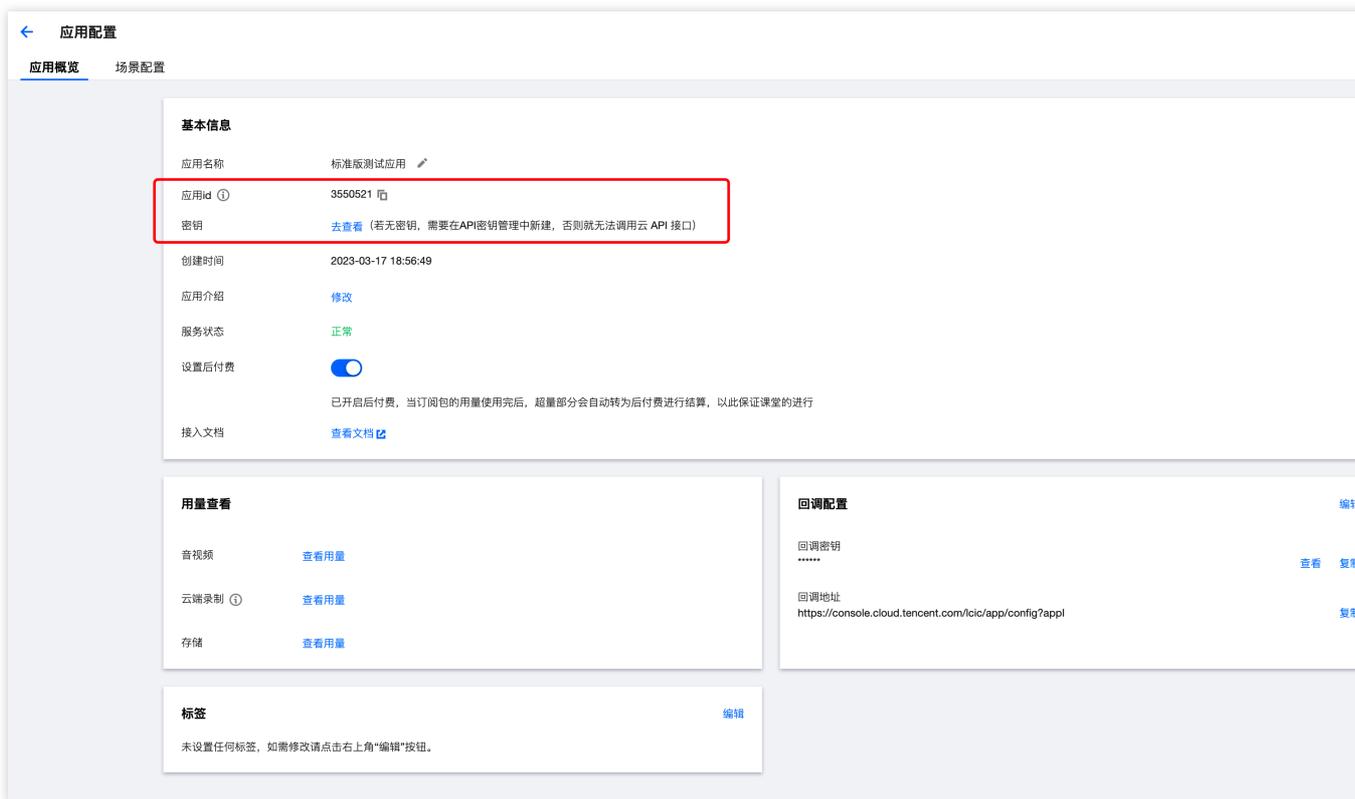
说明：

应用名称只允许使用下划线、数字或中英文字符。

标签用于标识和组织您在腾讯云的各种资源。例如：企业可能有多个业务部门，每个部门有1个或多个 LCIC 应用，这时，企业可以通过给 LCIC 应用添加标签来标记部门信息。标签并非必选项，您可根据实际业务需求添加或编辑。

步骤二：获取 SDKAppId 和密钥(SecretKey)

1. 进入 [应用管理](#) > [应用配置](#)，获取 SDKAppId。
2. 进入 [访问管理\(CAM\)控制台](#) 获取密钥，若无密钥，需要在 API 密钥管理中进行新建，具体可参见 [密钥管理](#)。



步骤三：获取进入课堂所需参数

1. 通过调用云 API 接口 [RegisterUser](#) 注册用户，可以获取到对应的用户 ID(`userid`)信息。
2. 通过云 API 接口 [LoginUser](#) 登录，可以获取到用户鉴权 `token` 信息。
3. 通过云 API 接口 [CreateRoom](#) 创建课堂，可以获取到课堂号(`classid`)信息。
4. 明确需要集成的课堂版本：`latest`。
5. 其中 `scene`、`debugjs`、`debugcss` 为非必填参数，在需要自定义 UI 时才需设置，具体可参见 [自定义 UI 集成](#)。其中 `debugjs` 和 `debugcss` 只用于自定义布局、组件时的调试，且只支持通过 `localhost` 或 `127.0.0.1` 的地址进行访问，在发布阶段请勿使用此参数。
6. `lng`、`location`、`layout` 也是非必填参数，业务侧可自行判断是否需要传入，不传则使用默认值，其中 `layout` 参数只有在教室布局为视频+文档布局(`videodoc`)时才生效。

字段	类型	含义	备注	必填	
<code>userid</code>	<code>string</code>	用户名	通过 RegisterUser 接口获取	是	
<code>classid</code>	<code>string</code>	课堂 ID	通过 CreateRoom 接口创建返回获取	是	
<code>token</code>	<code>string</code>	后台鉴权参数	通过 LoginUser 接口获取	是	
<code>version</code>	<code>string</code>	课堂版本号	通过发布日志选择对应版本。说明： 互动课	是	

			堂客户端版本，建议使用 latest		
scene	string	场景名称	用于区分不同的定制布局，通过 SetAppCustomContent 接口配置，默认为 <code>default</code>	否	
debugjs	string	自定义 UI 的 JS 链接	通过自定义 UI 集成方式获取	否	
debugcss	string	自定义 UI 的 CSS 链接	通过自定义 UI 集成方式获取	否	
role	string	进入课堂角色，默认空	可选参数 supervisor (巡课/内容审查)，只有已注册应用内巡课用户才有权限	否	
lng	string	语言参数，默认 zh-CN	当前支持 zh-CN、en-US	否	
location	boolean	是否上报经纬度位置信息	默认 false 不上报	否	
layout	string	页面布局	默认顶部布局(top)，当前仅视频文档模式有效，支持双排布局(double)、右侧布局(right)、左侧布局(left)、三分布局(three)	否	
boardColor	string	白板颜色	白板颜色设置，默认为： <code>#182E25</code> ，支持 Hex 格式，也支持 <code>rgba(0, 0, 0, .3)</code> 设置	否	
back_url	string	退出课堂回调地址	默认为空，单击返回或退出课堂时需要回跳的页面地址。参数需要使用 <code>encodeURIComponent</code> 编码	否	

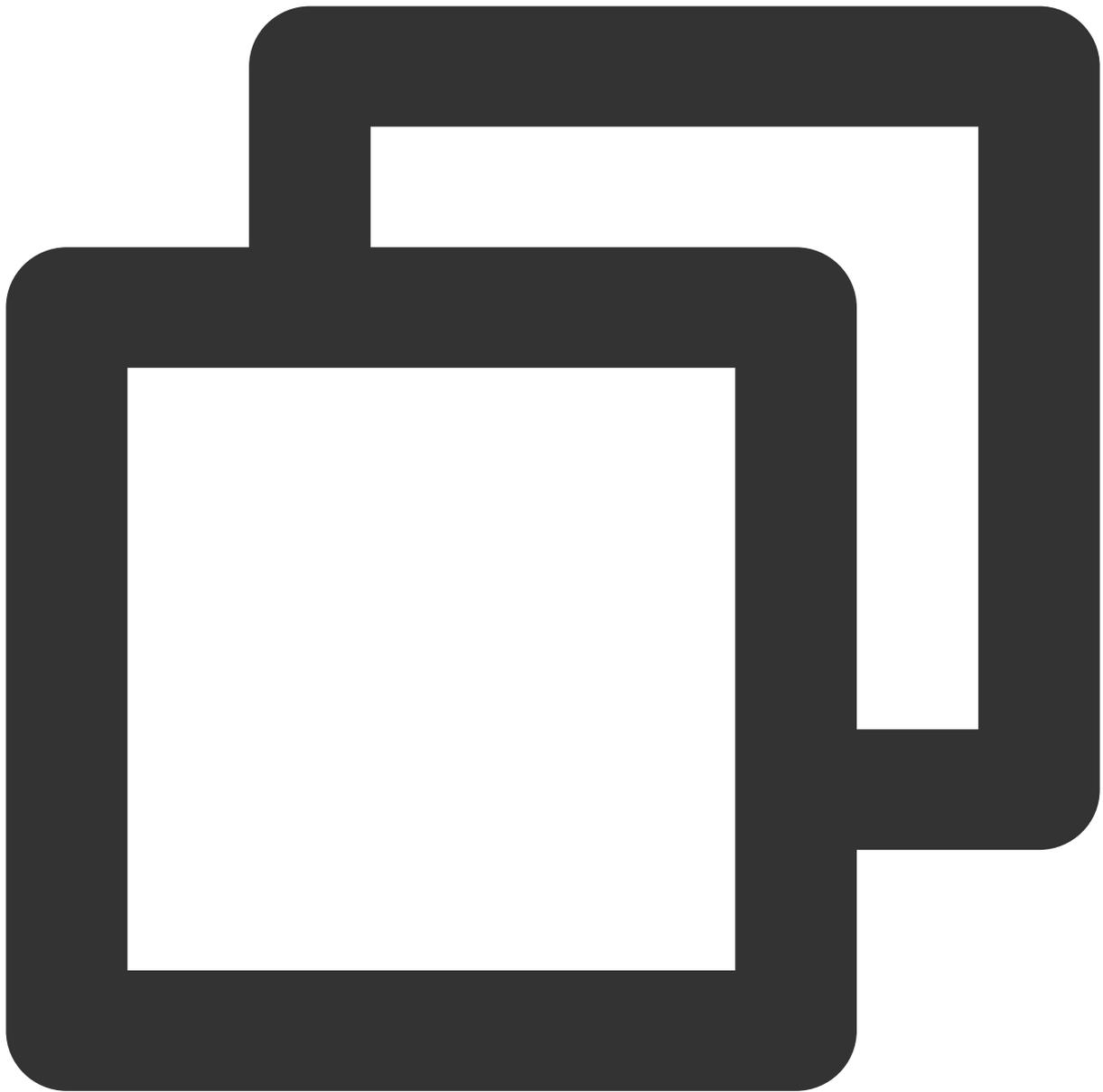
步骤四：进入课堂

标准功能

将刚才获取到的参数，按以下规范拼接出用户进入课堂的 **URL**，通过跳转此 URL 链接即可实现 Web 端的集成。

[https://class.qcloudclass.com/latest/class.html?userid=\\${userid}&token=\\${token}&classid=\\${classid}](https://class.qcloudclass.com/latest/class.html?userid=${userid}&token=${token}&classid=${classid})

具体链接如下所示：



```
https://class.qcloudclass.com/latest/class.html?userid=12345&token=yJhbGciOiJIUzI1N
```

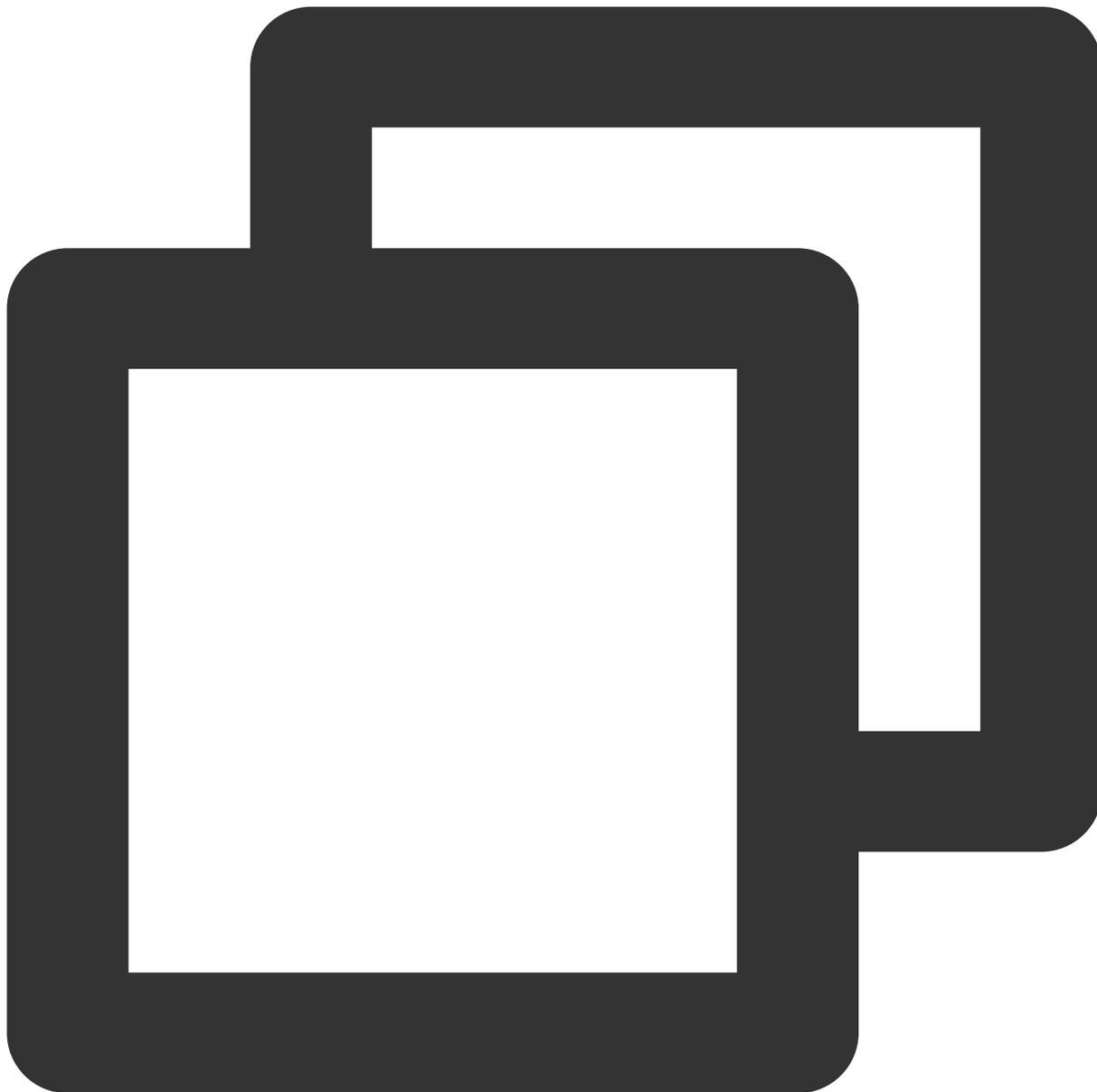
说明：

在拼接的 URL 中 `userid` 与 [创建课堂](#) 指定的老师 ID (`teacherid`) 是一致的，则当前用户为老师。若与当前课堂的助教 ID (`assistantid`) 一致，则为助教，否则为学生。

高级功能

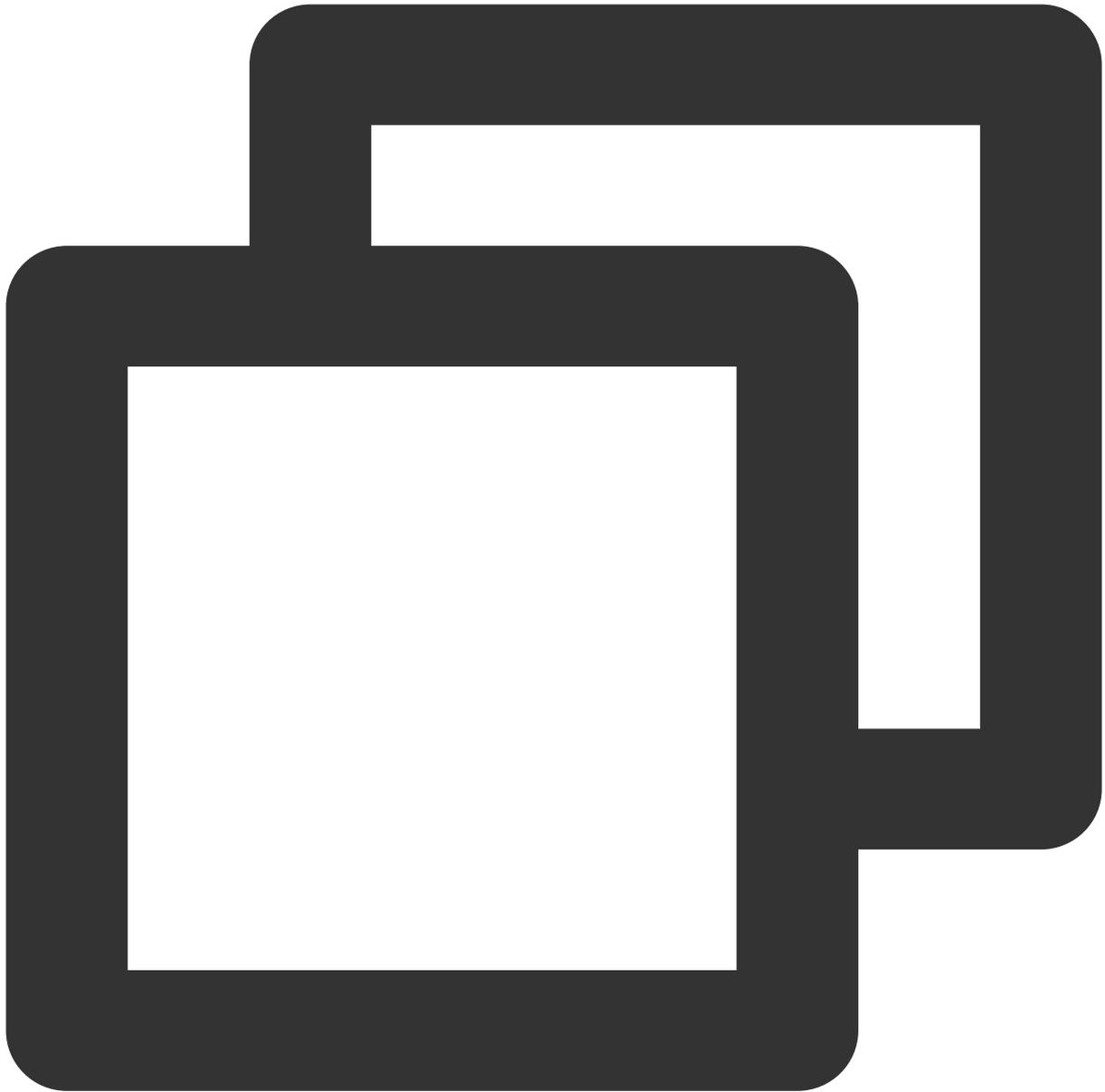
自定义 UI 集成

实时互动-教育版 LCIC Web/H5 目前还提供了自定义 UI 的集成方案。用户可自定义业务侧课中的布局及样式，通过 [界面定制](#) 可以获取到业务侧的 JS 及 CSS 链接，将 `debugjs` 及 `debugcss` 参数拼接到上方的链接上即可（此参数只用于调试），如下所示：



```
https://class.qcloudclass.com/latest/class.html?userid=12345&token=yJhbGciOiJIUzI1N
```

当自定义 JS 与 CSS 调试完成后，可通过云 API 接口 [SetAppCustomContent](#) 或 [控制台 > 应用配置 > 场景配置](#) 将场景与自定义的 JS、CSS 链接进行绑定，在进入课堂时将 `scene` 参数拼接到 URL 上，即可加载对应场景的布局及组件。在涉及多种班型、多种布局时，业务侧可根据此参数实现场景的切换。如下所示：



```
http://class.qcloudclass.com/latest/class.html?userid=12345&token=yJhbGciOiJIUzI1Ni
```

自定义业务域名

在课中页面时，若业务侧想隐藏课堂域名，只显示业务域名，可通过 [内容分发网络\(CDN\)控制台](#) 新建业务域名，并回源到课堂域名即可，详细流程请参见 [自定义业务域名](#)。

其他相关文档

[LCIC API](#)

[界面定制](#)

[Web 端相关问题](#)

Android

最近更新时间：2024-08-16 14:45:00

开发环境要求

Android studio 3.0+

Android 4.4（API 19）及以上系统

前提条件

您已 [注册腾讯云](#) 账号，并完成 [实名认证](#)。

步骤一：创建新的应用

1. 登录 [实时互动-教育版控制台](#)，在左侧导航栏选择**快速跑通应用**。
2. 默认进入“创建应用”界面，应用类型可选择“**创建新应用**”，输入应用名称，例如 TestLCIC。
若您已创建应用，应用类型项可单击“选择已有应用”。

说明：

移动端需要购买旗舰版或企业尊享版后，方可接入移动端。若需创建商用应用，可根据业务需求在 [购买页](#) 创建对应版本的应用。

3. 根据实际业务需求添加或编辑标签，单击**创建**。

快速跑通应用 体验零代码快速开发

1 创建应用 > 2 生成课堂参数 > 3 跑通应用成功

创建/选择应用

应用类型 创建新应用 选择已有应用

应用名称 0/15

添加标签 (可选填)

标签用于从不同维度对资源分类管理。如现有标签不符合您的要求，请前往 [标签管理](#)
标签更新后，请 [点击此处刷新标签](#)

标签

+添加标签

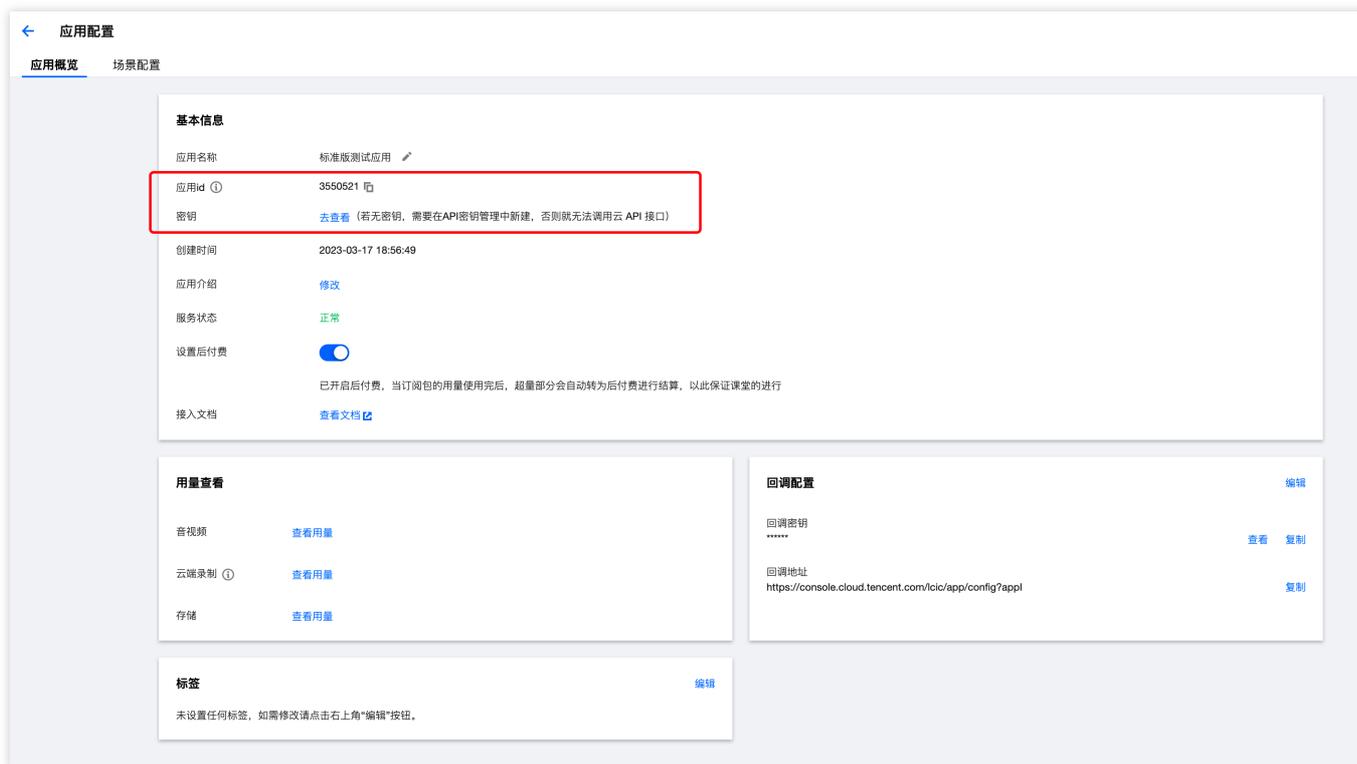
说明：

应用名称只允许下划线、数字或中英文字符。

标签用于标识和组织您在腾讯云的各种资源。例如：企业可能有多个业务部门，每个部门有1个或多个 LCIC 应用，这时，企业可以通过给 LCIC 应用添加标签来标记部门信息。标签并非必选项，您可根据实际业务需求添加或编辑。

步骤二：获取 SDKAppId 和密钥(SecretKey)

1. 进入 [应用管理](#) > [应用配置](#)，获取 SDKAppId。
2. 进入 [访问管理\(CAM\)控制台](#) 获取密钥，若无密钥，需要在 API 密钥管理中进行新建，具体可参见 [密钥管理](#)。

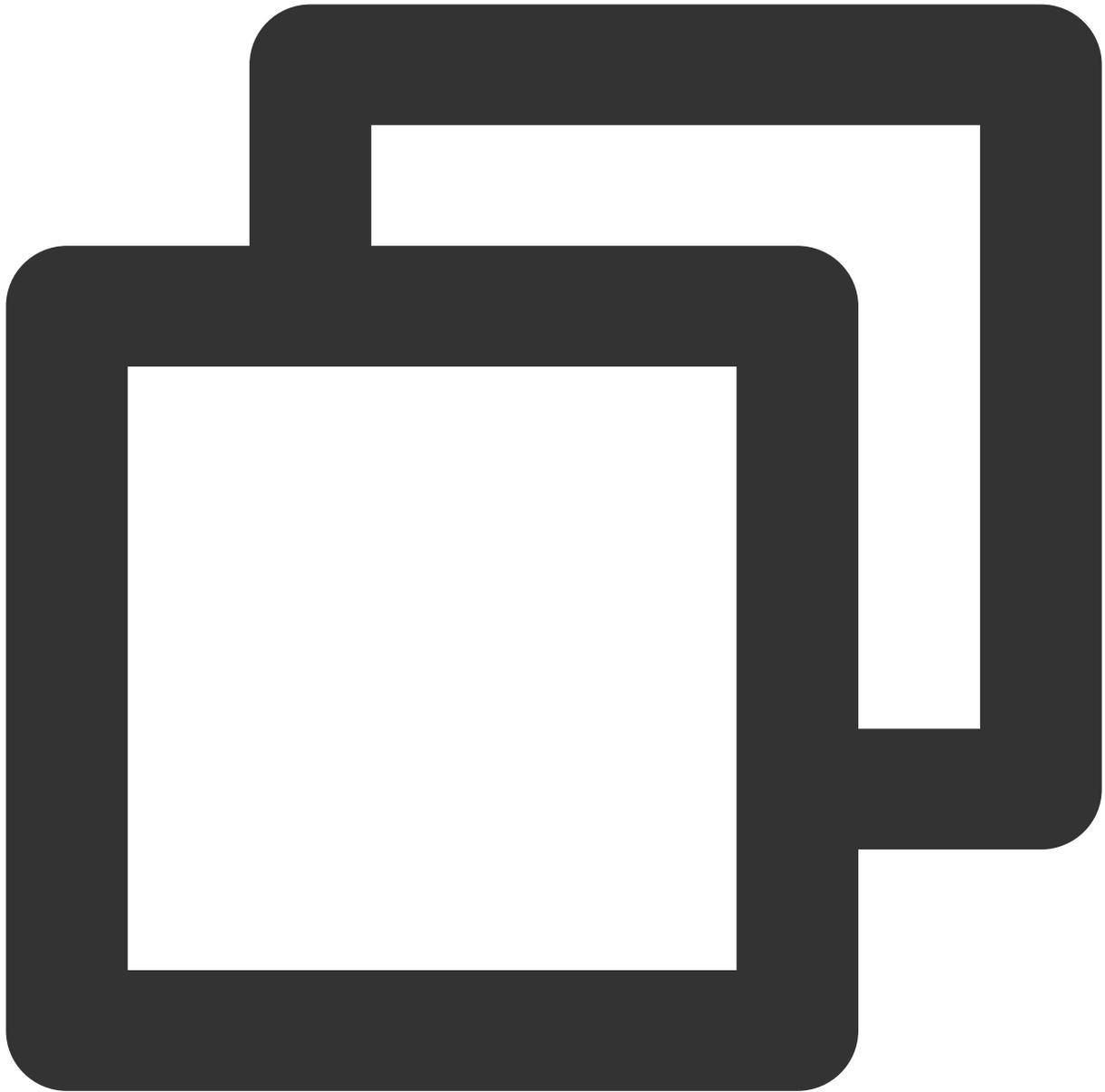


步骤三：导入 SDK

gradle 工程 SDK 远程构建

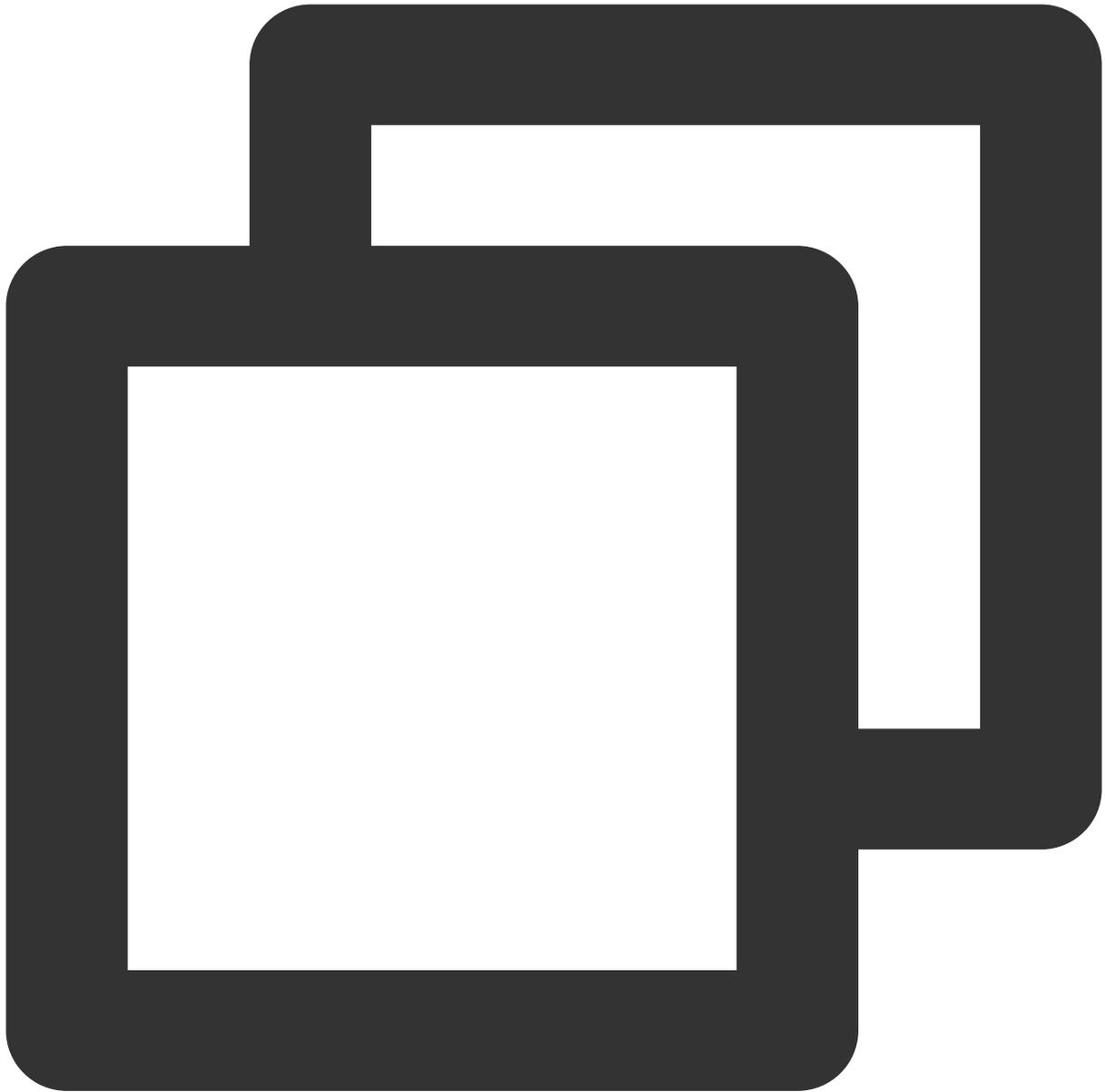
LCIC SDK 已经发布到 mavenCentral 库, 您可以通过配置 gradle 自动下载更新。

1. 应用模块的 build.gradle 中, 在 dependencies 中添加 LCIC SDK 的依赖。



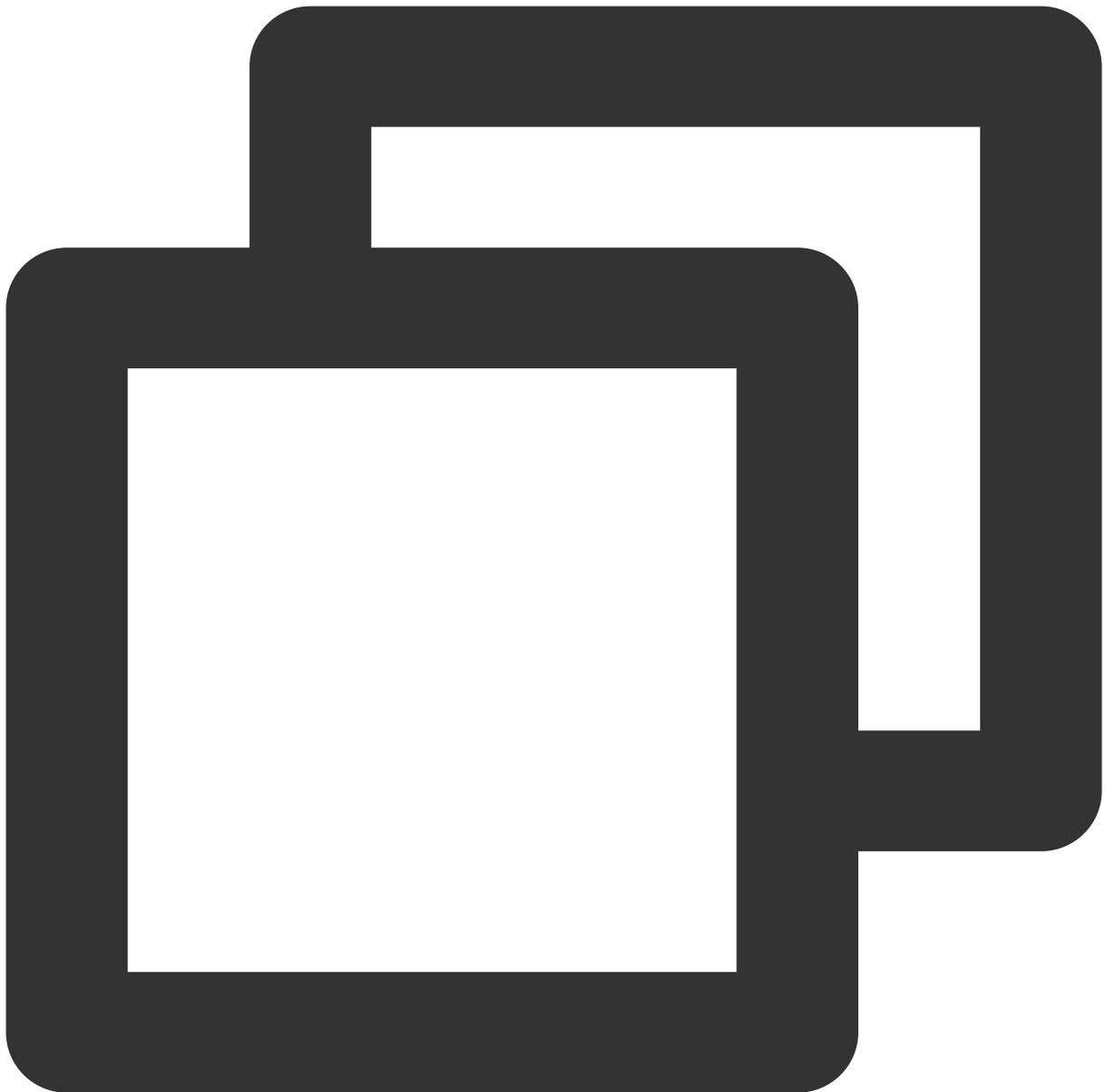
```
dependencies {  
    // LCICSDK 组件  
    implementation 'com.tencent.edu:TCICSDK:1.8.2'  
}
```

2. 在 `defaultConfig` 中，指定 App 使用的 CPU 架构。



```
defaultConfig {  
    ndk {  
        abiFilters "armeabi-v7a", "arm64-v8a"  
    }  
}
```

3. 在 `compileOptions` 中，使用 JDK 1.8 编译。



```
compileOptions {  
    sourceCompatibility 1.8  
    targetCompatibility 1.8  
}
```

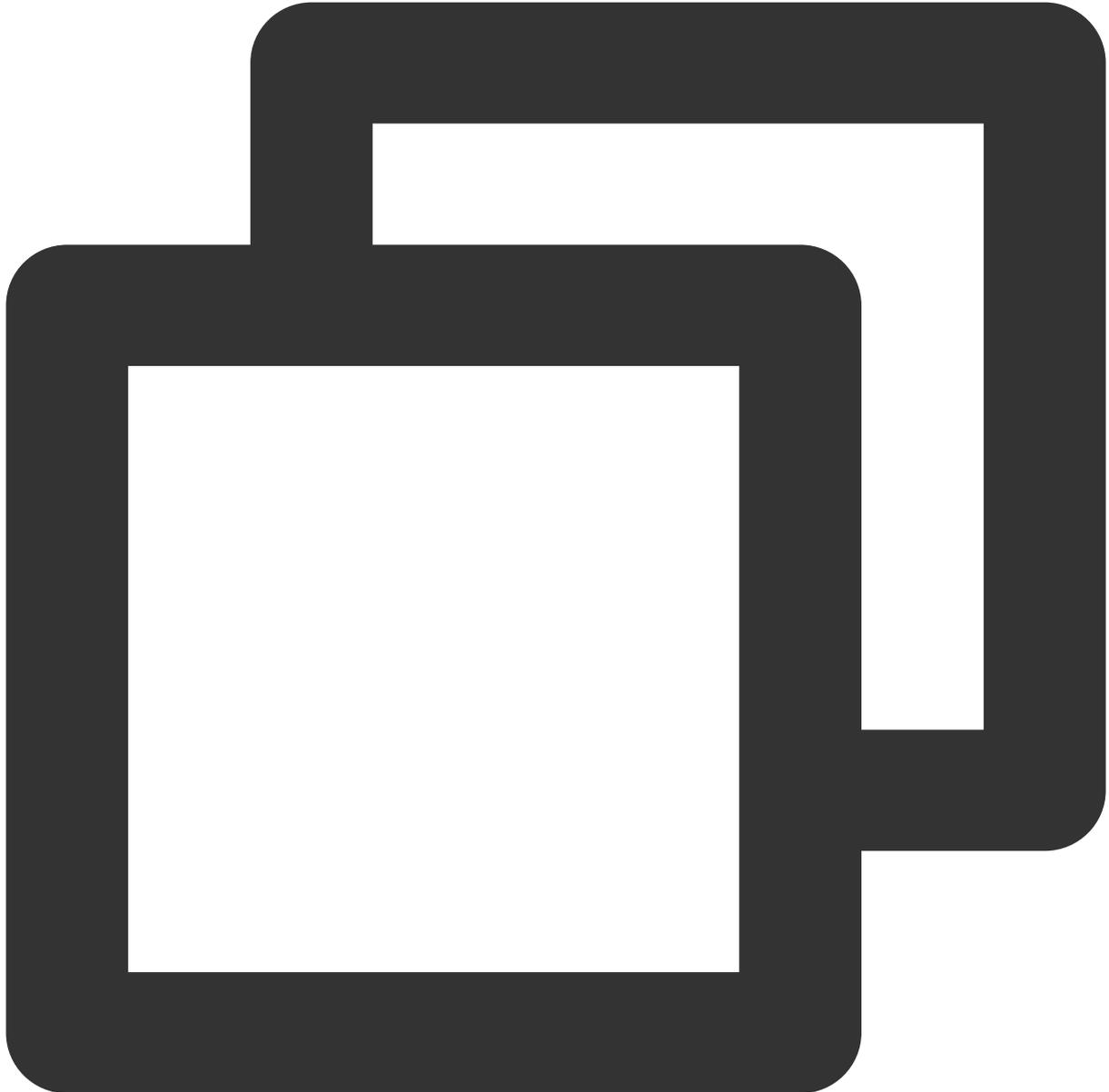
说明：

目前 LCIC SDK 支持 armeabi-v7a 和 arm64-v8a。

4. 单击 **Sync Now**，自动下载 SDK 并集成到工程里。

步骤四：配置 App 权限

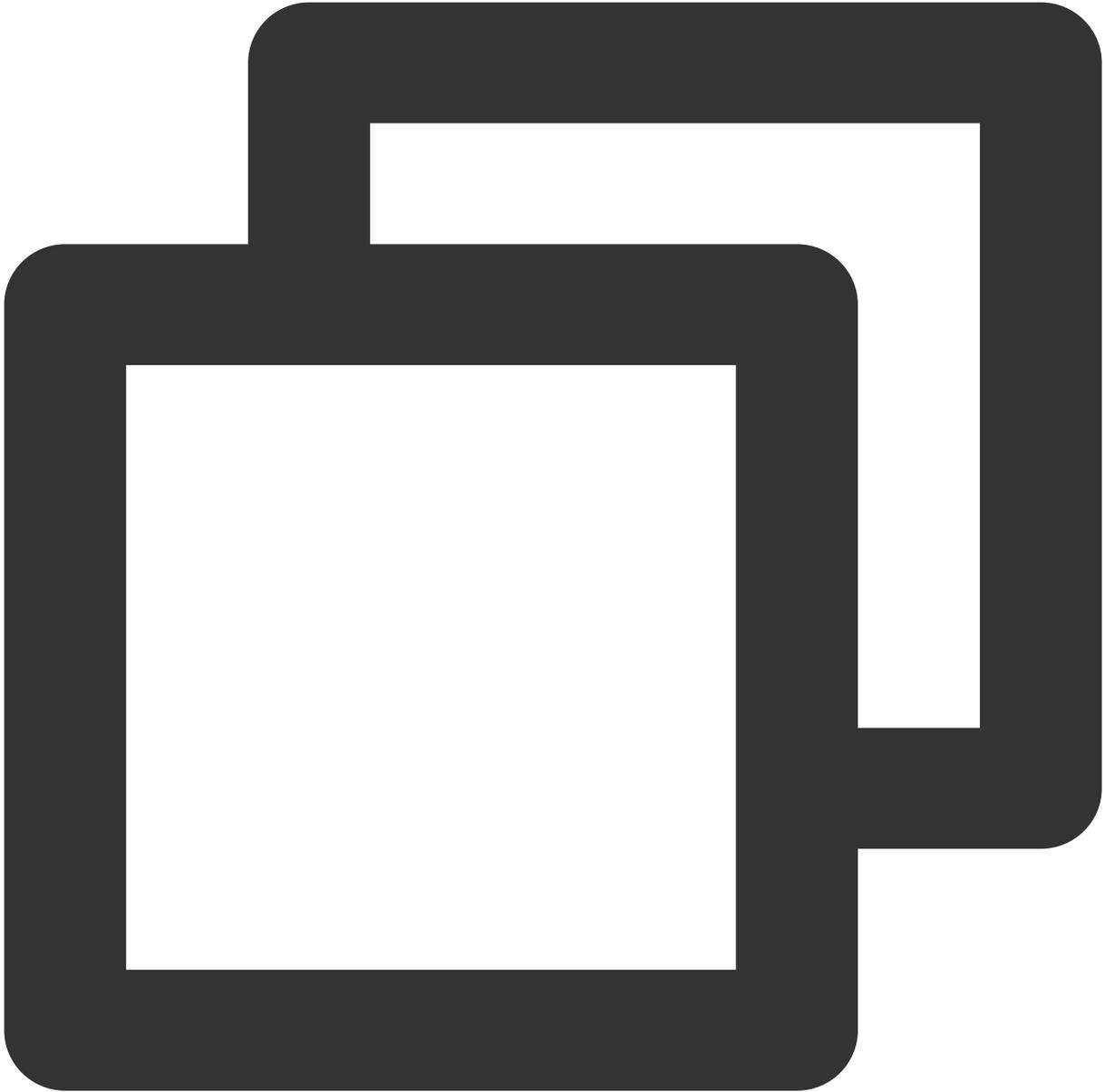
在 AndroidManifest.xml 中配置 App 的权限，LCIC SDK 需要以下权限：



```
<uses-permission android:name="android.permission.RECORD_AUDIO" />  
<uses-permission android:name="android.permission.CAMERA" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

步骤五：设置混淆规则（可选）

如设置了混淆，需在 `proguard-rules.pro` 文件，将 LCIC SDK 相关类加入不混淆名单：



```
-keep class com.tencent.** { *; }
```

步骤六：SDK 授权申请

需要您发送邮件申请 SDK 权限。请按以下模板发送邮件至 mediaservices@tencent.com 联系我们。信息确认无误的情况下我们将会在1个工作日完成。

说明：

包名用于 x5 内核以及快直播播放器签名授权，请提供所需授权的正式应用的 App Name、Package Name 和 Bundle ID 信息，并发送邮件。

收件人：mediaservices@tencent.com

主题

实时互动-教育版 Android SDK 授权申请

内容

公司名称：xxx 有限公司

个人姓名：

联系方式：

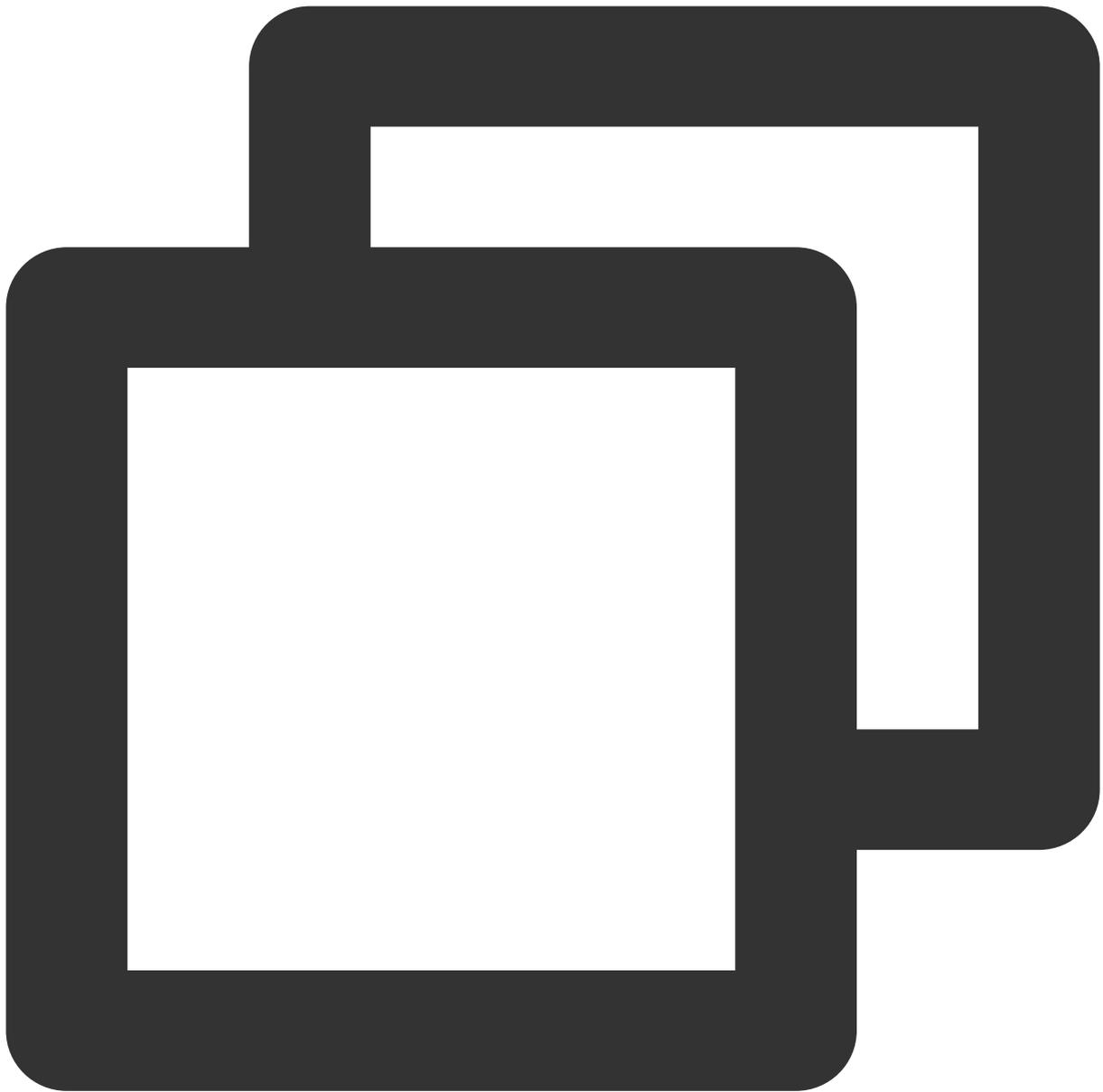
App Name:

Package Name (Android) :

Bundle ID (iOS) :

步骤七：初始化 X5 内核

X5 内核相对于系统 WebView，具有兼容性更好，速度更快等优势。Android 实时互动-教育版 SDK 的组件实现依赖于 x5 内核的 WebView。现提供 x5 内核静态集成方式，能提升 x5 内核加载成功率且无需进程重启即可生效。



```
//初始化x5内核
TCICManager.getInstance().initX5Core(licenseKey, new TBSSdkManageCallback() {

@Override
public void onCoreInitFinished() {
}

@Override
public void onViewInitFinished(boolean isX5Core) {
    if (isX5Core) {
        //x5内核初始化完成,可以进课堂
    }
}
```

```

    }
}
});
    
```

注意：

TCICManager.getInstance().initX5Core(licenseKey); 中的 licenseKey 参数需要通过 [步骤六](#) 发送邮件联系我们获取 X5 内核的 licenseKey。

建议在同意隐私政策协议之后，再调用初始化 X5 内核的方法，以免上架应用市场时未经用户同意，存在收集个人信息的行为。

建议在进入课堂前先判断 x5 内核是否安装成功。判断方法：isX5InstallSuccess。

步骤八：获取进入课堂所需参数

TCICClassConfig 参数解释：

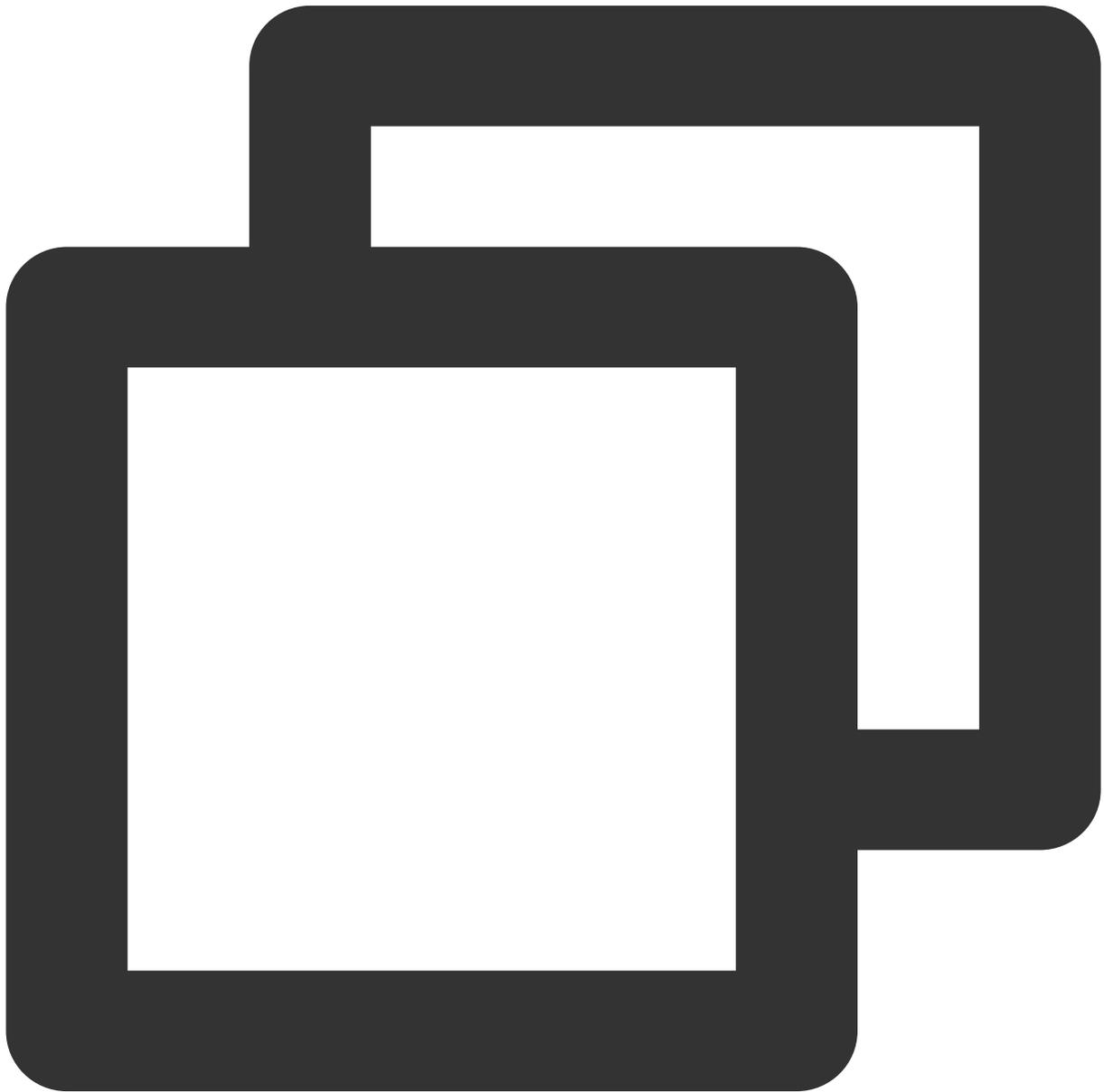
1. 通过 [控制台](#) 进入 [应用管理 > 应用配置](#)，获取 [SDKAppld](#)，即为学校编号(schoolId)信息。
2. 通过云 API 接口 [CreateRoom](#) 创建课堂，可以获取到课堂号(classId)信息。
3. 通过调用云 API 接口 [RegisterUser](#) 注册用户，可以获取到对应的用户 ID(userid)信息。
4. 通过云 API 接口 [LoginUser](#) 登录，可以获取到用户鉴权 token 信息。
5. scene、lng、camera、mic、speaker 为非必要参数，如果不设置则使用的是默认值。

字段	类型	含义	备注	必填
schoolId	int	学校编号	通过控制台进入 应用管理 > 应用配置 ，获取 SDKAppld	是
classId	long	课堂编号	通过 CreateRoom 接口创建返回 RoomId 获取	是
userId	string	用户账号	通过 RegisterUser 接口获取	是
token	string	后台鉴权参数	通过 LoginUser 接口获取	是
scene	string	场景名称	用于区分不同的定制布局，通过 SetAppCustomContent 接口配置	否
lng	string	语言参数	当前支持 zh、en，默认 zh。同时需要设置一下 TCICWebViewManager.getInstance().setClassLanguage(this, env, lng); lng 参数。	否
camera	int	初始化开启摄像头	1为开启摄像头，0为关闭摄像头，默认 1	否
mic	int	初始化开启麦克风	1为开启麦克风，0为关闭麦克风，默认 1	否

speaker	int	初始化开启扬声器	1为开启扬声器，0为关闭扬声器，默认 1	否
---------	-----	----------	----------------------	---

步骤九：调起组件主页面

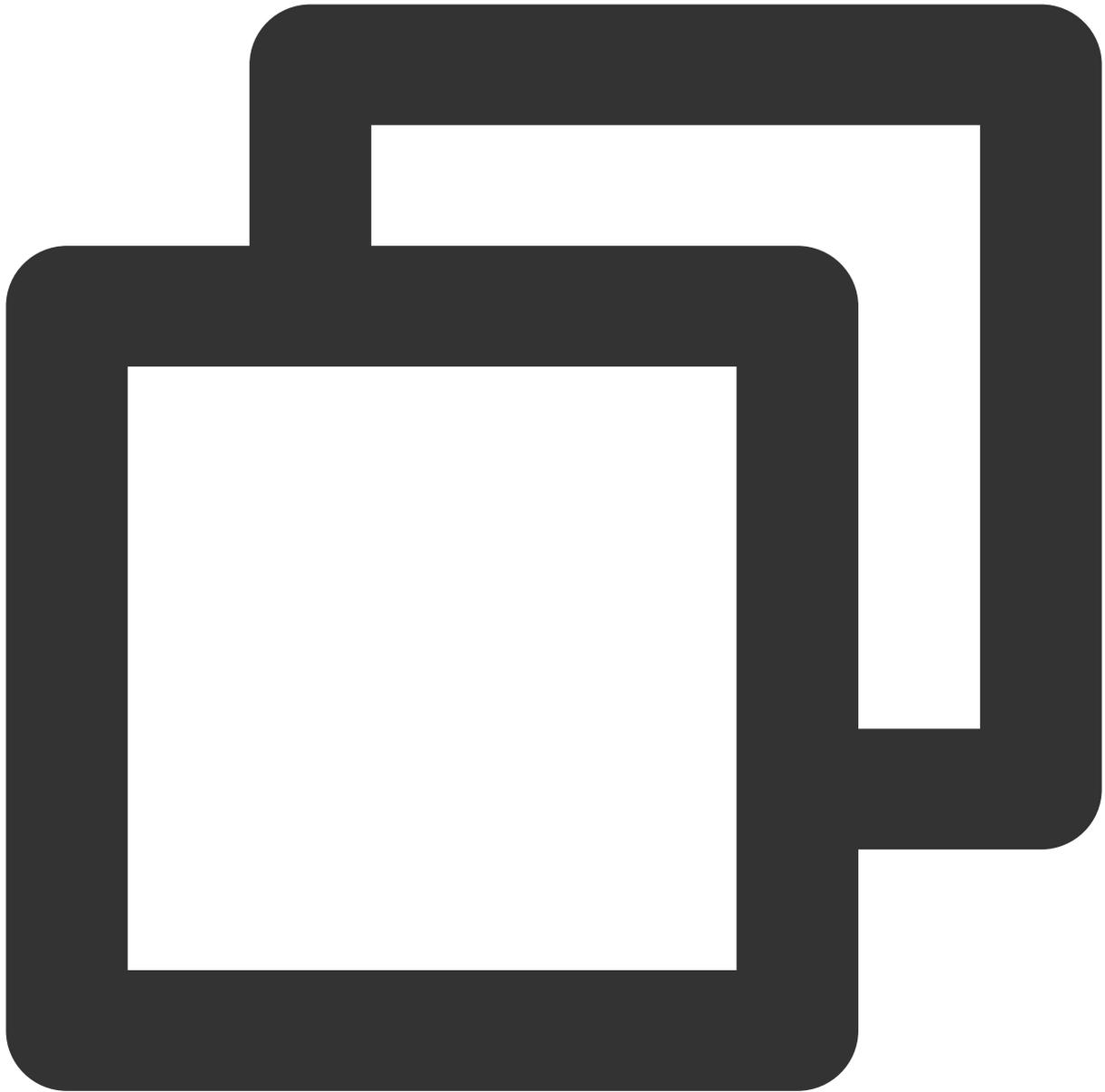
只需传递 4 个参数就可调起 LCIC 组件主页面，分别为学校编号、课堂编号、用户账号和 token。



```
Intent intent = new Intent(getActivity(), TCICClassActivity.class);  
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_SINGLE_TOP);
```

```
Bundle bundle = new Bundle();
TCICClassConfig initConfig = new TCICClassConfig.Builder()
    .schoolId(schoolId)
    .classId(classId)
    .userId(userId)
    .token(token)
    .build();
bundle.putParcelable(TCICConstants.KEY_INIT_CONFIG, initConfig);
intent.putExtras(bundle);
startActivity(intent);
```

如果您需要监听退出课堂通知，可以通过注册 `TCICConstants.ON_CLASS_EXITED_ACTION` 的本地广播，参见如下代码：



```
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(TCICConstants.ON_CLASS_EXITED_ACTION);
LocalBroadcastManager.getInstance(context).registerReceiver(broadcastReceiver, inte
```

其它

我们建议您在使用 LCICSDK 时，同时也接入腾讯 Bugly，帮助您快速发现并解决异常，同时掌握产品运营动态，及时跟进用户反馈。接入指南参见 [腾讯 Bugly 官网](#)。

参见 [开发 Demo](#)。

iOS

最近更新时间：2024-07-26 17:36:13

开发环境要求

Xcode 14

前提条件

您已[注册腾讯云](#)账号，并完成[实名认证](#)。

步骤一：创建新的应用

1. 登录 [实时互动-教育版控制台](#)，左侧导航栏选择**快速跑通应用**。
2. 默认进入“创建应用”界面，应用类型可选择“**创建新应用**”，输入应用名称，例如 TestLCIC。
若您已创建应用，应用类型项可单击“选择已有应用”。

说明：

移动端需要购买旗舰版或企业尊享版后，方可接入移动端。若需创建商用应用，可根据业务需求在[购买页](#)创建对应版本的应用。

3. 根据实际业务需求添加或编辑标签，单击**创建**。

快速跑通应用 体验零代码快速开发

1 创建应用 > 2 生成课堂参数 > 3 跑通应用成功

创建/选择应用

应用类型 创建新应用 选择已有应用

应用名称 0/15

添加标签 (可选填)

标签用于从不同维度对资源分类管理。如现有标签不符合您的要求，请前往 [标签管理](#)
标签更新后，请 [点击此处刷新标签](#)

标签

+添加标签

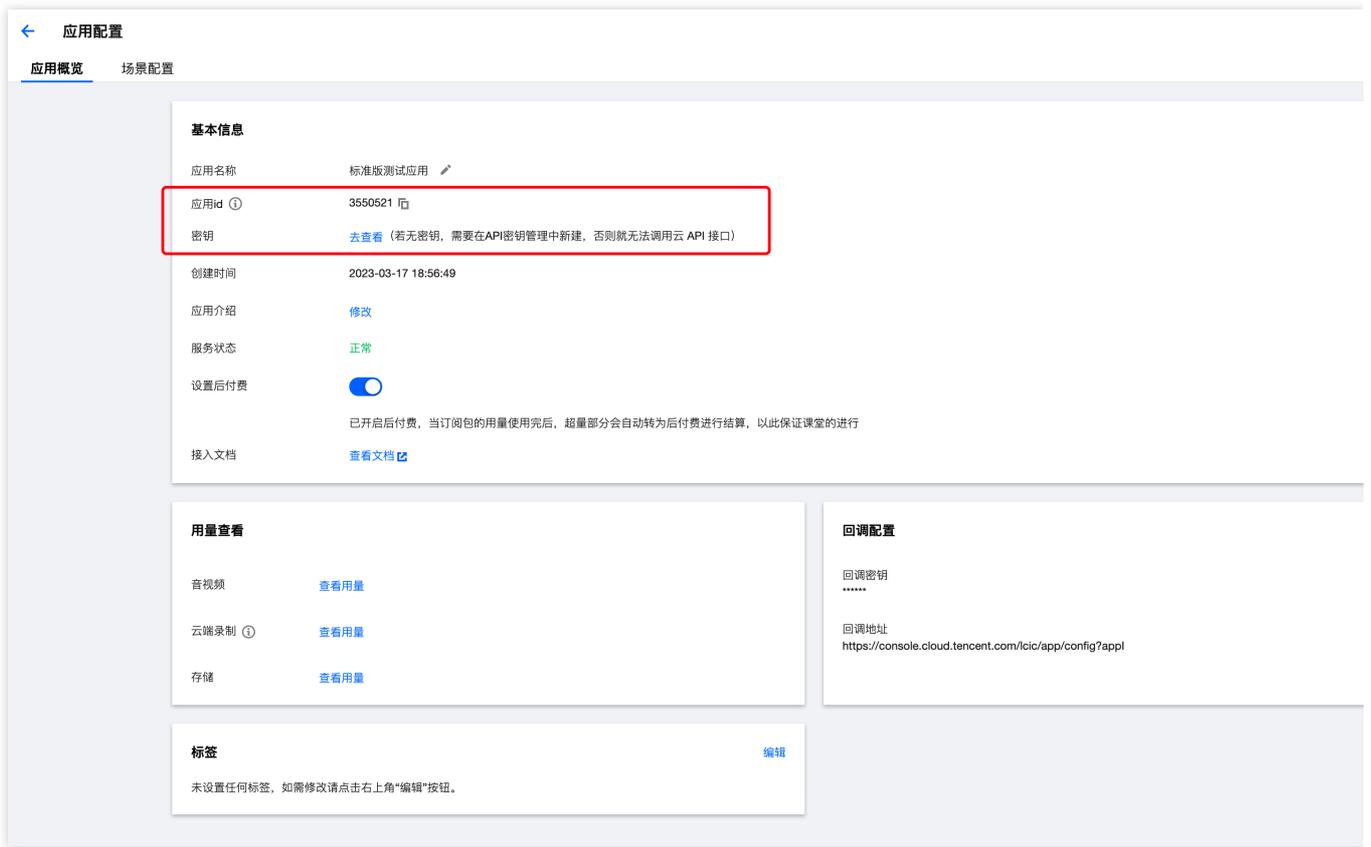
说明：

应用名称只允许下划线、数字或中英文字符。

标签用于标识和组织您在腾讯云的各种资源。例如：企业可能有多个业务部门，每个部门有一个或多个 LCIC 应用，这时，企业可以通过给 LCIC 应用添加标签来标记部门信息。标签并非必选项，您可根据实际业务需求添加或编辑。

步骤二：获取 SDKAppId 和密钥(SecretKey)

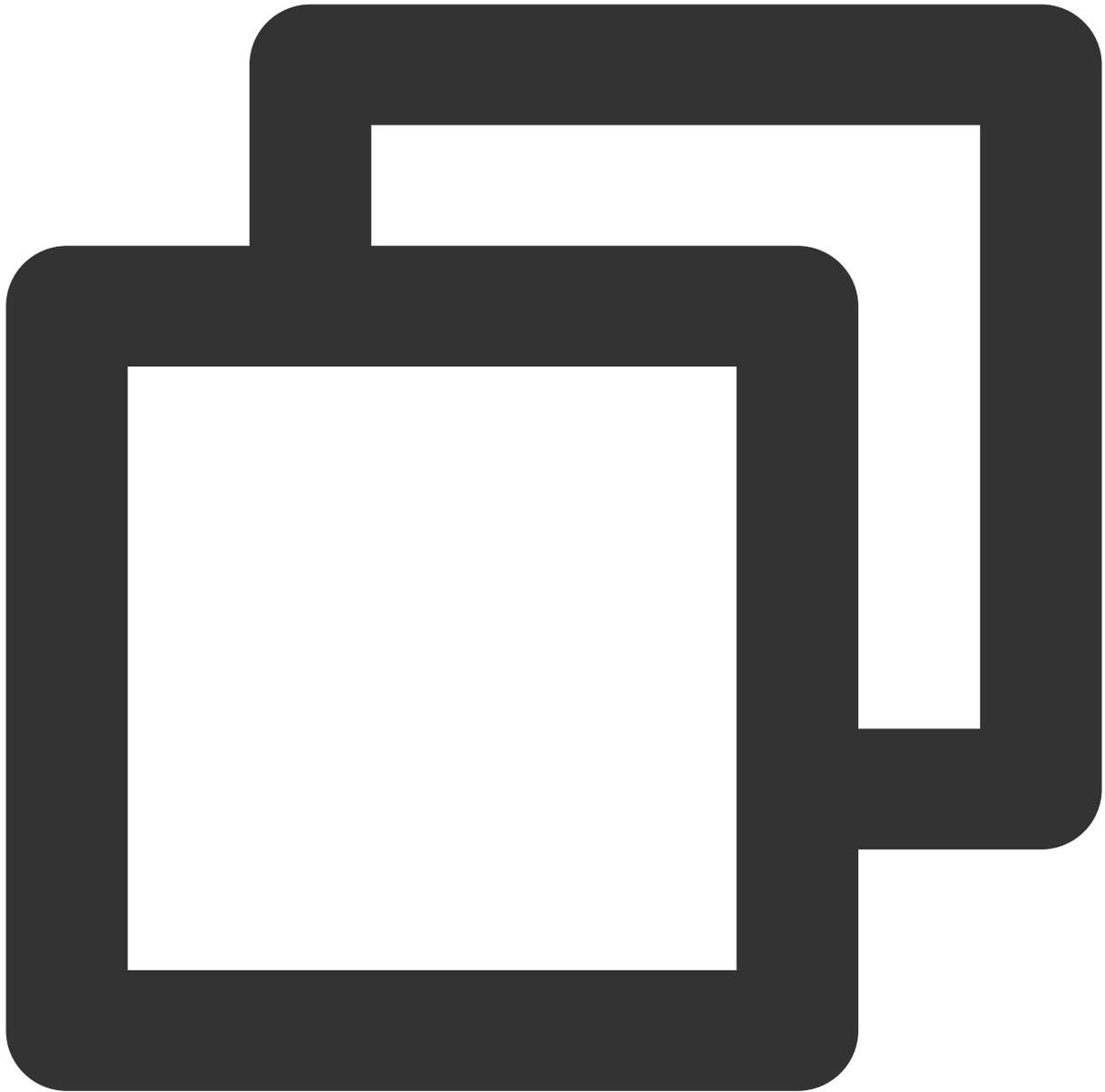
1. 进入 [应用管理](#) > [应用配置](#)，获取 SDKAppId。
2. 进入 [访问管理\(CAM\)控制台](#) 获取密钥，若无密钥，需要在 API 密钥管理中进行新建，具体可参考 [密钥管理](#)。



步骤三：导入SDK

pod 集成 SDK

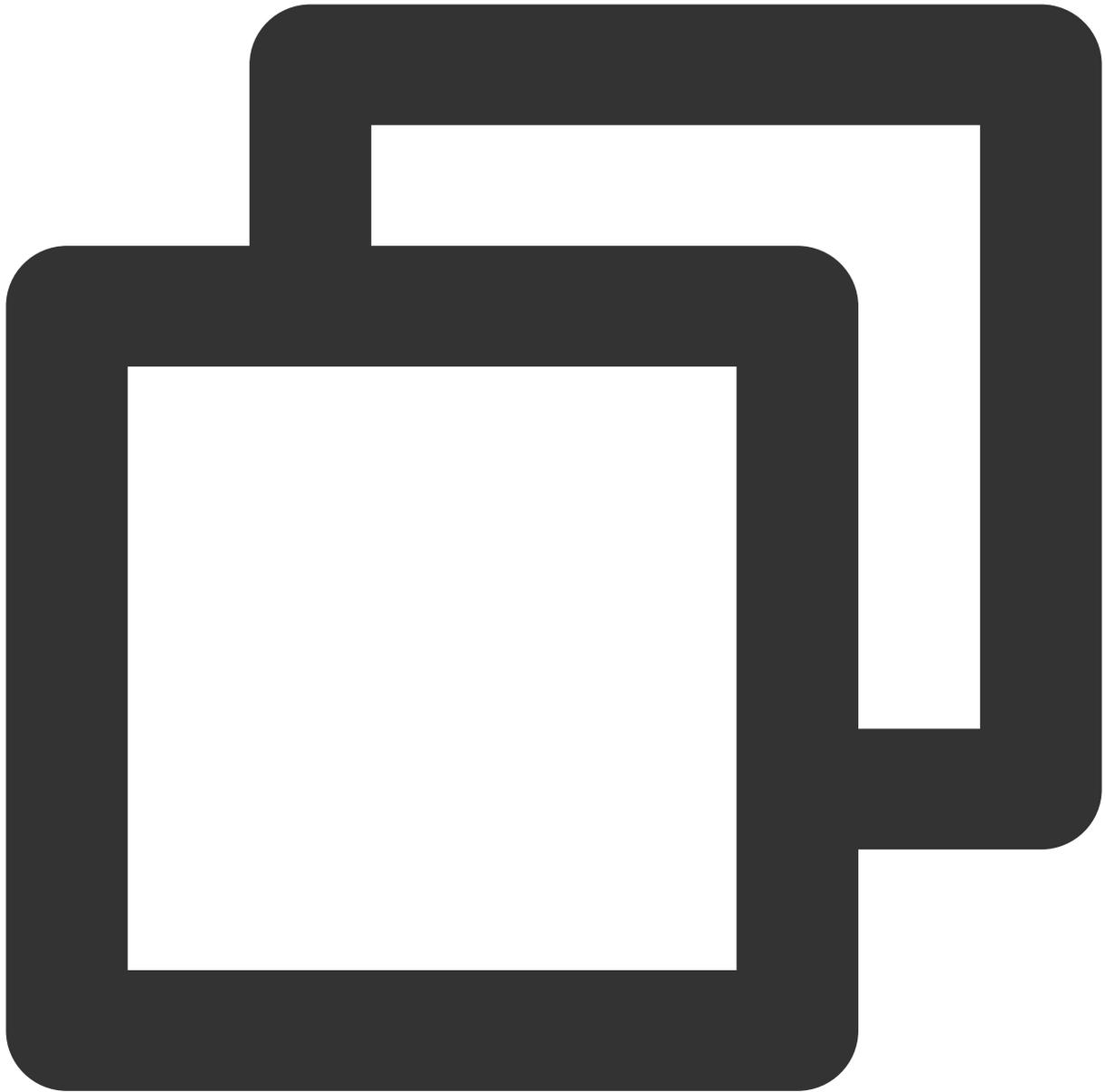
LCIC SDK 已经发布到 cocoapods 库, 您可以通过配置 podfile 下载安装。



```
pod 'TCICSDK_Pro', '1.8.3.3'
```

步骤四：配置 App 权限

在主 App 的 info.plist 中配置 App 的权限，LCIC SDK 需要以下权限：



```
<key>NSCameraUsageDescription</key>  
<key>NSMicrophoneUsageDescription</key>  
<key>NSPhotoLibraryAddUsageDescription</key>  
<key>NSPhotoLibraryUsageDescription</key>
```

步骤五：获取进入课堂所需参数

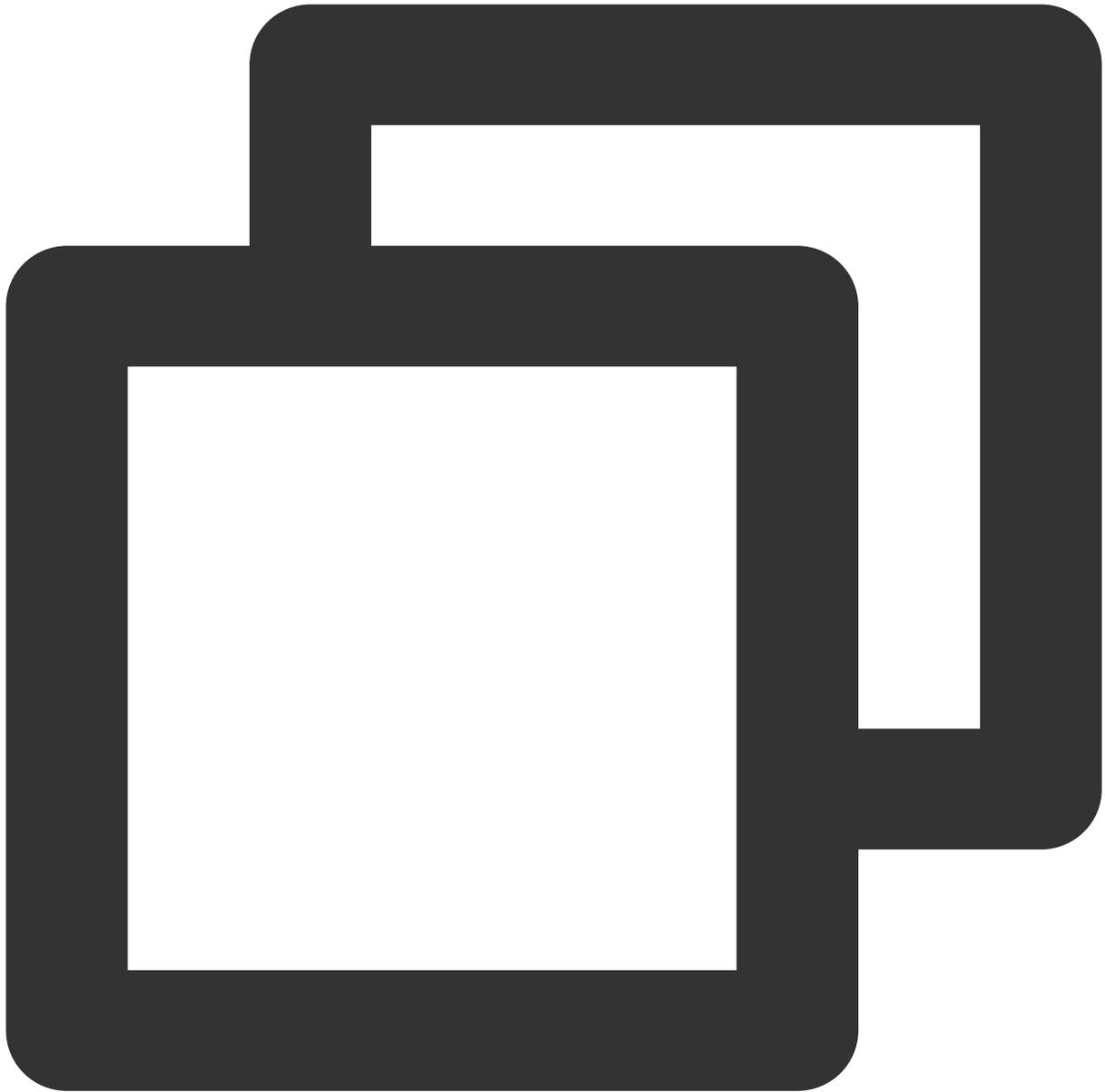
1. 通过 [控制台](#) 进入 [应用管理](#) > [应用配置](#)，获取 [SDKAppId](#)，即为学校编号(schoolId)信息。

2. 通过云 API 接口 [CreateRoom](#) 创建课堂，可以获取到课堂号(classid)信息。
3. 通过调用云 API 接口 [RegisterUser](#) 注册用户，可以获取到对应的用户 ID(userid)信息。
4. 通过云 API 接口 [LoginUser](#) 登录，可以获取到用户鉴权 token 信息。
5. scene、lng、camera、mic、speaker 为非必要参数，如果不设置则使用的是默认值。

字段	类型	含义	备注	必填
schoolId	int	学校编号	通过控制台进入 应用管理 > 应用配置 ，获取 SDKAppld	是
classId	long	课堂编号	通过 CreateRoom 接口创建返回 RoomId 获取	是
userId	string	用户账号	通过 RegisterUser 接口获取	是
token	string	后台鉴权参数	通过 LoginUser 接口获取	是
scene	string	场景名称	用于区分不同的定制布局，通过 SetAppCustomContent 接口配置 [roomConfig setValue:@"scene_name" forKey:@"scene"];	否
lng	string	语言参数	可传入zh-Hans、en，默认为中文（zh-Hans）。 可通过此接口设置 [roomConfig setValue:@"en" forKey:@"language"];	否
camera	int	初始化开启摄像头	1为开启摄像头，0为关闭摄像头，可通过 roomConfig.jsParams 设置	否
mic	int	初始化开启麦克风	1为开启麦克风，0为关闭麦克风，可通过 roomConfig.jsParams 设置	否
speaker	int	初始化开启扬声器	1为开启扬声器，0为关闭扬声器，可通过 roomConfig.jsParams 设置	否

步骤六：调起组件主页面

只需传递 4 个参数就可调起 LCIC 组件主页面，分别为学校编号、课堂编号、用户账号和 token。

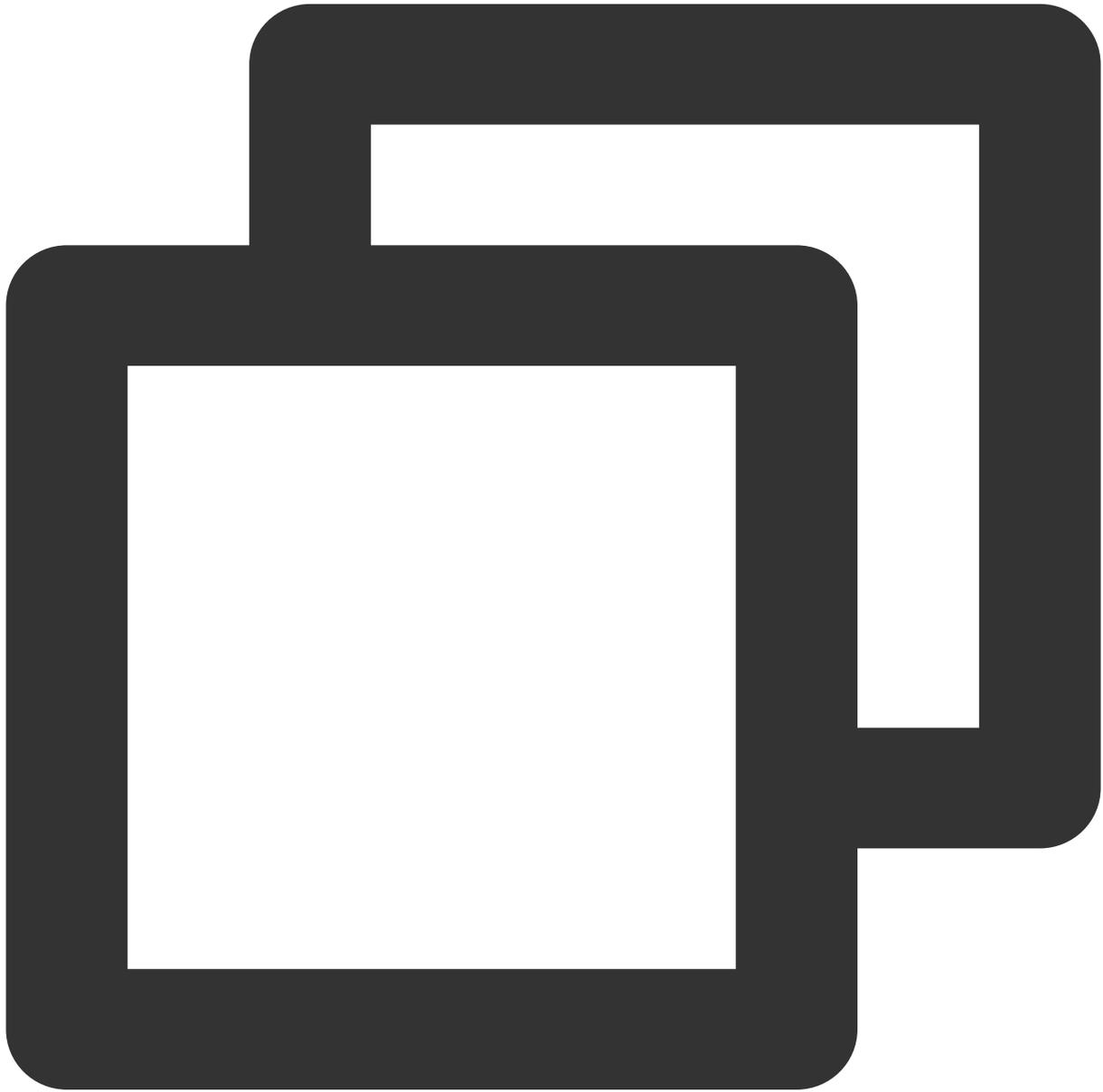


```
TCICClassConfig *roomConfig = [[TCICClassConfig alloc] init];
roomConfig.schoolId = 123456;
roomConfig.userId = "test";
roomConfig.token = "test_token";
roomConfig.classId = 654321;
[roomConfig setValue:@"en" forKey:@"language"]; //语言设置, 可选
[roomConfig setValue:@"scene_name" forKey:@"scene"]; //可根据场景配置不同的定制, 可选
[roomConfig setValue:@(0) forKey:@"preferPortrait"]; //默认横屏, 可选 (0 是横屏, 1是竖屏

TCICClassController *vc = [TCICClassController classRoomWithConfig:roomConfig];
if (vc) {
```

```
[(UINavigationController *)self.window.rootViewController pushViewController:vc  
}else {  
    NSLog(@"参数有误");  
}
```

如果您需要监听退出课堂通知，可以通过注册 TCICExitClassRoomCompleteNotify 的本地广播，参见如下代码：



```
[[NSNotificationCenter defaultCenter] addObserver:self  
                                         selector:@selector(yourselector)  
                                         name:@"TCICExitClassRoomCompleteNotify" objec
```

步骤七：SDK授权申请

需要您发送邮件申请 SDK 权限。请按以下模板发送邮件至 mediaservices@tencent.com 联系我们。信息确认无误的情况下我们将会在1个工作日完成。

说明：

包名用于快直播播放器签名授权， 请提供所需授权的正式应用的 App Name、Package Name 和 Bundle ID 信息，并发送邮件。

收件人：mediaservices@tencent.com

主题

实时互动-教育版 iOS SDK 授权申请

内容

公司名称：xxx 有限公司

个人姓名：

联系方式：

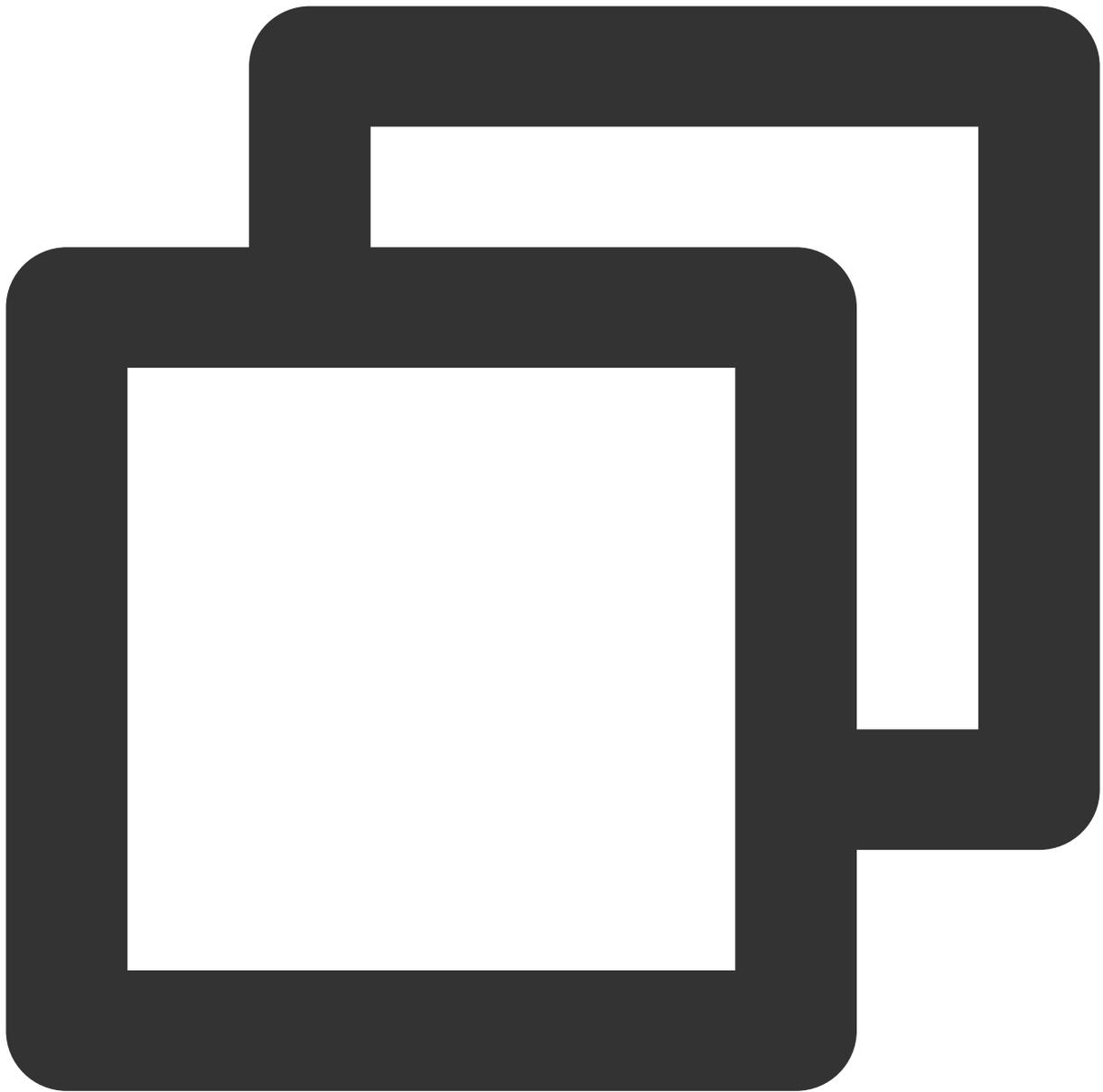
App Name:

Package Name (Android) :

Bundle ID (iOS) :

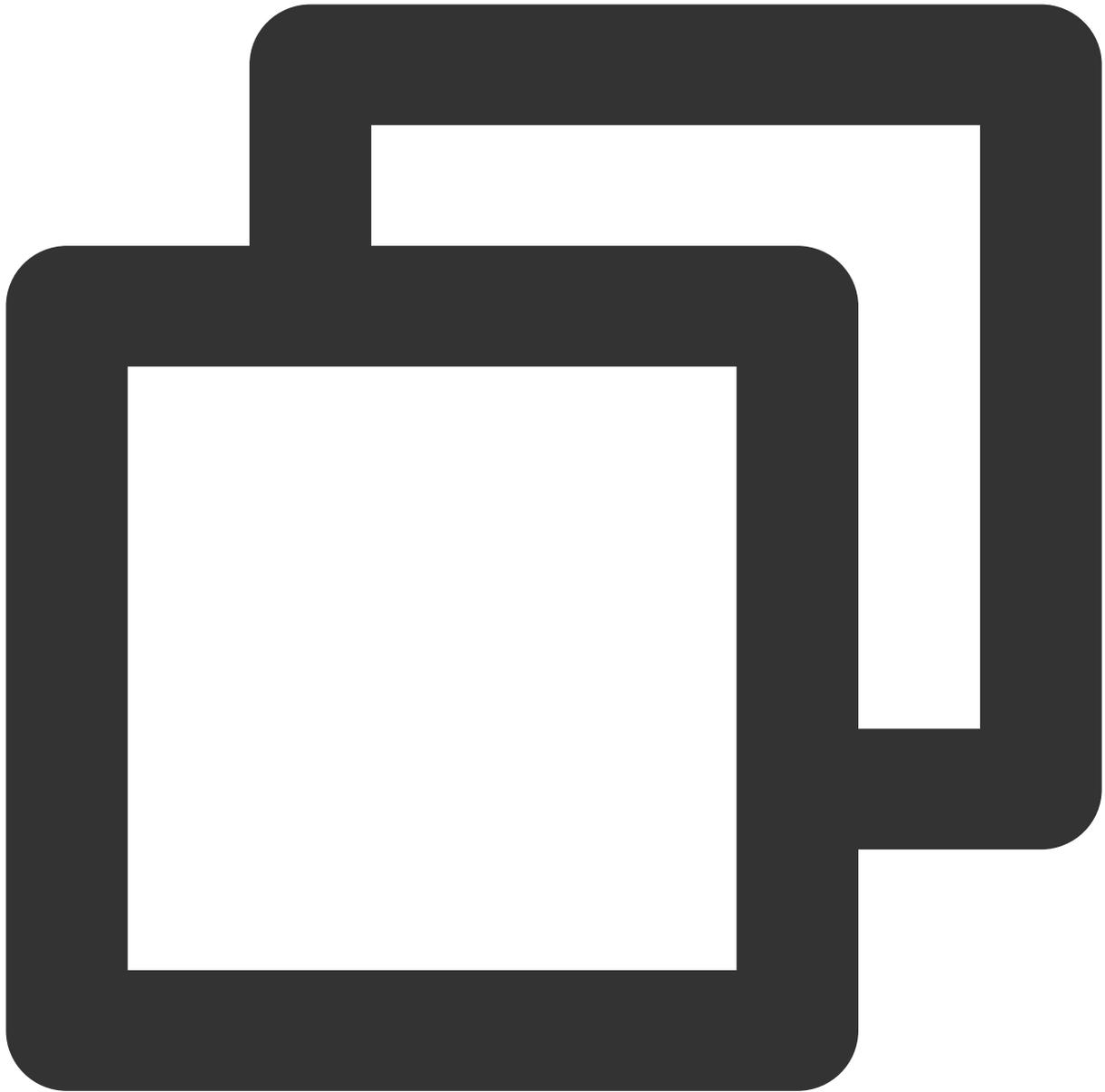
高级功能：移动端屏幕分享

1. 创建 `App Group` , 参见 [TRTC 文档](#)。
2. 创建 `Broadcast Upload Extension` , 参见 [TRTC 文档](#)。
3. 为新创建的 `Target` , 依赖依赖 `TCICSDK_ReplayKit` , 如下, 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_Pro_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 *> *)setupI
```

```

[[TXReplayKitExt sharedInstance] setupWithAppGroup:APPGROUP delegate:self];

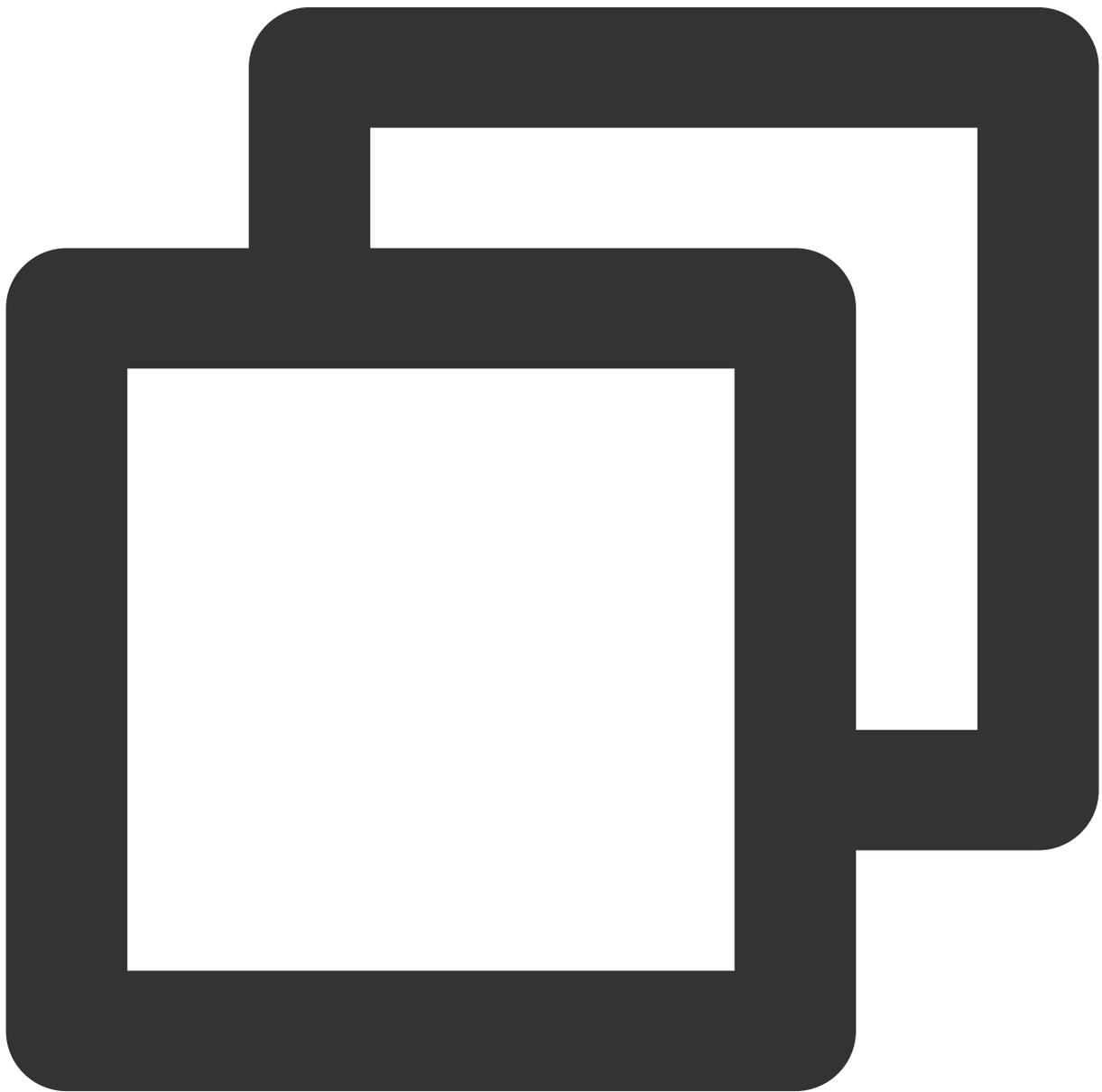
[[TCICScreenKit sharedScreenKit] onScreenKitStarted];
}
- (void)broadcastPaused {
    // User has requested to pause the broadcast. Samples will stop being delivered
    [[TCICScreenKit sharedScreenKit] onScreenKitPaused];
}
- (void)broadcastResumed {
    // User has requested to resume the broadcast. 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)
{
    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
        NSLocalizedFailureReasonErrorKey:tip
    ];
    [self finishBroadcastWithError:error];
}
- (void)processSampleBuffer:(CMSampleBufferRef)sampleBuffer withType:      (RPSampleB
    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 即可，且在进入课堂前，设置 AppGroup 即可。



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

// 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 [TRTC 文档](#)（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。

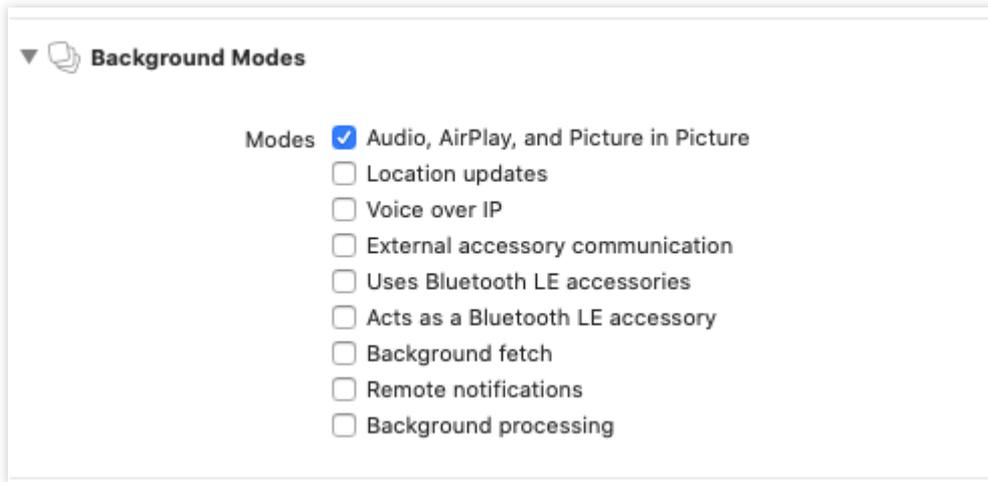


1.2 对于 iOS11 的机型，需要业务侧引导用户从远程控制中**长按录屏**进行触发，并选择业务自创建的 `Broadcast Upload Extension` 进行触发，下图以**腾讯会议**为例：



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`。



其它

我们建议您在使用 LCICSDK 时，同时也接入腾讯 Bugly，帮助您快速发现并解决异常，同时掌握产品运营动态，及时跟进用户反馈。接入指南参见 [腾讯 Bugly 官网](#)。

参见 [开发 Demo](#)。

Windows 和 macOS

最近更新时间：2024-06-28 10:11:10

为提供跨平台的桌面应用程序，Windows 和 macOS 端使用了 Electron 的方案进行开发，本文主要介绍如何快速将腾讯云 LCIC Electron SDK 集成到您的项目中。

前提条件

您已经 [注册腾讯云](#) 账号，并完成 [实名认证](#)。

步骤一：创建新的应用

1. 登录 [实时互动-教育版控制台](#)，在左侧导航栏选择**快速跑通应用**。
2. 默认进入“创建应用”界面，应用类型可选择“**创建新应用**”，输入应用名称，例如 TestLCIC。
若您已创建应用，应用类型项可单击“关联已有应用”。

说明：

每个账号可免费创建一个试用应用，若当前账号下无试用应用，可直接创建试用应用进行体验。若需创建商用应用，可根据业务需求在 [购买页](#) 创建对应版本的应用。

3. 根据实际业务需求添加或编辑标签，单击创建。

快速跑通应用 体验零代码快速开发

1 创建应用 > 2 生成课堂参数 > 3 跑通应用成功

创建/选择应用

应用类型 创建新应用 选择已有应用

应用名称 0/15

添加标签 (可选填)

标签用于从不同维度对资源分类管理。如现有标签不符合您的要求，请前往 [标签管理](#)

标签更新后，请 [点击此处刷新标签](#)

标签

+添加标签

说明：

应用名称只允许下划线、数字或中英文字符。

标签用于标识和组织您在腾讯云的各种资源。例如：企业可能有多个业务部门，每个部门有1个或多个 LCIC 应用，这时，企业可以通过给 LCIC 应用添加标签来标记部门信息。标签并非必选项，您可根据实际业务需求添加或编辑。

步骤二：获取 SDKAppId 和密钥(SecretKey)

1. 进入 [应用管理](#) > [应用配置](#)，获取 SDKAppId。
2. 进入 [访问管理\(CAM\)控制台](#) 获取密钥，若无密钥，需要在 API 密钥管理中进行新建，具体可参见 [密钥管理](#)。



步骤三：获取进入课堂所需参数

1. 通过调用云 API 接口 [RegisterUser](#) 注册用户，可以获取到对应的用户 ID(`userid`)信息。
2. 通过云 API 接口 [LoginUser](#) 登录，可以获取到用户鉴权 `token` 信息。
3. 通过云 API 接口 [CreateRoom](#) 创建课堂，可以获取到课堂号(`classid`)信息。
4. 明确需要集成的课堂版本：`latest`。
5. 其中 `scene`、`debugjs`、`debugcss` 为非必填参数，在需要自定义 UI 时才需设置，具体可参考 [自定义 UI 集成](#)。其中 `debugjs` 和 `debugcss` 只用于自定义布局、组件时的调试，且只支持通过 `localhost` 或 `127.0.0.1` 的地址进行访问，在发布阶段请勿使用此参数。
6. `lng`、`location`、`layout` 也是非必填参数，业务侧可自行判断是否需要传入，不传则使用默认值，其中 `layout` 参数只有在教室布局为视频+文档布局(`videodoc`)时才生效。

字段	类型	含义	备注	必填
<code>userid</code>	<code>string</code>	用户名	通过 RegisterUser 接口获取	是

classid	string	课堂 ID	通过 CreateRoom 接口创建返回获取	是	
token	string	后台鉴权参数	通过 LoginUser 接口获取	是	
version	string	课堂版本号	(从tcic-electron-sdk v1.9.0版本起将正式废弃此参数) 通过发布日志选择对应版本	否	
scene	string	场景名称	区分不同的定制布局, 通过 SetAppCustomContent 接口配置	否	
role	string	进入课堂角色, 默认空	可选参数 supervisor (巡课/内容审查) 只有已注册应用内巡课用户才有权限	否	
debugjs	string	自定义 UI 的 JS 链接	通过自定义 UI 集成方式获取	否	
debugcss	string	自定义 UI 的 CSS 链接	通过自定义 UI 集成方式获取	否	
lng	string	语言参数, 默认 zh-CN	当前支持 zh-CN、en-US	否	
location	boolean	是否上报经纬度位置信息	默认 false 不上报	否	
layout	string	页面布局	默认顶部布局(top), 当前支持双排布局(double)、右侧布局(right)、左侧布局(left)、三分布局(three)	否	

步骤四：进入课堂

针对不同业务场景需要, 我们提供了以下两种接入集成方式。

方式一：URL 拼接

通过浏览器呼起客户端的能力。客户端分为腾讯云提供的 LCIC-Demo 客户端和自定义客户端。区别在于客户端图标、名称、初始页面不一样。

使用浏览器直接呼起客户端

1. 下载客户端并安装。

[Windows 64位](#)

[Windows 32位](#)

[macOS](#)

2. 通过链接唤起客户端，当用户在浏览器中访问或跳转到

URL `tcic://class.qcloudclass.com/latest/class.html?`

`classid=${classId}&userid=${userId}&token=${token}` 上时，浏览器会请求打开 LCIC-Demo 客户端。

中转页唤起方式（推荐）

通过简单快速集成我们的 LCIC Electron SDK，业务侧实现跳转中转页，识别成功就直接跳转，失败则优先下载客户端。如果用户取消或者不下载，再通过跳转打开 web 版本的上课页面。

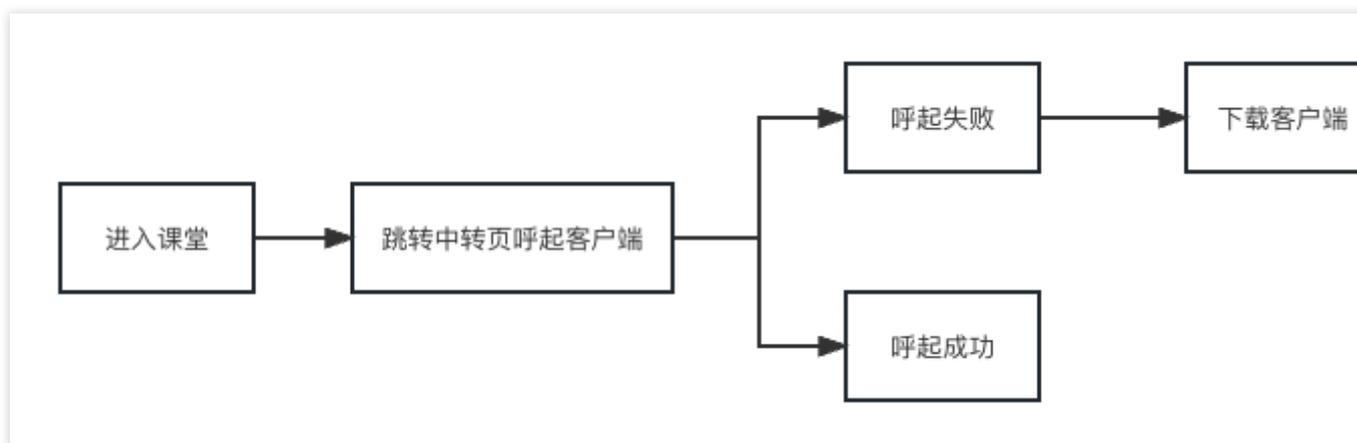
具体流程：

在进入课堂时需先跳转至一个中转页，在中转页内处理用户跳转逻辑。获取上课参数拼接到此

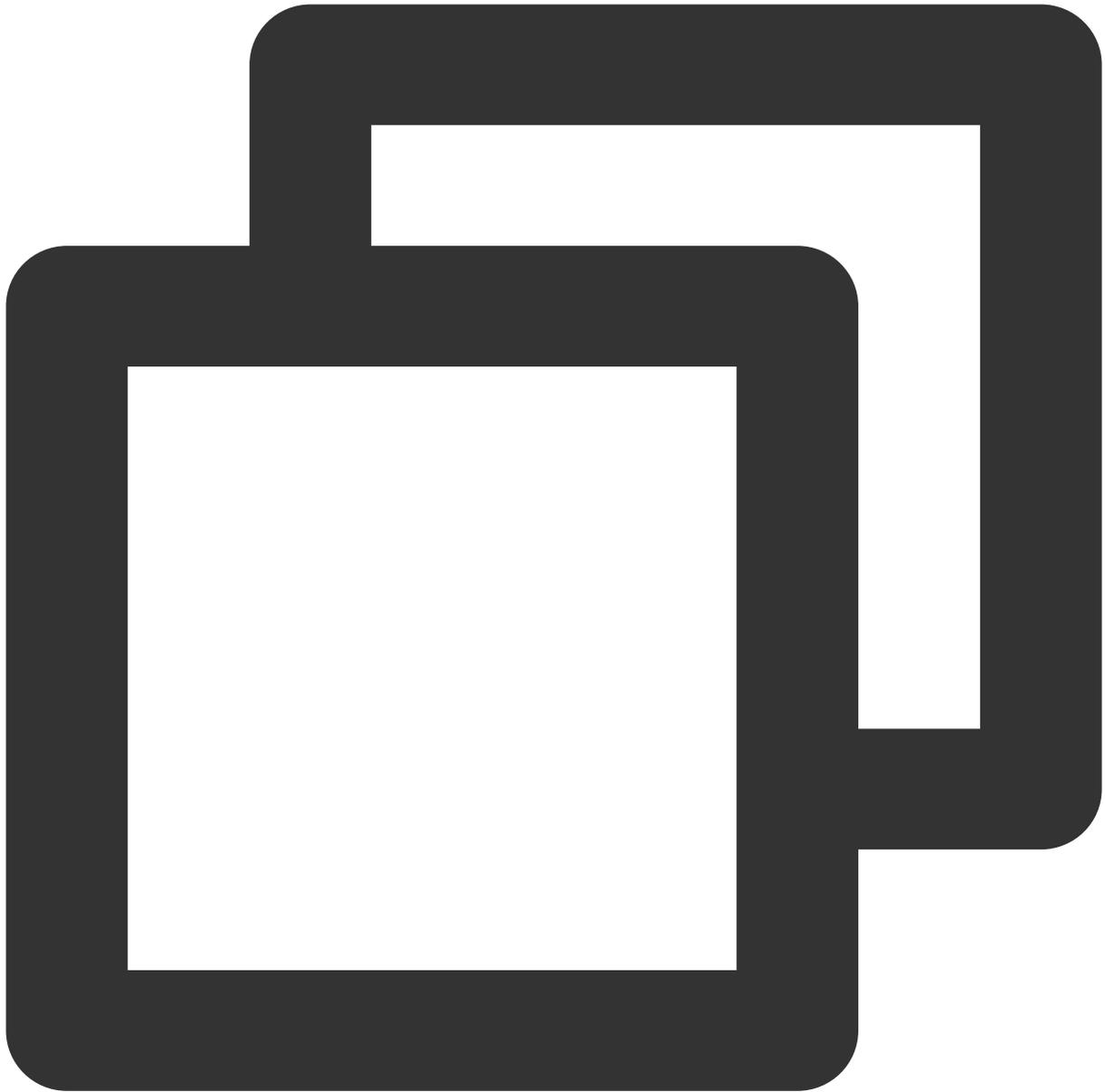
URL `tcic://class.qcloudclass.com/latest/class.html?`

`classid=${classId}&userid=${userId}&token=${token}` 上，即可唤起客户端应用。

流程可参见下图：



下载 [ElectronProtocolCheck.js](#) 引入自己的项目中。中转页中的示例代码如下：



```
// ElectronProtocolCheck 文件代码见github示例
import ElectronProtocolCheck from './ElectronProtocolCheck';

// 需要进入课堂的id
const classId = 368507569;
// 当前进入课堂用户的id
const userId = "JIUzI1NiIsIn123456";
// token 需要动态从后台接口获取, 防止登录态过期失效
const token = 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhhd
const url = `tcic://class.qcloudclass.com/latest/class.html?classid=${classId}&us
console.log(`callClient->start: ${url}`);
```

```

// 唤起客户端
ElectronProtocolCheck (
    url,
    successCallBack: (res) => {
        // 呼起成功
        console.log('callClient->success! ');
    },
    failCallBack: (res) => {
        console.log('callClient->failed! ');
        // 没有呼起来, 建议此处实现提示下载的弹窗

        // 如果用户点击下载 --> 换弹窗口信息, 打开客户端。

        // 如果用户取消下载或关闭弹窗。在让用户尝试体验web/H5 版本的课堂
        // 也可以加上超时弹窗 一般 2500ms 内用户没有做出点击操作, 可以出现模态点击框
    },
    unsupportedCallBack: () {
        // 浏览器不支持等情况, 可以走 web 链接加载
    });
    
```

我们为您提供完整的 Demo, 单击 [LCIC-Electron-Demo](#) 查看。

说明：

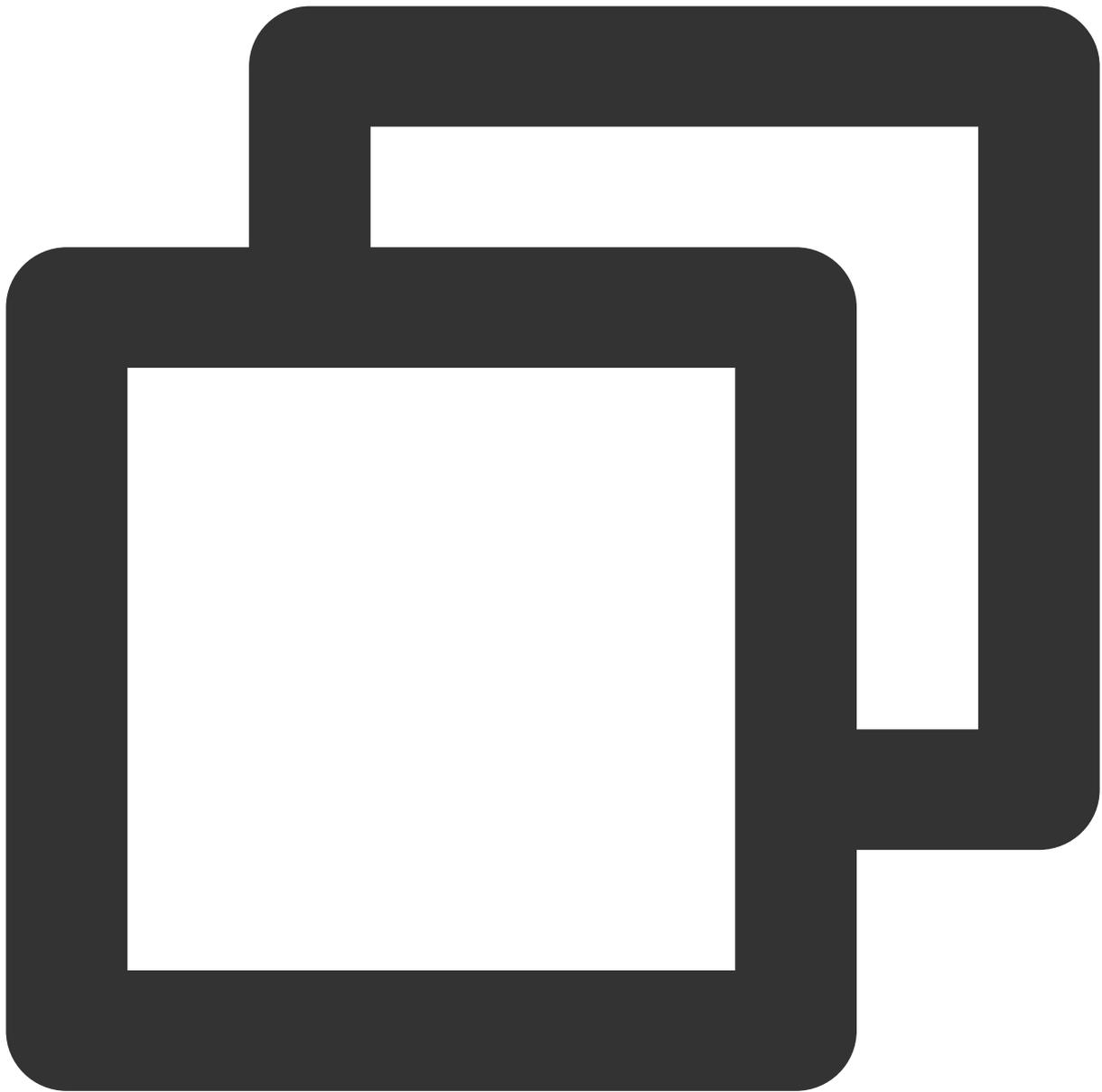
在拼接的 URL 中, 若传入的 `userid` 与当前课堂指定的 `teacherid` 是一致的, 则当前用户为老师。若与当前课堂的助教 ID(`assistantid`)一致, 则为助教, 否则为学生。

方式二：SDK 集成

1. 由于部分依赖库的原因, 请确保您使用以下基础库版本。

开发框架	版本
Electron	10.4.3+
Node	10.14.1+

2. 在您的项目中使用 `npm` 命令安装 SDK 包。

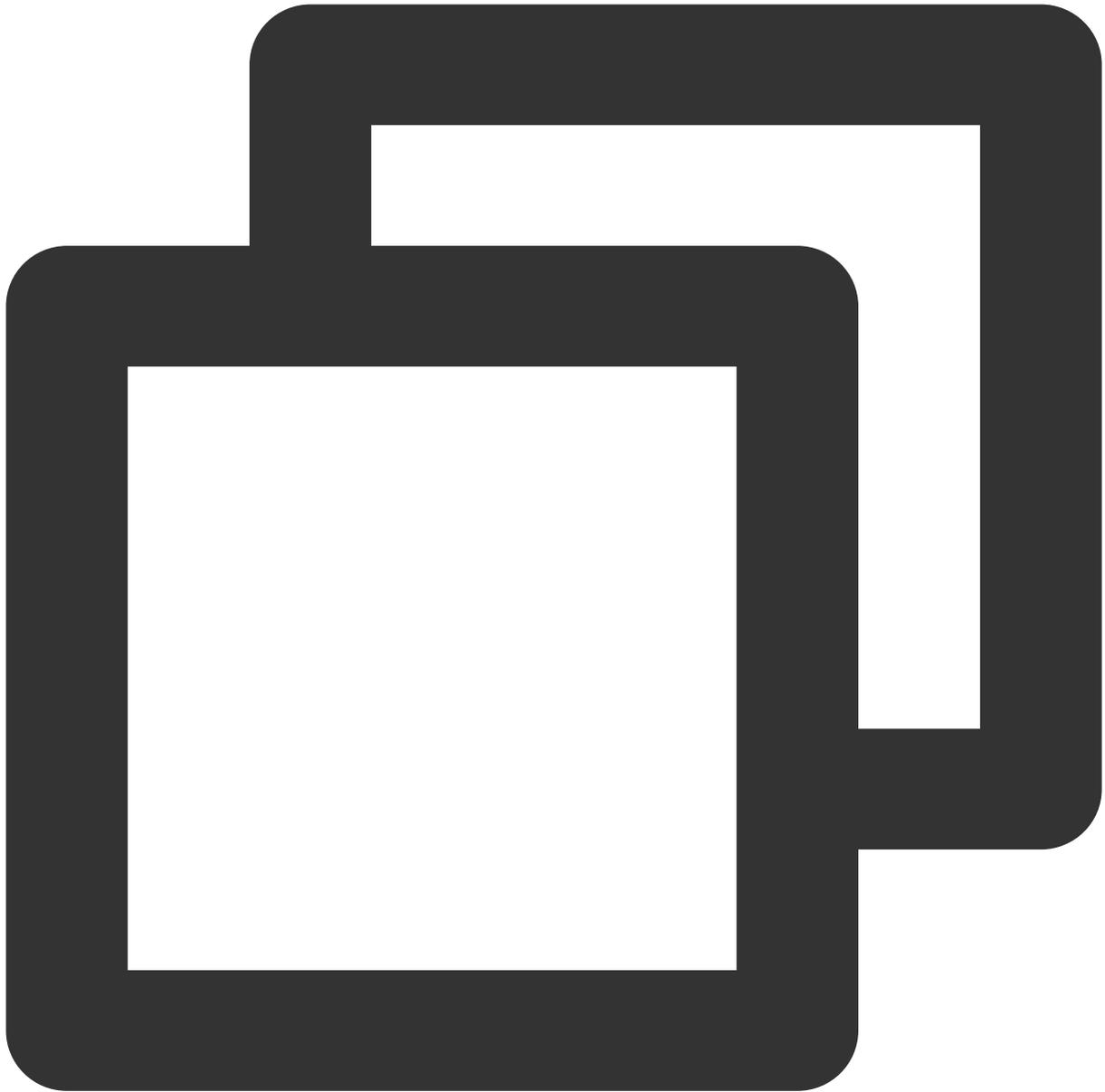


```
npm install tcic-electron-sdk@latest
```

说明：

TCIC Electron SDK 最新版可在 [tcic-electron-sdk](#) 中查看

3. 在项目脚本里引入模块后，调用初始化接口并传入之前获取的参数调起课中页面。



```
const TCIC = require('tcic-electron-sdk')

TCIC.initialize({
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhhdCI6MTY3O
})
```

高级功能

流水线打包接入

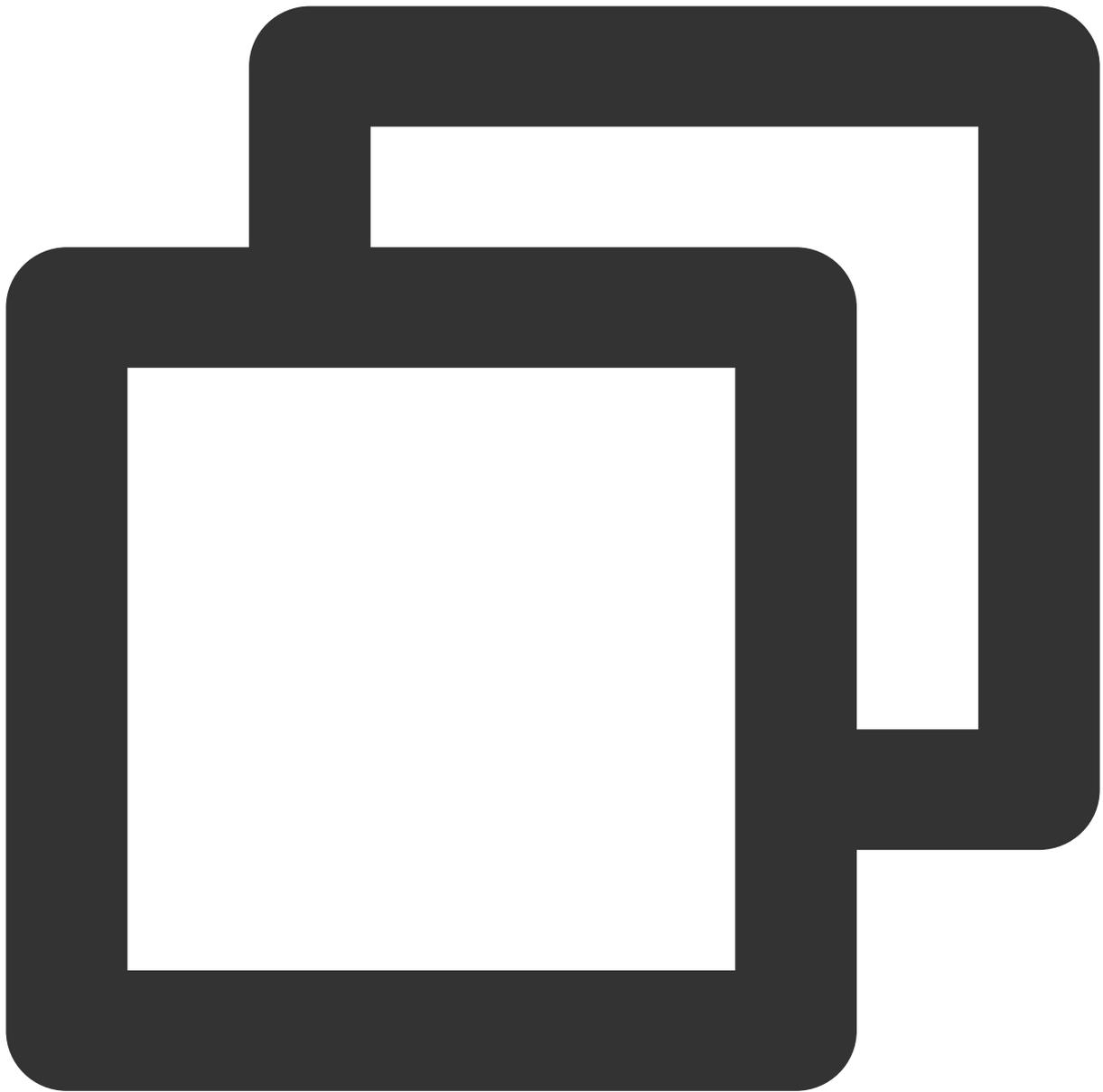
若您希望生成的客户端拥有自己的品牌 logo 和应用名的同时，能快速集成现有业务系统，可以通过使用我们提供的流水线打包方式进行集成。

具体流程参见下图：



1. 处理业务逻辑

在打包后，业务侧可通过 `window` 全局变量上的 `joinClass` 方法进入课堂，若之前业务侧已通过 URL 或 SDK 集成方式实现进课逻辑，可参考以下示例进行兼容。

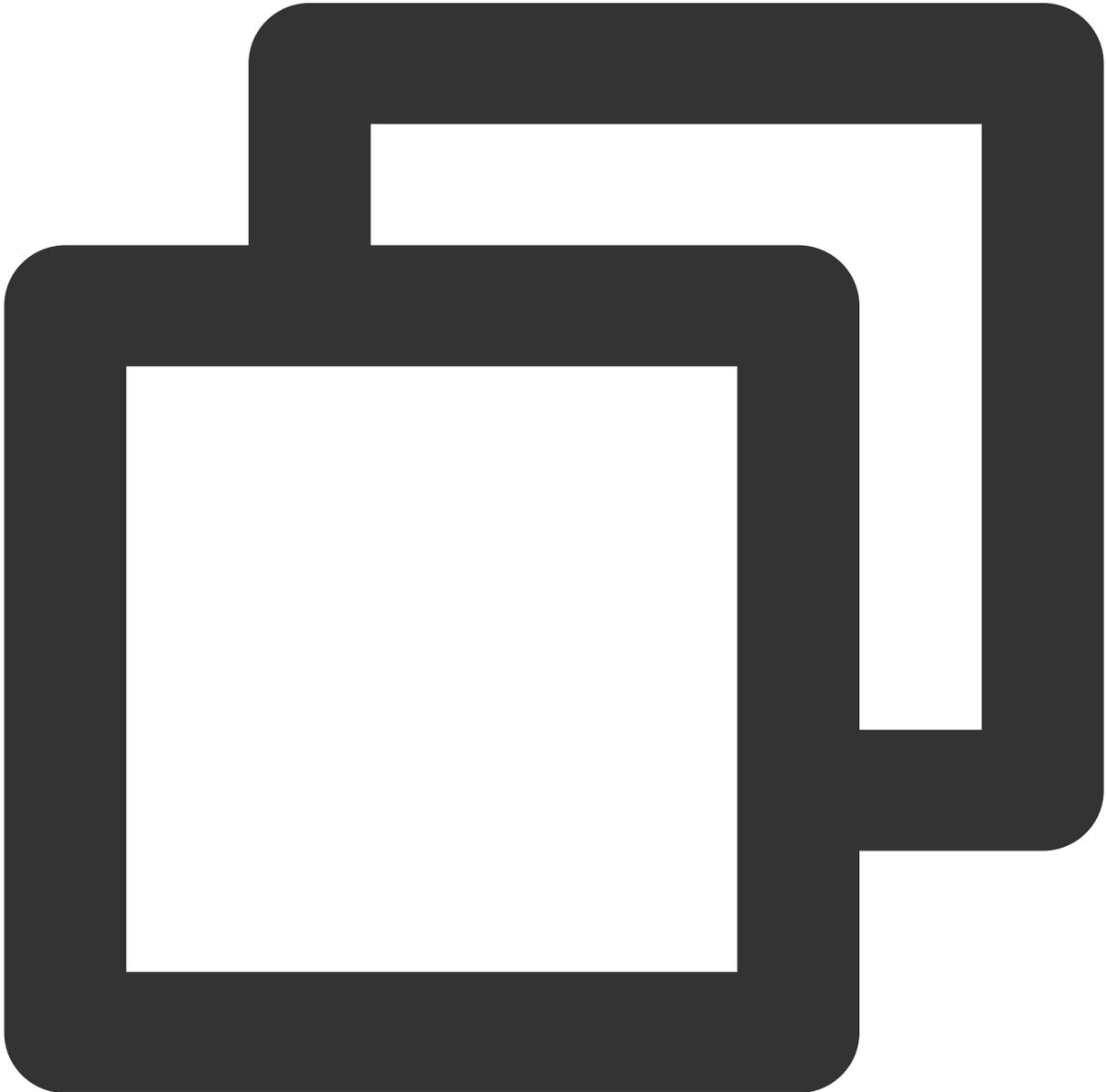


```
const options = {
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsIm1hdCI6MTY3O
};
// 点击进入课堂的逻辑方法
window.joinClass(options);

// 如何同时兼容原有逻辑呢？请参考下面的代码逻辑
```

```
// 判断是否存在这个方法
if (window.joinClass) {
    window.joinClass(options);
} else {
    // 原有的进入课堂逻辑
}
```

业务侧需通过 `window` 全局变量上的 `closeWin` 关闭当前的客户端。



```
window.closeWin();
```

2. 准备对应的物料清单，具体如下所示：

字段	含义	必填
AppName	App 名称，如XX课堂	是
Logo	应用 Logo，请提供规格为256x256的ico/png 图片	是
URL	业务 URL	是

注意：

业务 URL 应是已经具备了登录态的完整 URL，如果没有登录态，能自动跳转至登录入口。配置的业务 URL 不应是登录页，否则用户每次打开都需要重新登录。

3. 发送打包申请邮件，请按照以下格式进行发送，并以附件的形式附带上述物料，信息及物料确认无误的情况下我们将会在一个工作日完成打包。

说明：

收件人：mediaservices@tencent.com

主题

申请客户端打包

内容

公司名称：xxx 有限公司

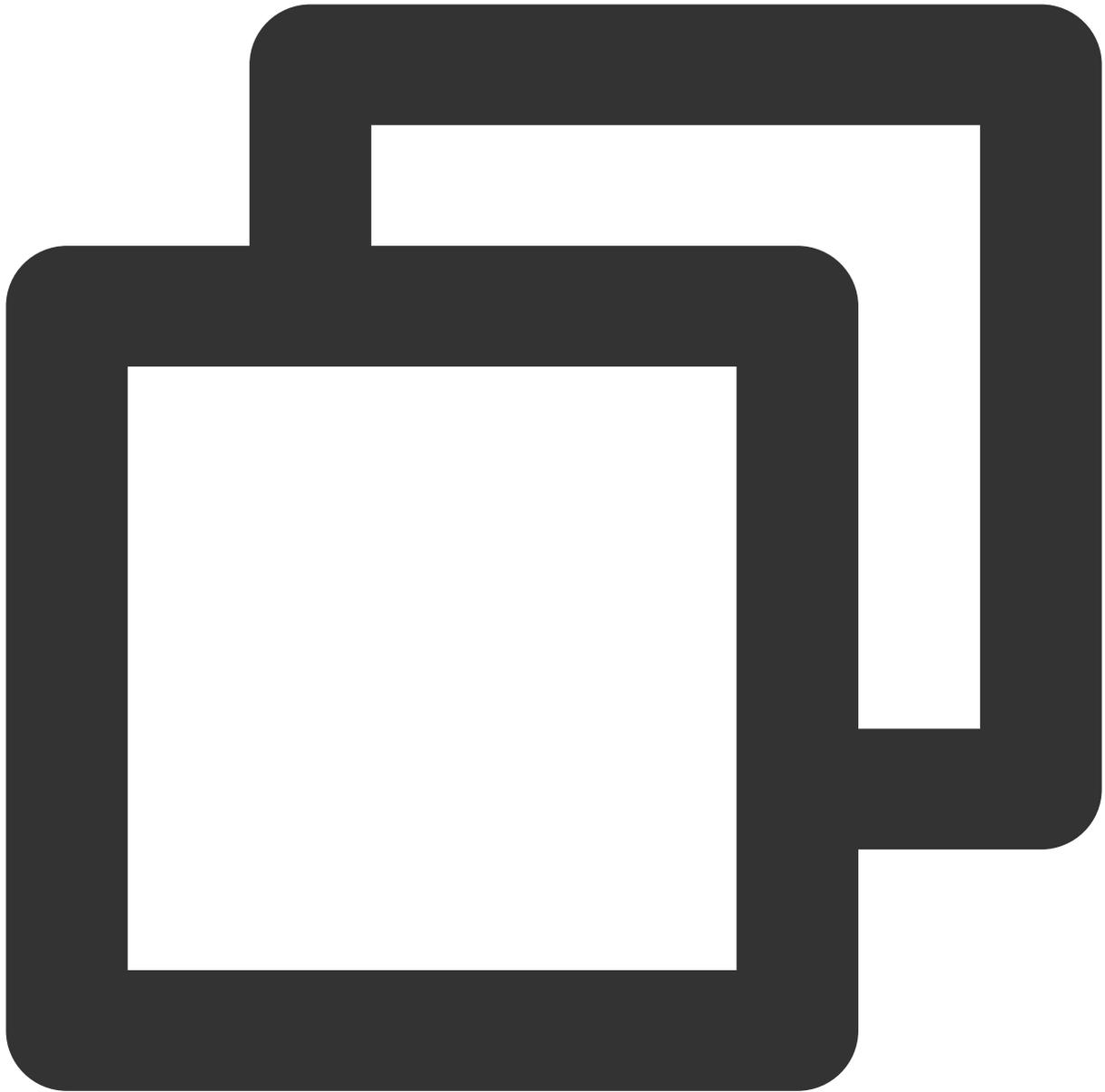
个人姓名：

联系方式：

打包物料（附件）

自定义 UI 集成

为满足不同客户需求，LCIC Electron 目前还提供了自定义 UI 的集成方案。用户可自定义业务侧课中的布局及样式，通过 [界面定制](#) 可以获取到业务侧的 JS 及 CSS 链接，将 `debugjs` 及 `debugcss` 参数拼接到上方的链接上即可（此参数只用于调试），如下代码所示：



```
// URL拼接方式
const url = `tcic://class.qcloudclass.com/latest/class.html?classid=${classId}&user

// SDK集成方式
TCIC.initialize({
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhhdCI6MTY3O
  debugjs: 'http://localhost:443/demo/dist/myLib.umd.min.js',
  debugcss: 'http://localhost:443/demo/dist/myLib.css',
})
```

```
// 流水线打包方式
const options = {
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsIm1hdCI6MTY3O
  debugjs: 'http://localhost:443/demo/dist/myLib.umd.min.js',
  debugcss: 'http://localhost:443/demo/dist/myLib.css',
};
window.joinClass(options);
```

当自定义 JS 与 CSS 调试完成后，可通过云 API 接口 [SetAppCustomContent](#) 或 [控制台 > 应用配置 > 场景配置](#) 将场景与自定义的 JS、CSS 链接进行绑定，在进入课堂时将 `scene` 参数拼接接到 URL 或添加到对应入参上，即可加载对应场景的布局及组件。在涉及多种班型、多种布局时，业务侧可根据此参数实现场景的切换。

自定义业务域名

在课中页面时，若业务侧想隐藏课堂域名，只显示业务域名，可通过 [内容分发网络控制台\(CDN\)](#) 新建业务域名，并回源到课堂域名即可，详细流程请参见 [自定义业务域名](#)。

其他相关文档

[LCIC API](#)

[自定义 UI 集成](#)