

Low-code Interactive Classroom

Client Integration Guide

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Client Integration Guide

Web and H5

Android

iOS

Windows and macOS

Client Integration Guide

Web and H5

Last updated : 2024-06-28 10:04:46

This document mainly describes how to quickly integrate Tencent Cloud LCIC Web/HTML5 Application into your project.

Prerequisites

1. You have completed the procedures for [Sign up for a Tencent Cloud account](#) and [Identity Verification](#).
2. The desktop or mobile browser you are using supports audio and video services. You can refer to [FAQs About Web](#) for detailed requirements.

Step 1: Create New Application

1. Log in to the [Low-code Interactive Classroom](#), and select **Quick Start** from the left navigation bar.
2. By default, you enter the "Create application" interface, where you can select "**Create new application**" for the application type. Enter the application name, such as TestLCIC.

If you have already created an application, you can click "Select an existing application" for the application type.

Note:

Each account can get a trial version application for free. If you need to create a commercial version, you can choose and purchase the corresponding version based on your business needs on the [Purchase Page](#).

3. Add or edit Tag (optional), and then click **Create**.

1 Create application > 2 Generate Class Parameters > 3 Success

Create/Select Application

Type

☒ Create new application ☐ Select an existing application

Application

Enter an application name

0/15

Add tags (optional)

Tags are used to classify and manage resources from different dimensions. If the existing label does not meet your requirements, please go to [Tag r](#)
Once the tags are updated, please [Click here to refresh tab](#)

Tags

Please select a tab key

Please select a tag value

+ add tags

Create

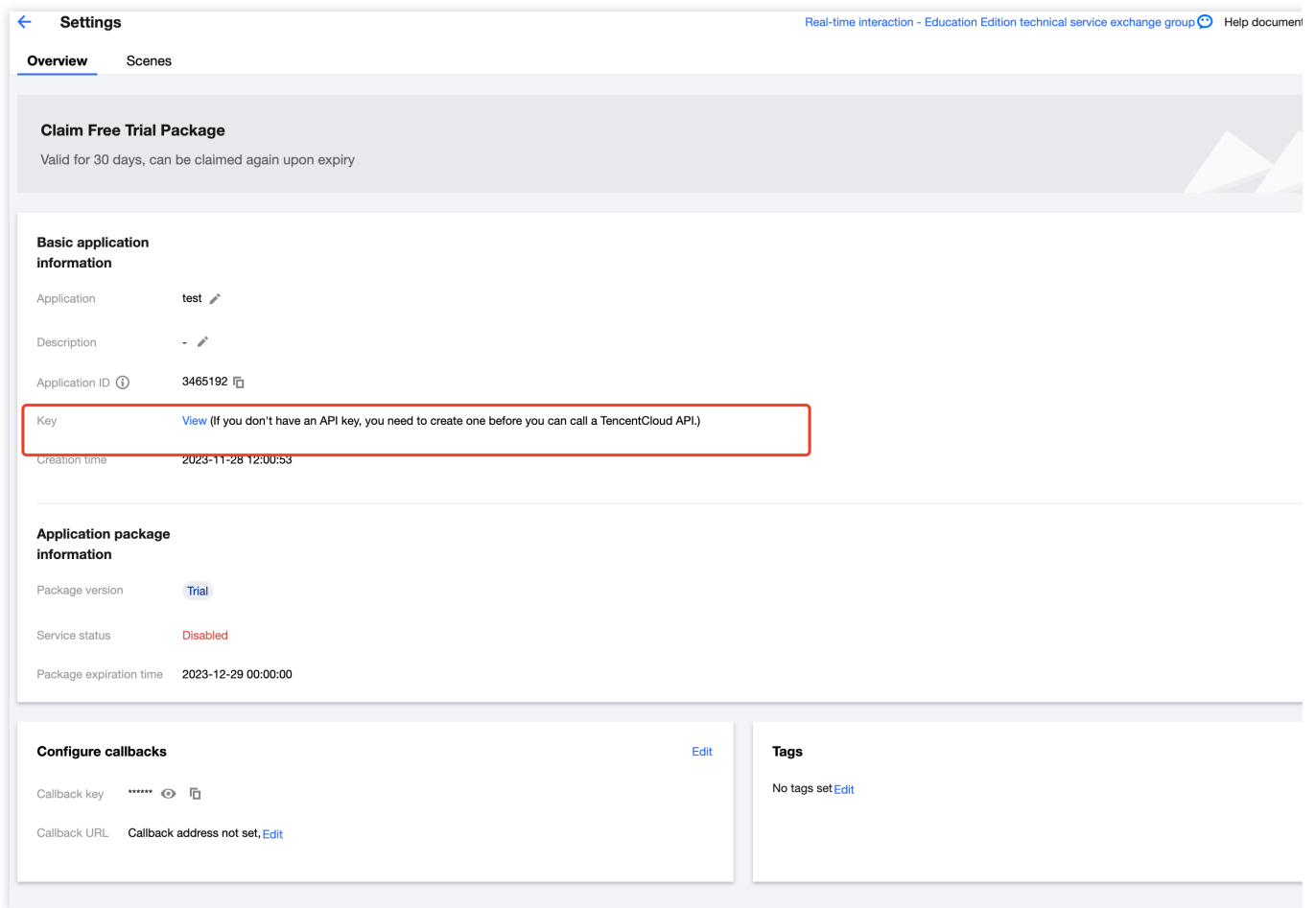
Reset

Note:

Application names may only contain underscores, digits, or alphanumeric characters in both English and Chinese. Tags are used to identify and organize your resources in Tencent Cloud. For instance, a corporation with several business units, each of which has one or more LCIC applications, can use tags to label departmental information on LCIC applications. Tags are optional, and you can add or edit them based on your actual business needs.

Step 2: Enter SDKAppId and Key (SecretKey)

1. Choose [Application Management](#)> **Settings** to obtain the SDKAppId.
2. Enter [Cloud Access Management](#) to obtain the key. If there is no key, you need to create a new one in **API Key Management**. For details, refer to [Key Management](#).



Step 3: Obtain Parameters Required to Enter the Classroom

1. By calling the TencentCloud API interface [RegisterUser](#) to register a user, you can obtain the corresponding user ID (`userid`) information.
2. By calling the TencentCloud API interface [LoginUser](#) to log in, you can obtain the user authentication `token` information.
3. By using the Application Programming Interface [CreateRoom](#) to create a classroom, you can obtain the Classroom ID (`classid`) information.
4. Specify the version of the classroom you need to integrate: latest.
5. The `scene` , `debugjs` , and `debugcss` are optional parameters, which are only needed when defining the UI. See [Custom UI Integration](#) for details. `debugjs` and `debugcss` are used only for the debugging of customized layouts and components, and only accessed via `localhost` or `127.0.0.1` . Do not use these parameters in the release stage.
6. `lng` , `location` , and `layout` are also optional parameters. You can determine whether to enter these parameters based on actual business needs. If not, they will use the default values. The `layout` parameter is only effective when the classroom layout is set to video plus document layout (`videodoc`).

Field	Type	Meaning	Remarks	Mandatory	
-------	------	---------	---------	-----------	--

userid	String	Username	It is obtained through the RegisterUser interface.	Yes	
classid	String	Classroom ID	It is obtained through the return of creating the CreateRoom interface.	Yes	
token	String	Backend authentication parameters	It is obtained through the LoginUser interface.	Yes	
version	String	Version number of the classroom	The corresponding version can be selected from the release logs. Note: For the interactive classroom client version, 'latest' is recommended.	Yes	
scene	String	Scene name	It is used to distinguish different custom layouts. It is configured through the SetAppCustomContent interface, and set to <code>default</code> by default.	No	
debugjs	String	JS link of the custom defined UI	It is obtained through the custom defined UI integration.	No	
debugcss	String	CSS link of the custom defined UI	It is obtained through the custom defined UI integration.	No	
role	String	Role for entering the classroom, which is empty by default	Optional parameter supervisor (patrolling/content review). Only registered users in the application have the authority.	No	
lng	String	Language parameter, which is zh-CN by default	Currently zh-CN and en-US are supported.	No	
location	Boolean	Whether to report geographical location information	It is set to false by default, indicating no reporting.	No	
layout	String	Page layout	It is set to top layout (top) by default. Currently, this parameter is only effective for video document pattern. It supports double row layout (double), right layout (right), left layout (left), and three-part layout (three)	No	

boardColor	String	Whiteboard color	It is used to set the whiteboard color. It is set to #182E25 by default. It supports Hex format and rgba(0, 0, 0, .3) setting.	No	
back_url	String	Exit Classroom Callback Address	It is empty by default. It is the page address that needs to be bounced back when clicking return or exiting the classroom. The parameter needs to be encoded using encodeURIComponent	No	

Step 4: Enter the Classroom

Standard Features

Connect the recently obtained parameters to form the **URL** for accessing the classroom according to the following standard. You can achieve web integration after being redirected to this URL.

[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})

The specific link is as follows:



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

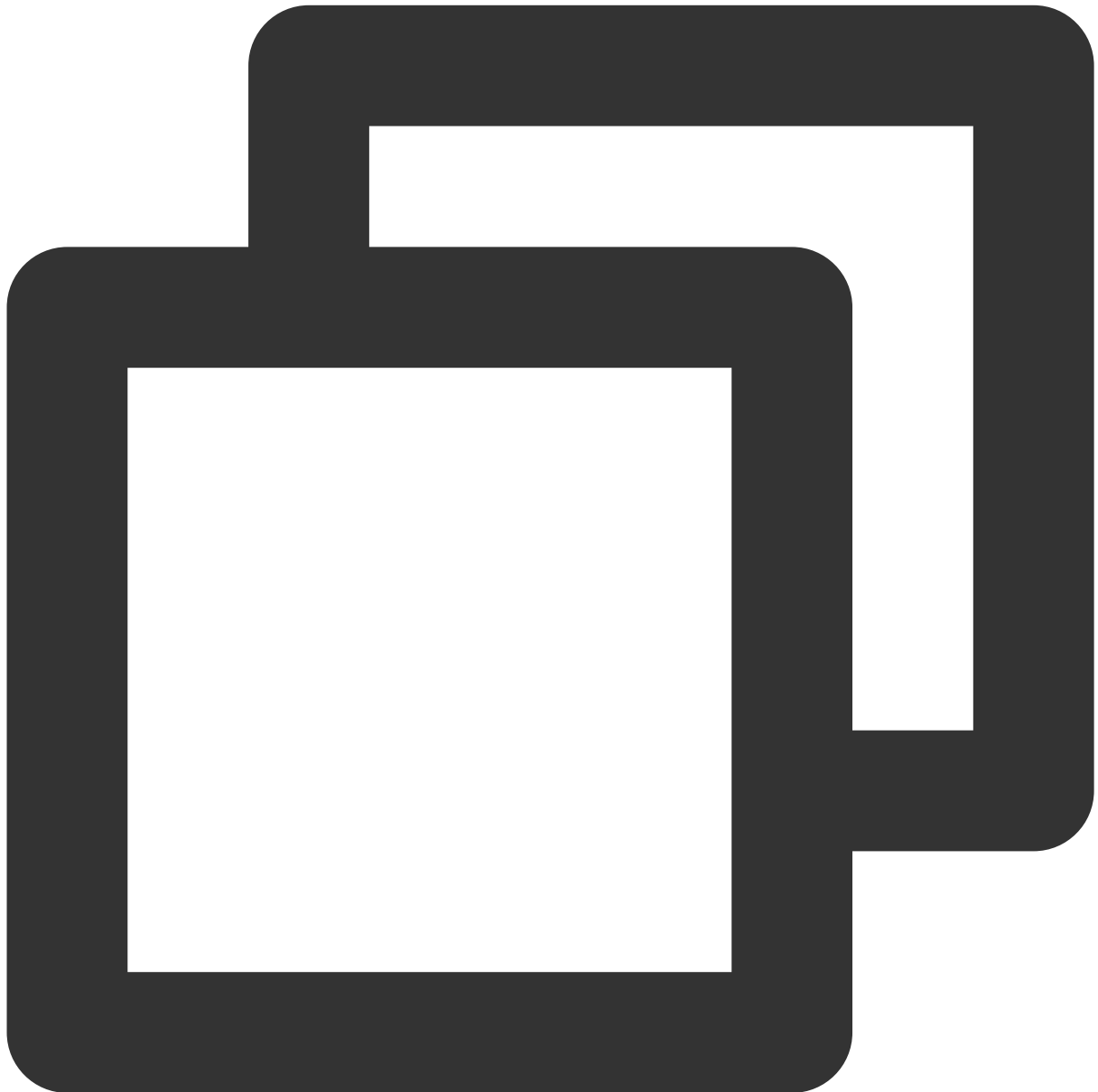
Note:

If the `userid` in the appended URL matches the teacher's ID (`teacherid`) specified in the [create classroom](#), the current user is a teacher. If it matches the current classroom's assistant's ID (`assistantid`), the user is an assistant. Otherwise, the user is a student.

Advanced Features

Custom UI Integration

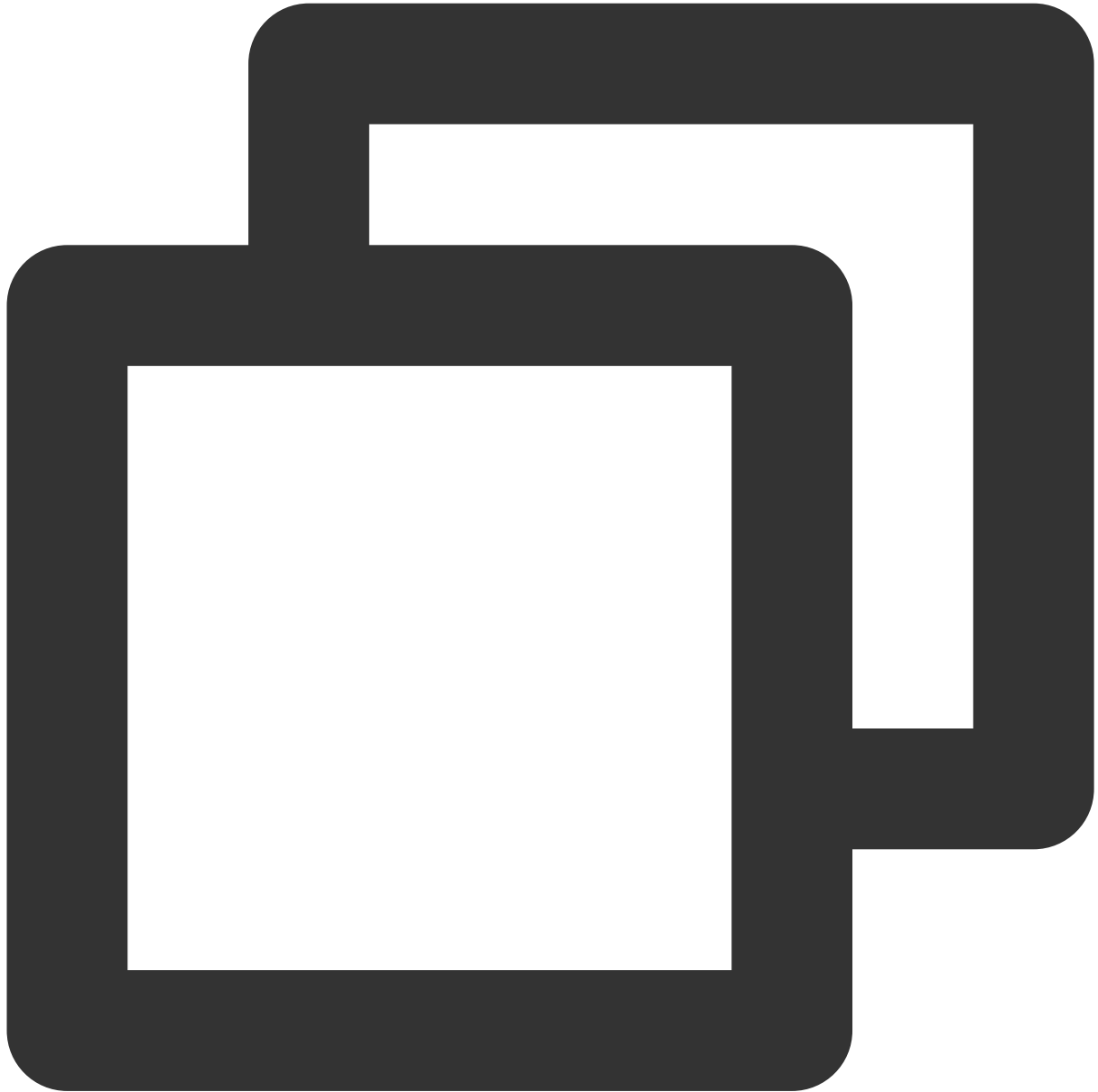
LCIC Web/H5 also provides an integration solution for customizing the UI. Users can customize the layout and style on the business side. By [customizing the interface](#), you can access the JS and CSS links on the business side. You can add the parameters `debugjs` and `debugcss` to the link above (these parameters are only used for debugging) as shown below:



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

When your custom JS and CSS have been debugged, you can use the [SetAppCustomContentAPI](#) or the [Low-code Interactive Classroom > ApplicationManagement > Settings > Scenes](#) to bind the scene, and the links to the

custom JS and CSS. When accessing the classroom, append the `scene` parameter to the URL to load the layout and components of the corresponding scene. When dealing with various types of classes and various layouts, you can switch scenes with this parameter on the business side, as shown below:



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

Custom Business Domain

On the course page, if you need to hide the classroom domain and only display the business domain on the business side, you can create a new business domain through the [CDN \(CDN\) Console](#) and backsource to the classroom

domain. For the detailed process, refer to [Custom Business Domain](#).

Other Related Documents

[LCIC API](#)

[Customize Interface](#)

[Web related problems](#)

Android

Last updated : 2024-08-16 14:44:50

Development Environment Requirements

Android studio 3.0+

Android 4.4 (API 19) and later

Prerequisites

[Sign up for a Tencent Cloud account](#) and complete [Identity Verification](#).

Step 1: Creating a New Application

1. Log in to the [LCIC console](#), and select **Quick Run** in the left navigation bar.
 2. You will enter the **application creation** page. Then check **Create new application** for the application type, and enter an application name, such as `TestLCIC`.
- If you have already created an application, you can also check "Select an existing application" for the application type.

Note :

To Integrate the Mobile client, you need to purchase the Premium or Enterprise edition. If a commercial application is needed, you can create a corresponding version of application according to business requirements on the [purchase page](#).

3. Based on actual business requirements, add or edit tags, and then click **Create**.

1 Create application > **2 Generate Class Parameters** > **3 Success**

Create/Select Application

Type ☒ Create new application ☐ Select an existing application

Application 0/15

Add tags (optional)

Tags are used to classify and manage resources from different dimensions. If the existing label does not meet your requirements, please go to [Ta](#). Once the tags are updated, please [Click here to refresh tab](#).

Tags

+ add tags

[Create](#) [Reset](#)

Note :

Application names only support underscores, digits, or Chinese and English characters.

Tags are used to identify and organize various resources you possess within Tencent Cloud. For instance, an enterprise may include several business divisions, and each division may have 1 or more LCIC applications. In such cases, the enterprise can identify departmental information by assigning tags to the LCIC applications. Tags are not required. You can add or edit them based on actual business requirements.

Step 2: Acquiring the `SDKAppId` and the Key (`SecretKey`)

1. Navigate to [Application Management](#) > **Settings** to acquire the `SDKAppId` .
2. Log in to the [CAM console](#) to acquire the key. In case of no key, you must create a key in **API Key Management**. For details, refer to [Key Management](#).

Settings Real-time interaction - Education Edition technical service exchange group [Help docun](#)

Overview **Scenes**

Claim Free Trial Package
Valid for 30 days, can be claimed again upon expiry

Basic application information

Application	test ✎
Description	- ✎
Application ID ①	3465192 🔗
Key	View (If you don't have an API key, you need to create one before you can call a TencentCloud API.)
Creation time	2023-11-28 12:00:53

Application package information

Package version	Trial
Service status	Disabled
Package expiration time	2023-12-29 00:00:00

Configure callbacks [Edit](#)

Callback key	***** 👁 🔗
Callback URL	Callback address not set, Edit

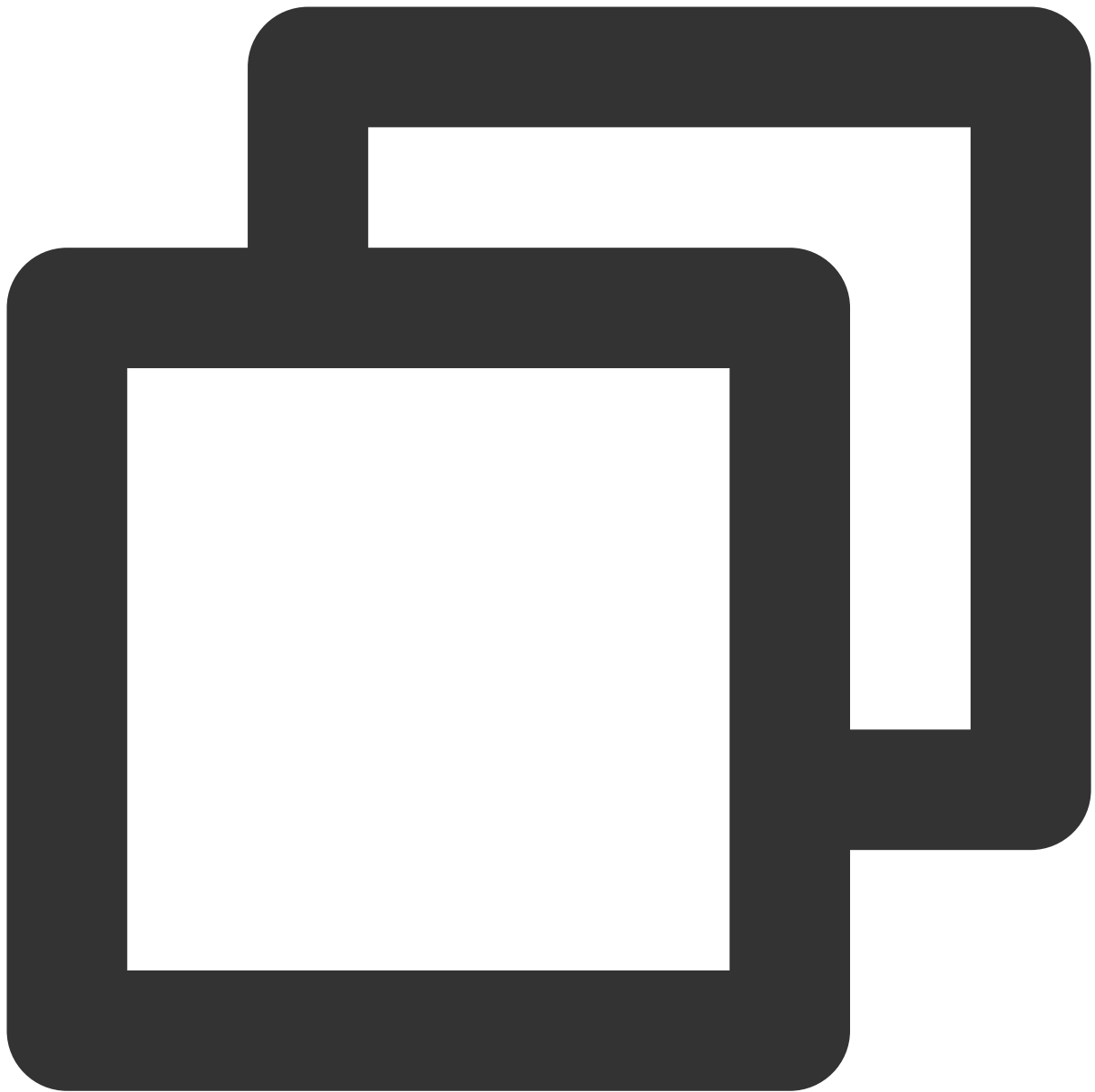
Tags
No tags set [Edit](#)

Step 3: Importing the SDK

Remote SDK Building for a Gradle Project

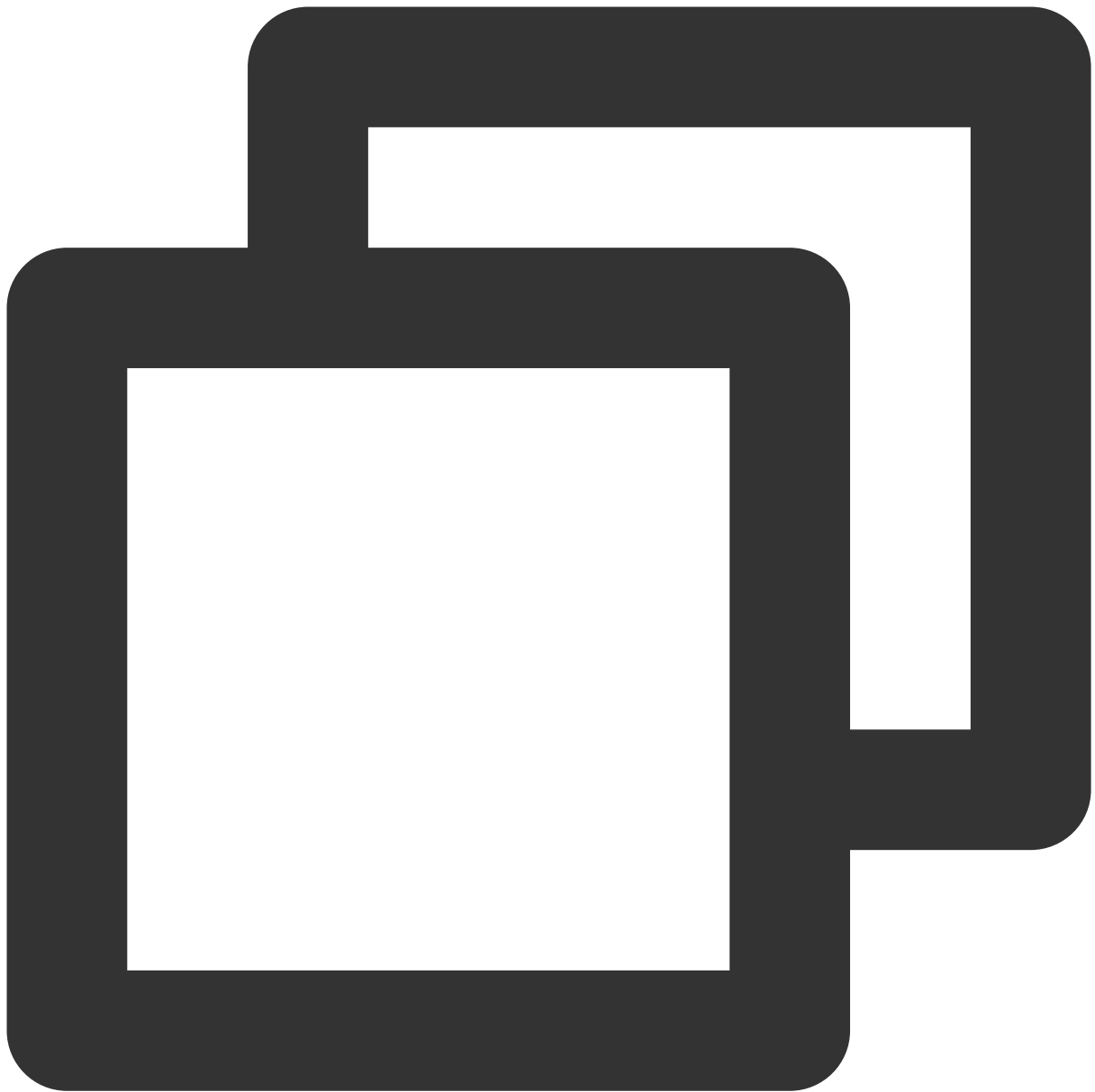
The LCIC SDK has been released to the Maven central repository. You can enable automatic downloading of updates through Gradle configuration.

1. In `build.gradle` of the application module, add the LCIC SDK dependency to dependencies.



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

2. In `defaultConfig` , specify the CPU architecture utilized by the Application.



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

3. In `compileOptions` , specify JDK 1.8 for compilation.



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

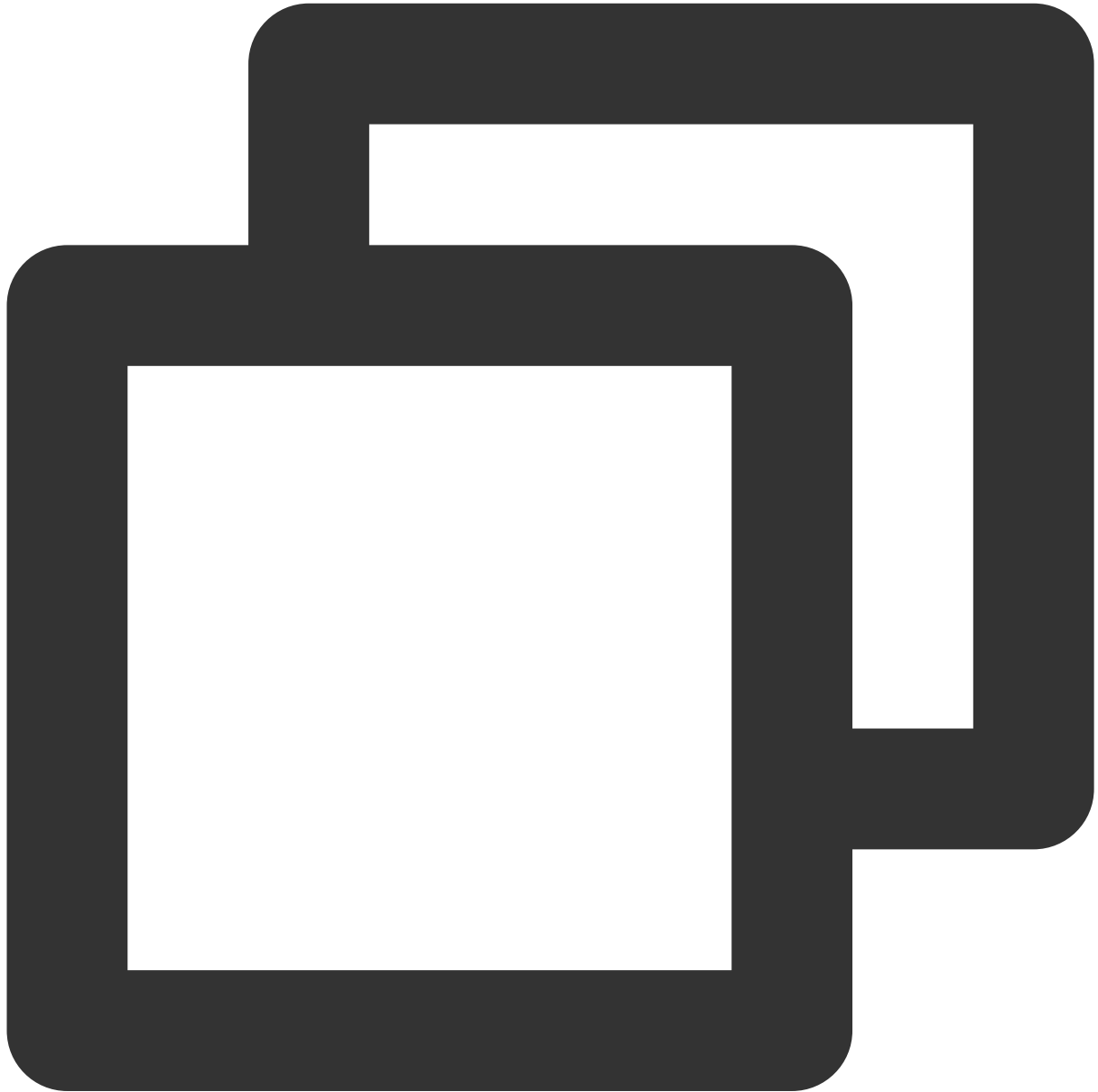
Note :

The current LCIC SDK supports both armeabi-v7a and arm64-v8a architectures.

4. Click **Sync Now**, to automatically download the SDK and integrate it into the project.

Step 4: Configuring Application Permissions

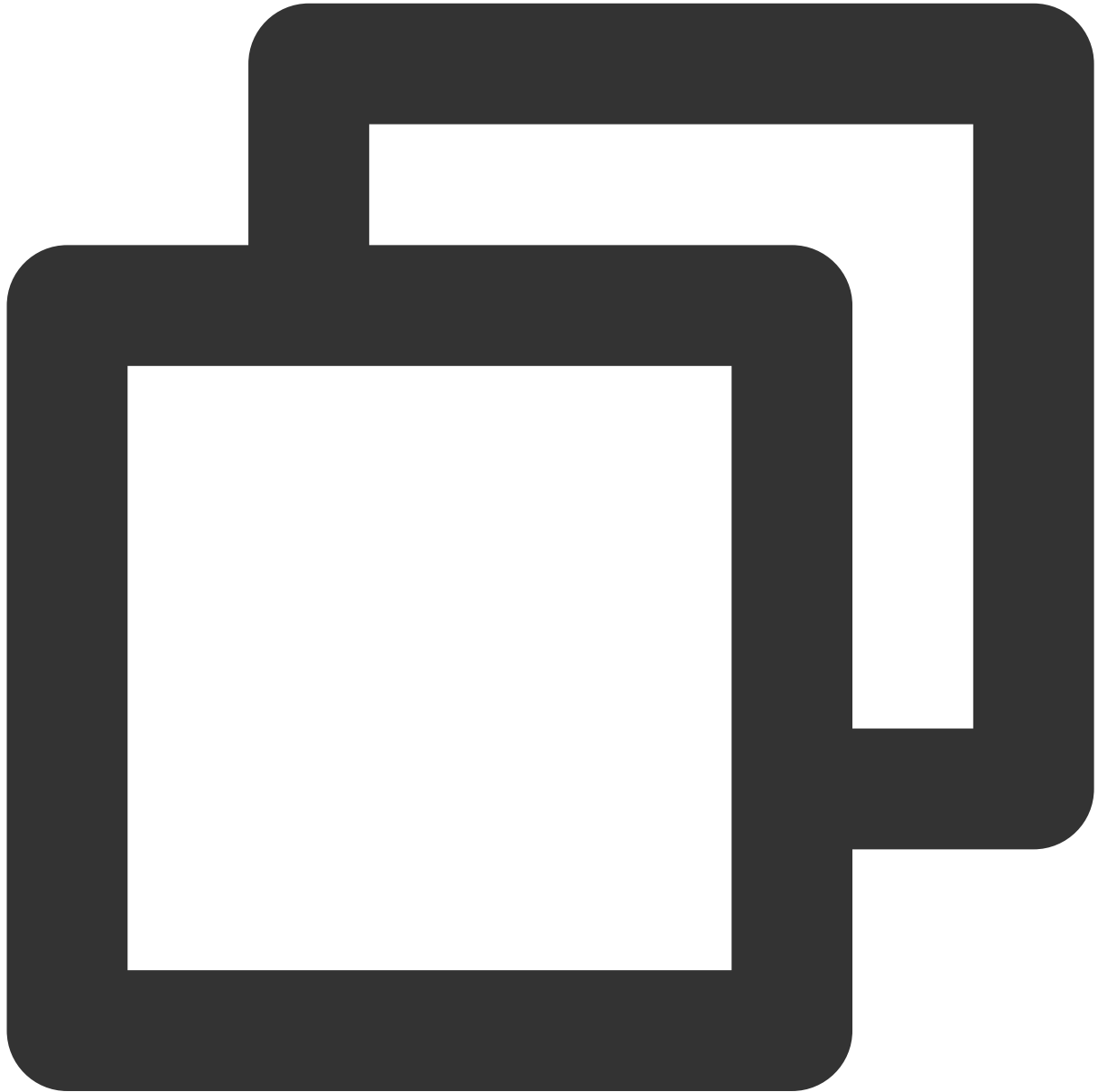
In the `AndroidManifest.xml`, configure the application permissions. The LCIC SDK requires the following permissions:



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

Step 5: Configuring Obfuscation Rules (Optional)

If obfuscation is enabled, you must add the LCIC SDK-related classes to the non-obfuscation list in the `proguard-rules.pro` file.



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

Step 6: Applying for a License of the SDK

To apply for the SDK license, kindly send an email according to the template below to mediaservices@tencent.com to contact us. Upon confirmation we will issue the license within 1 business day.

Note :

The package name is used for signature authorization in the X5 kernel and the Live Event Broadcasting player. Send an email to provide the information such as App Name, Package Name, and Bundle ID of the official application requiring a license.

Recipient: mediaservices@tencent.com

Subject:

LCIC Android SDK License Application

Content:

Company Name: xxx Ltd.

Personal Name:

Contact Information:

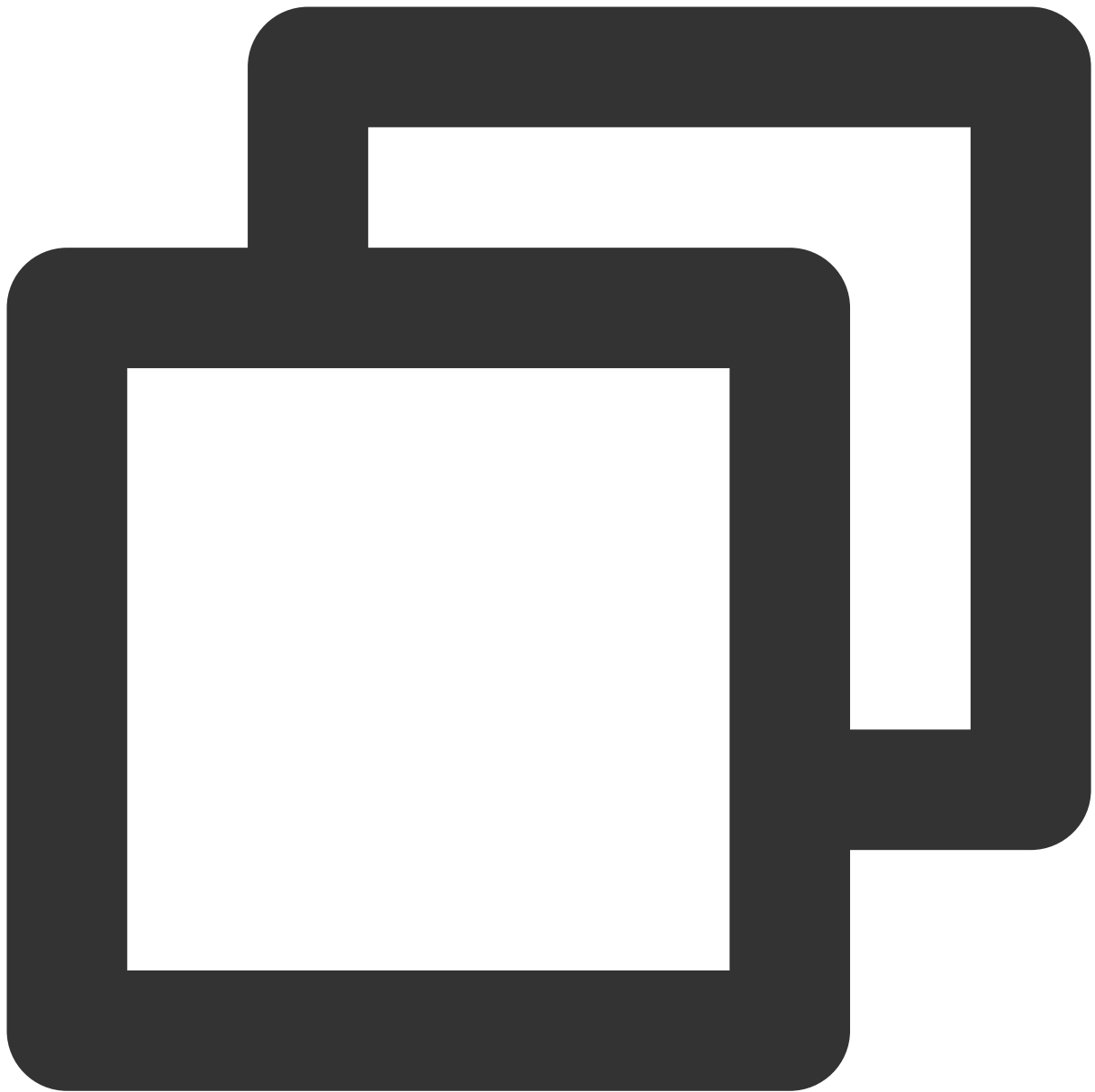
App Name:

Package Name (Android):

Bundle ID (iOS):

Step 7: Initializing the X5 Kernel

Compared to the system WebView, the X5 kernel boasts superior compatibility and enhanced speed. The implementation of the LCIC Android SDK depends on the X5-based WebView. A method for static integration of the X5 kernel is now provided to increase the success rate of loading the X5 kernel, and it can take effect without the need to restart the process.



```
//Initializing the X5 Kernel
TCICManager.getInstance().initX5Core(licenseKey, new TBSSdkManageCallback() {

@Override
public void onCoreInitFinished() {
}

@Override
public void onViewInitFinished(boolean isX5Core) {
    if (isX5Core) {
        //The X5 Kernel initialization completed, you can enter the class.
    }
}
```

```

    }
}
});

```

Note :

You must acquire the X5 kernel's `licenseKey` , namely the `licenseKey` parameter in

`TCICManager.getInstance().initX5Core(licenseKey);` by sending an email to contact us through [Step 6](#).

You should agree to the privacy policy before calling the method for initializing the X5 kernel, to avoid the collection of personal information without user consent when the application is launched in the app market.

Before entering the classroom, it is recommended to verify whether the X5 kernel has been successfully installed. The verification method is `isX5InstallSuccess` .

Step 8: Acquiring the Parameters Necessary for Classroom Entry

TCICClassConfig Parameter Explanation:

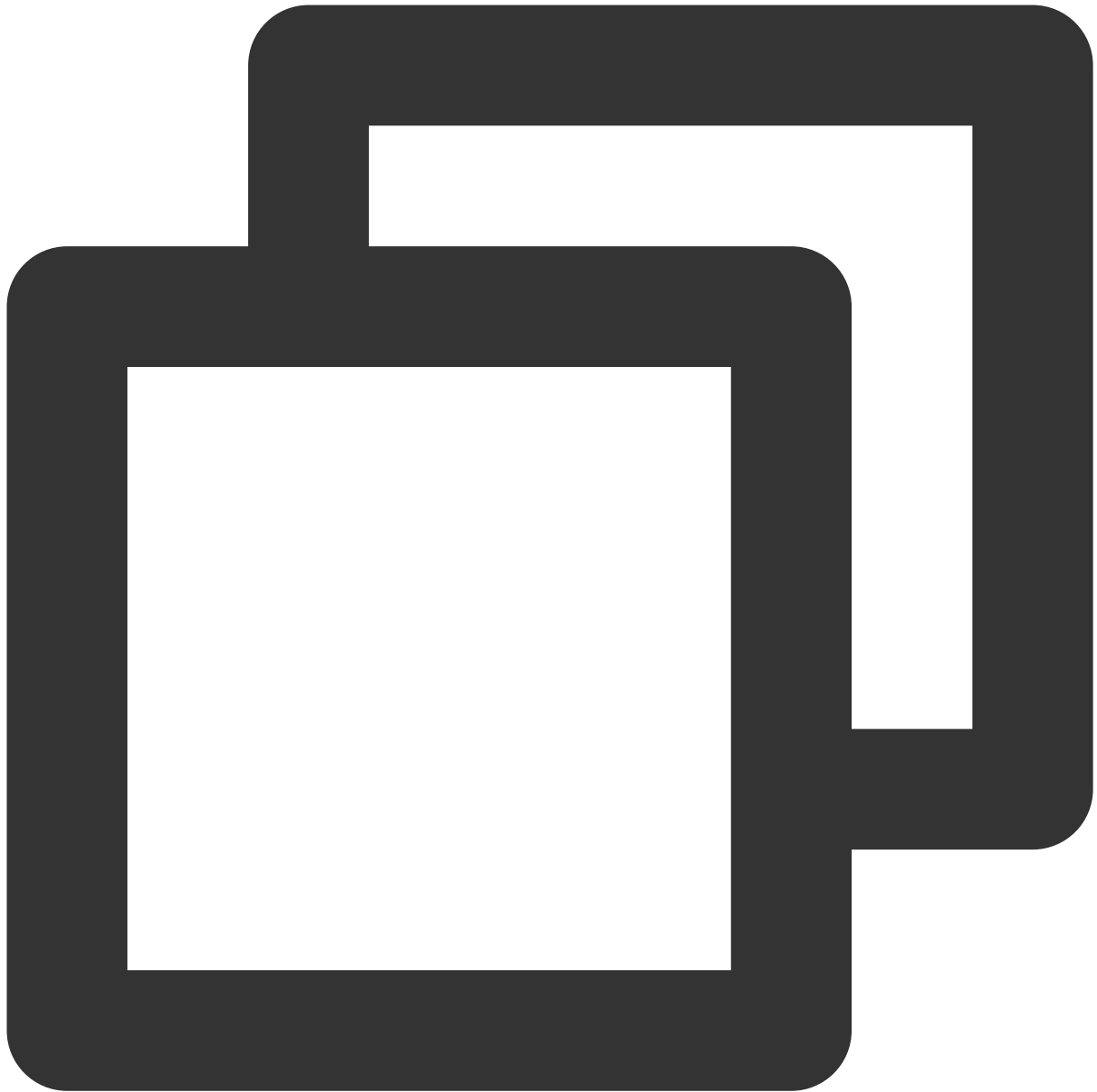
1. Navigate to **Application Management > Settings** through the [Console](#) to acquire the [SDKAppld](#), namely the school ID (`schoolId`).
2. Create a classroom through the TencentCloud API [CreateRoom](#), to obtain the classroom ID (`classId`).
3. Register a user by calling the TencentCloud API [RegisterUser](#), to acquire the corresponding user ID (`userId`).
4. Log in through the TencentCloud API [LoginUser](#), to acquire the user authentication token.
5. Parameters such as `scene` , `lng` , `camera` , `mic` , and `speaker` are not required. if they are not specified, the default values are used.

Field	Type	Description	Remarks
schoolId	Integer	School ID	Navigate to Application Management > Settings through the console, acquire the SDKAppld .
classId	Long	Classroom ID	RoomId returned by creating a classroom through the CreateRoom API.
userId	String	User account	Acquired through the RegisterUser API.
token	String	Background authentication parameter	Acquired through the LoginUser API.
scene	String	Scenario name	Used to distinguish different custom layouts. It is configured through the SetAppCustomContent API.
lng	String	Language parameter	Currently supports zh and en, zh by default. Additionally, you must set the <code>lng</code> parameter in

			<code>TCICWebViewManager.getInstance().setClassLanguage(th env, lng);</code>
camera	Integer	Initializing and turning on the camera	1 indicates turning on the camera, and 0 indicates turning off the camera The default is 1.
mic	Integer	Initializing and turning on the microphone	1 indicates turning on the microphone, and 0 indicates turning off the microphone. The default is 1.
speaker	Integer	Initializing and turning on the speaker	1 indicates turning on the speaker, and 0 indicates turning off the speake The default is 1.

Step 9: Initiating the Main Page of the Component

The main page of the LCIC component can be initiated by merely 4 parameters, respectively school ID, classroom ID, user account, and token.

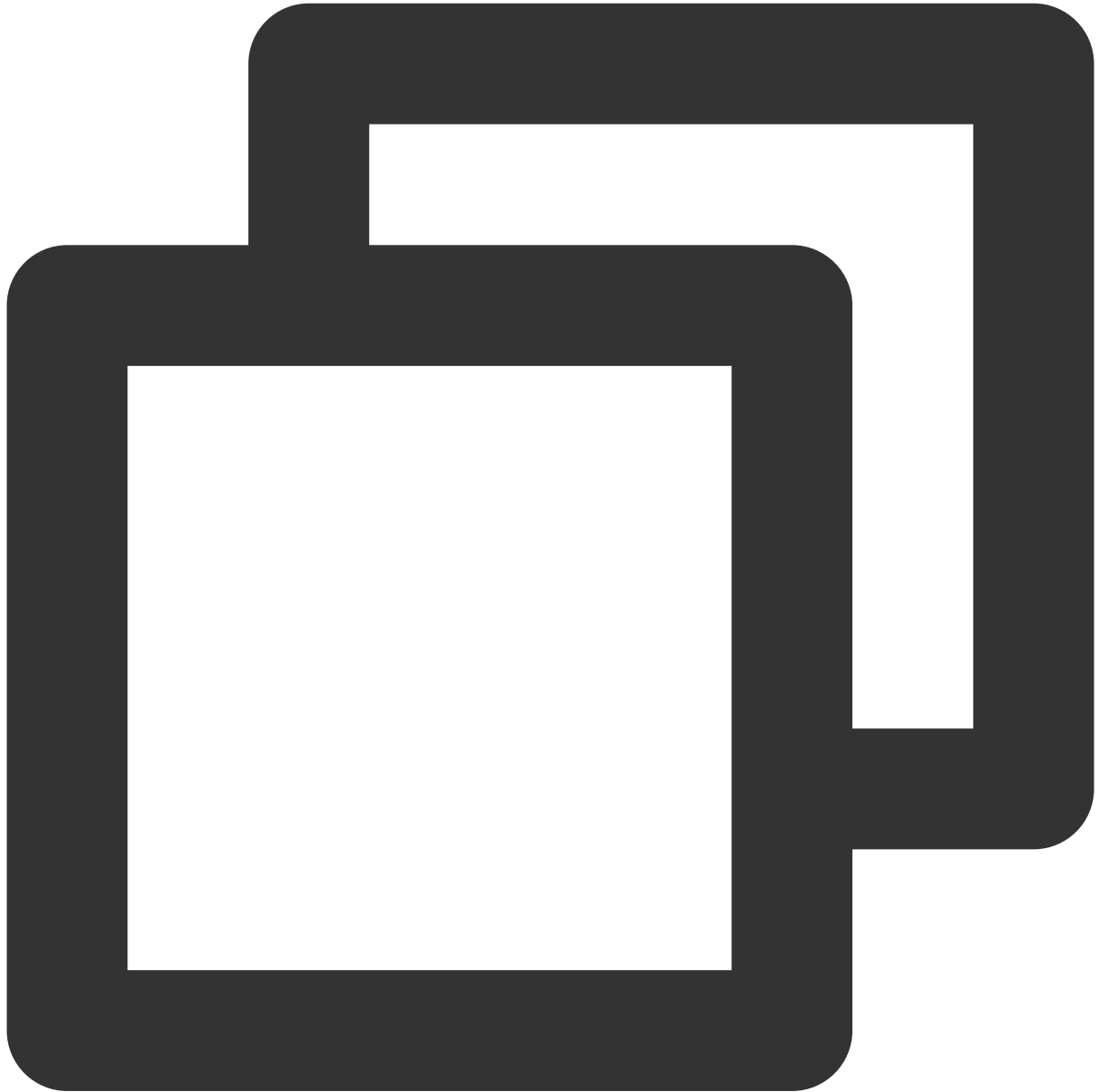


```
Intent intent = new Intent(getActivity(), TCICClassActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_SINGLE_TOP);
Bundle bundle = new Bundle();
TCICClassConfig initConfig = new TCICClassConfig.Builder()
    .schoolId(schoolId)
    .classId(classId)
    .userId(userId)
    .token(token)
    .build();
bundle.putParcelable(TCICConstants.KEY_INIT_CONFIG, initConfig);
intent.putExtras(bundle);
```

```
startActivity(intent);
```

If you want to receive a notice on exiting the classroom, you may register for the local broadcast of

`TCICConstants.ON_CLASS_EXITED_ACTION` , as shown in the following code snippet:



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

Other Recommendations

We suggest you to integrate Tencent Bugly when using LCICSDK, so that you can quickly identify and solve exceptions, keep abreast of product operation trends, and promptly address user feedback. For the integration guidance, refer to the [Tencent Bugly Official Website](#).

For further reference, consult the [Development Demo](#).

iOS

Last updated : 2024-07-26 17:35:51

Environment Requirements

Xcode 14

Prerequisites

You have completed the procedures for [Sign up for a Tencent Cloud account](#) and [Identity Verification](#).

Step 1: Create New Application

1. Log in to the [Low-code Interactive Classroom](#), and select **Quick Start** from the left navigation bar.
2. By default, you enter the "Create application" interface, where you can select **"Create new application"** for the application type. Enter the application name, such as TestLCIC.
If you have already created an application, you can click "Select an existing application" for the application type.

Note:

To Integrate the Mobile client, you need to purchase the Premium or Enterprise edition. If you need to create a commercial version, you can choose and purchase the corresponding version based on your business needs on the [Purchase Page](#) according to business needs.

3. Add or edit tags based on actual business needs, and then click **Create**.

1 Create application > **2 Generate Class Parameters** > **3 Success**

Create/Select Application

Type ☒ Create new application ☐ Select an existing application

Application 0/15

Add tags (optional)

Tags are used to classify and manage resources from different dimensions. If the existing label does not meet your requirements, please go to [Tags](#). Once the tags are updated, please [Click here to refresh tab](#).

Tags

+ add tags

Note:

Application names may only contain underscores, digits, or letters.

Tags are used to identify and organize your resources in Tencent Cloud. For instance, a corporation with several business units, each of which has one or more LCIC applications, can use tags to label departmental information on LCIC applications. Tags are optional, and you can add or edit them based on your actual business needs.

Step 2: Enter SDKAppId and Key (SecretKey)

1. Choose [Application Management](#) > **Settings** to obtain the SDKAppId.
2. Enter [Cloud Access Management](#) to obtain the key. If there is no key, you need to create a new one in **API Key Management**. For details, refer to [Key Management](#).

Settings Real-time interaction - Education Edition technical service exchange group [Help docu](#)

Overview **Scenes**

Claim Free Trial Package
Valid for 30 days, can be claimed again upon expiry

Basic application information

Application	test ✎
Description	- ✎
Application ID ①	3465192 📄
Key	View (If you don't have an API key, you need to create one before you can call a TencentCloud API.)
Creation time	2023-11-28 12:00:53

Application package information

Package version	Trial
Service status	Disabled
Package expiration time	2023-12-29 00:00:00

Configure callbacks [Edit](#)

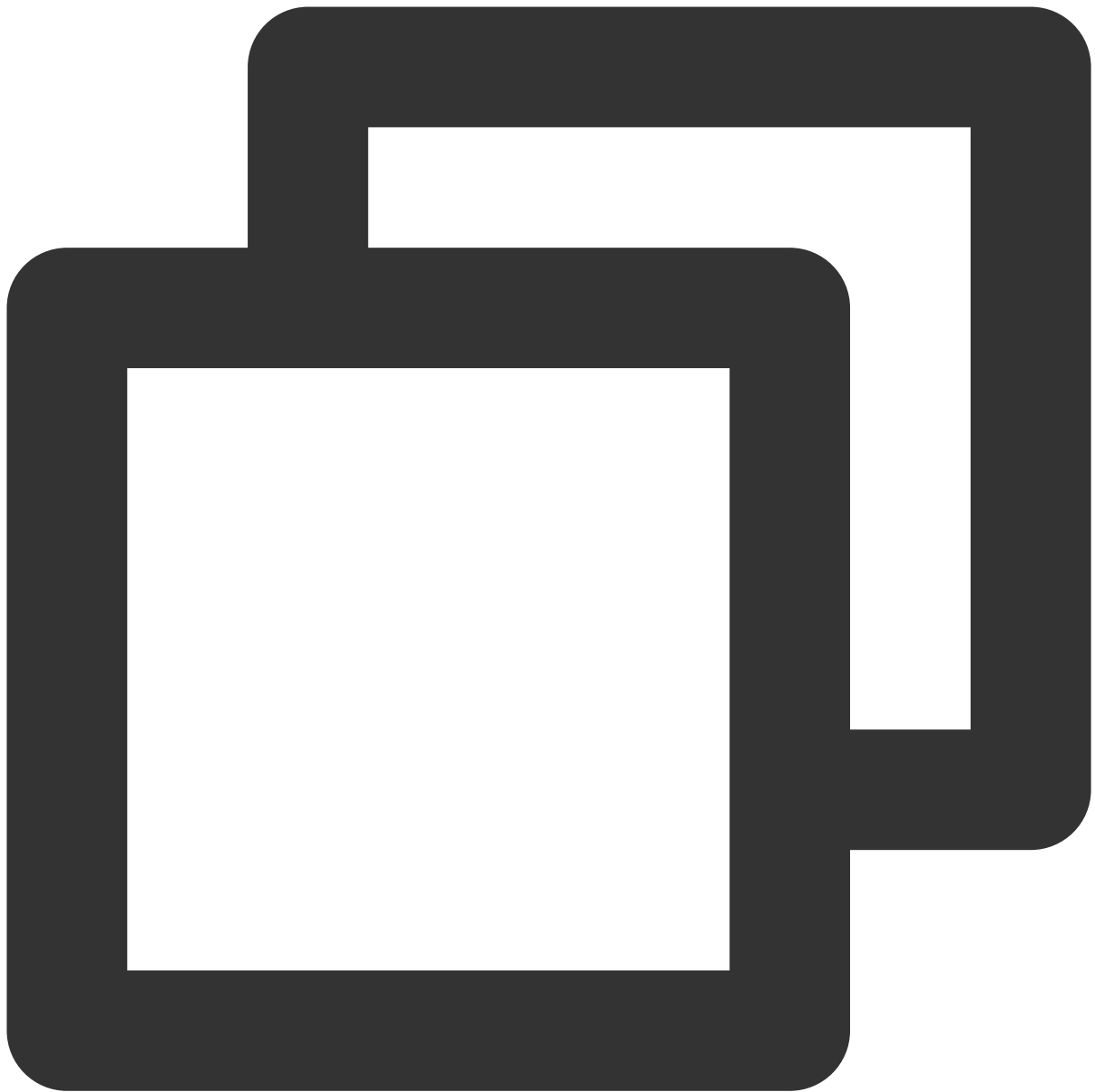
Callback key	***** 👁 📄
Callback URL	Callback address not set, Edit

Tags
No tags set [Edit](#)

Step 3: Import the SDK

Pod Integration SDK

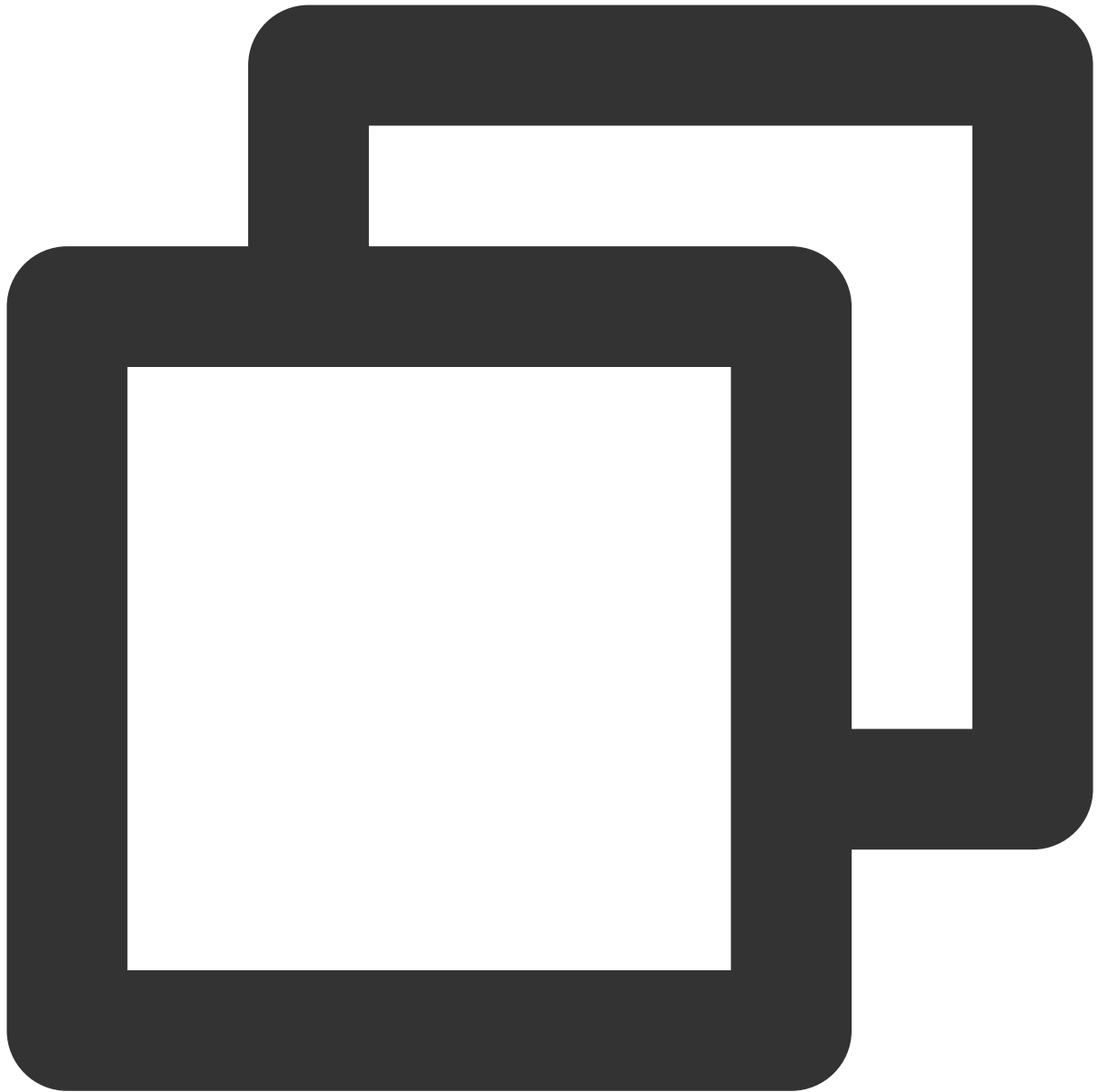
The LCIC SDK has been released to the Cocoapods library. You can download and install it by configuring the podfile.



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

Step 4: Configure app Permissions

Configure app permissions in the main app's info.plist. The LCIC SDK requires the following permissions:



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

Step 5: Obtain Parameters Required to Enter the Classroom

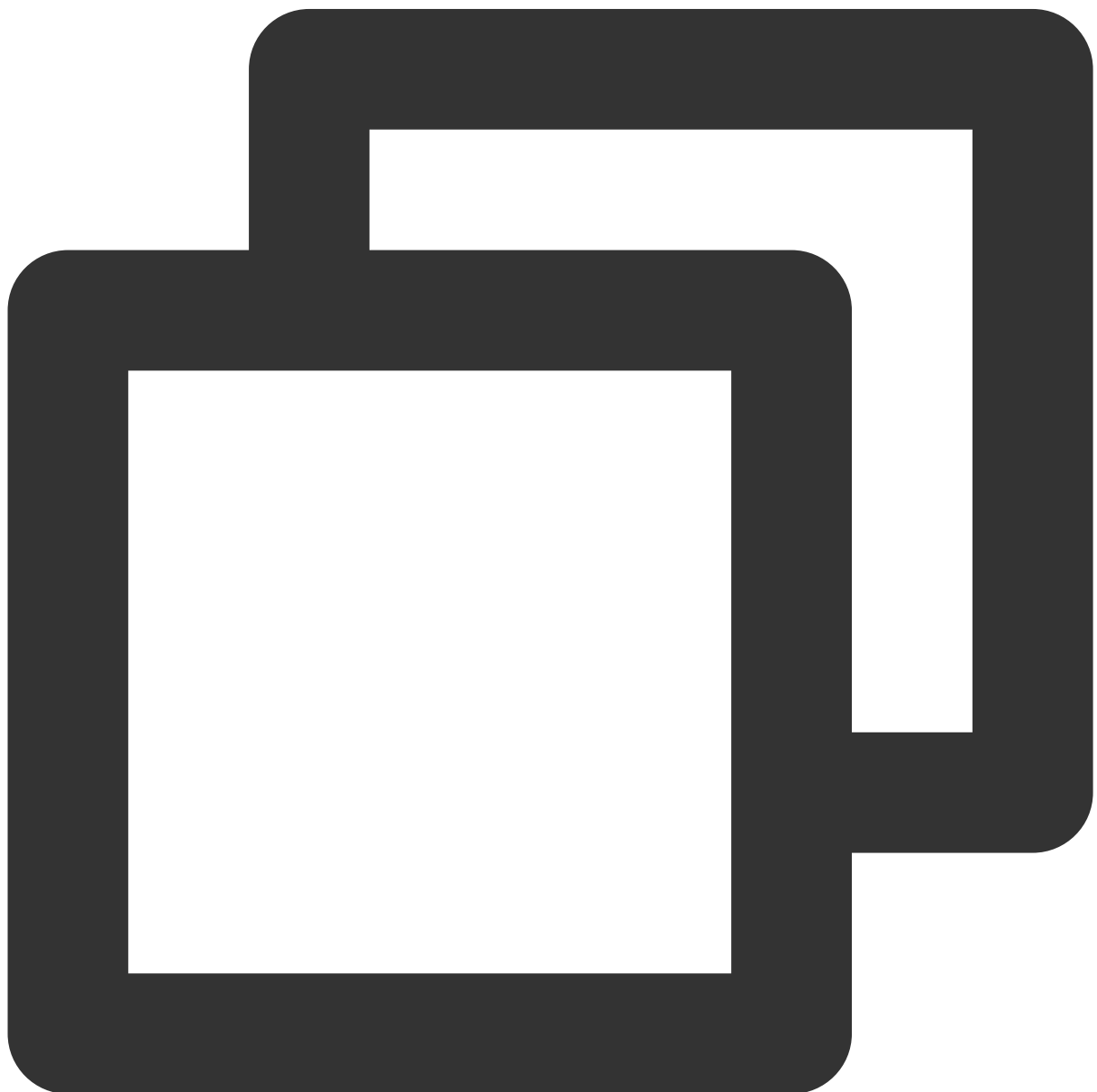
1. Choose **Application Management > Settings** through the [Console](#) to retrieve the [SDKAppld](#), which serves as the identifier for the school (schoolId).
2. By utilizing the Cloud API endpoint [CreateRoom](#), you can create a classroom and obtain the classroom number (classId).
3. Registering a user through the Cloud API endpoint [RegisterUser](#) allows for the acquisition of the corresponding user ID (userId).
4. Logging in via the Cloud API endpoint [LoginUser](#) grants access to the user authentication token.
5. Parameters such as scene, lng, camera, mic, and speaker are optional. Default values are employed if these parameters are not specified.

Field	Type	Meaning	Remarks	Mandatory
schoolId	Int	School Number	Choose Application Management > Application Configuration through the console to obtain the SDKAppld .	Yes
classId	Long	Classroom Number	By creating through the CreateRoom interface, you can obtain the returned RoomId.	Yes
userId	string	User account	It is obtained through the RegisterUser interface.	Yes
token	String	Backend authentication parameters	It is obtained through the LoginUser interface.	Yes
scene	String	Scene name	It is used to distinguish different custom layouts. It is configured through the SetAppCustomContent interface. [roomConfig setValue:@"scene_name" forKey:@"scene"];	No
lng	String	Language parameters	You can enter zh-Hans or en. The default value is Chinese (zh-Hans). You can set this parameter on this interface. [roomConfig setValue:@"en" forKey:@"language"];	No
camera	Int	Initialize and turn on the camera.	1 indicates to turn on the camera, and 0 indicates to turn it off. You can set this parameter through roomConfig.jsParams.	No
mic	Int	Initialize and turn on the microphone.	1 indicates to turn on the microphone, and 0 indicates to turn it off. You can set this parameter through roomConfig.jsParams.	No

speaker	Int	Initialize and turn on the speaker.	1 indicates to turn on the speaker, and 0 indicates to turn it off. You can set this parameter through roomConfig.jsParams.	No
---------	-----	-------------------------------------	---	----

Step 6: Launch the Main Page of the Component

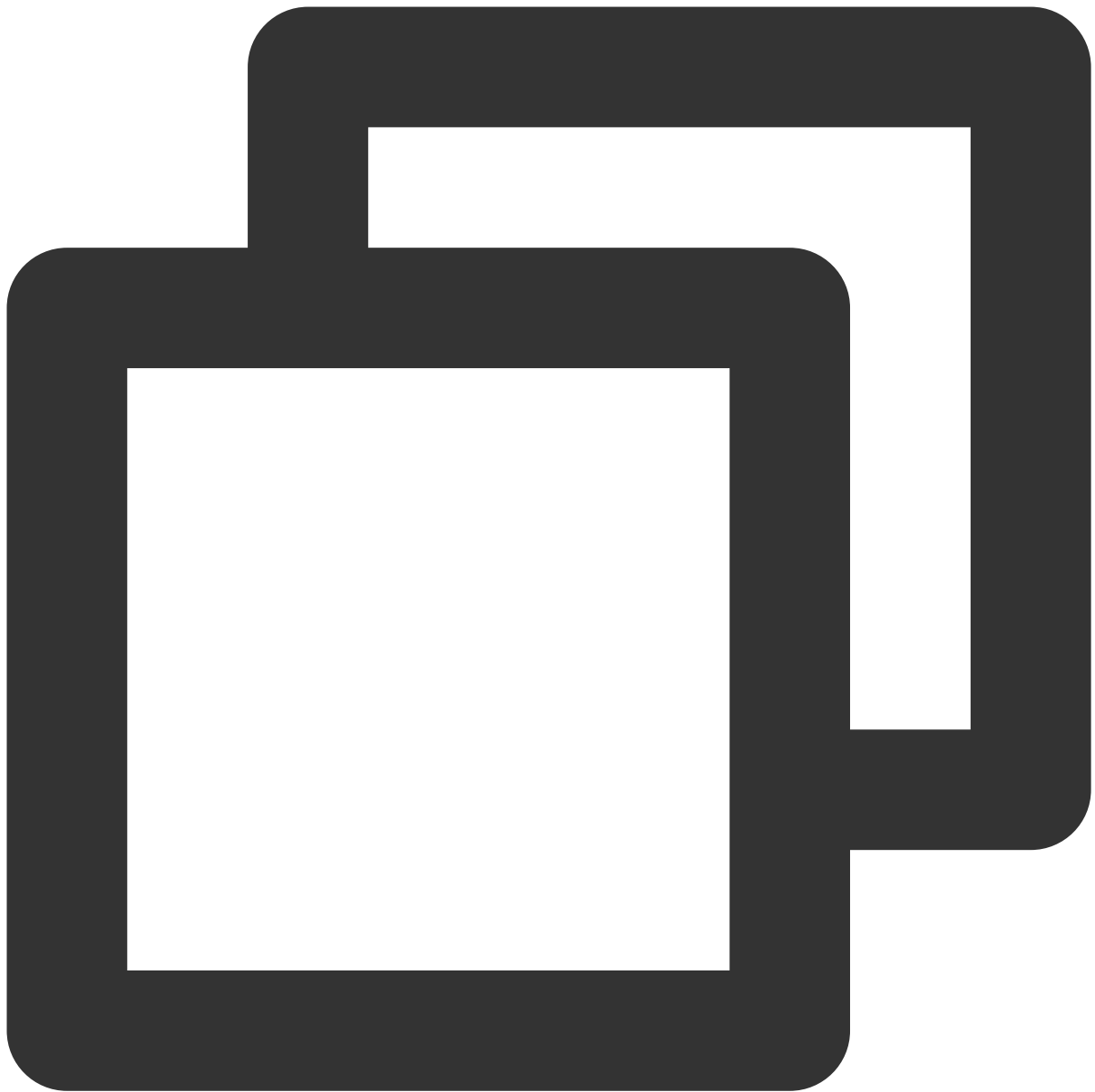
Only 4 parameters are needed to launch the main page of the LCIC component, namely, School Number, Classroom Number, User account, and token.



```
TCICClassConfig *roomConfig = [[TCICClassConfig alloc] init];
roomConfig.schoolId = 123456;
roomConfig.userId = "test";
roomConfig.token = "test_token";
roomConfig.classId = 654321;
[roomConfig setValue:@"en" forKey:@"language"]; //Language settings, optional
[roomConfig setValue:@"scene_name" forKey:@"scene"]; //Customized according to diff
[roomConfig setValue:@(0) forKey:@"preferPortrait"]; //Default landscape, optional

TCICClassController *vc = [TCICClassController classRoomWithConfig:roomConfig];
if (vc) {
    [(UINavigationController *)self.window.rootViewController pushViewController:vc
}
}else {
    NSLog(@"Incorrect parameter");
}
```

If you need to receive notifications of leaving the classroom, you can register a local broadcast for `TCICExitClassRoomCompleteNotify`. See the following code for reference:



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

Step 7: SDK Authorization Application

You need to send an email to apply for SDK permissions. Follow the template below and send an email to mediaservices@tencent.com. We will complete the processing within 1 working day, provided the information is

correct.

Note:

The package name is used for LEB player signature authorization. Provide the App Name, Package Name, and Bundle ID information for the official application that needs authorization, and send an email.

Recipient: mediaservices@tencent.com

Topic

LCIC iOS SDK Authorization Application

Content

Company Name: xxx Limited Company

Individual Name:

Contact Information:

App Name:

Package Name (Android):

Bundle ID (iOS):

Advanced Feature: Mobile Screen Sharing

1. Create an `App Group` . Refer to [TRTC Documentation](#).
2. Create a `Broadcast Upload Extension` . Refer to [TRTC Documentation](#).
3. For the newly created `Target` , depend on `TCICSDK_ReplayKit` as follows, and then simply `re-run pod install` .



```
target 'New target name' do
# Comment the next line if you do not need to use dynamic frameworks.
# use_frameworks!
pod 'TCICSDK_Pro_ReplayKit'
end
```

4. Add the following code to `SampleHandler.m` , and replace `APPGROUP` with the `App Group` created in Step 1.



```
#import "SampleHandler.h"
#import <TXLiteAVSDK_ReplayKitExt/TXLiteAVSDK_ReplayKitExt.h>
#import <TCICScreenKit/TCICScreenKit.h>
// Note: Replace `APPGROUP` with the ID of the App Group created earlier.
#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 {
    // The user has requested to pause the broadcast. Samples will stop being delivered.
    [[TCICScreenKit sharedScreenKit] onScreenKitPaused];
}
- (void)broadcastResumed {
    // The user has requested to resume the broadcast. Samples delivery will resume.
    [[TCICScreenKit sharedScreenKit] onScreenKitResumed];
}
- (void)broadcastFinished {
    [[TXReplayKitExt sharedInstance] finishBroadcast];
    // The 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 = @"Screen sharing ended";
            break;
        case TXReplayKitExtReasonDisconnected:
            tip = @"Application disconnected";
            break;
        case TXReplayKitExtReasonVersionMismatch:
            tip = @"Integration error (SDK version mismatch)";
            break;
    }
    NSError *error = [NSError errorWithDomain:NSStringFromClass(self.class) code:0
                                localizedFailureReasonErrorKey: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

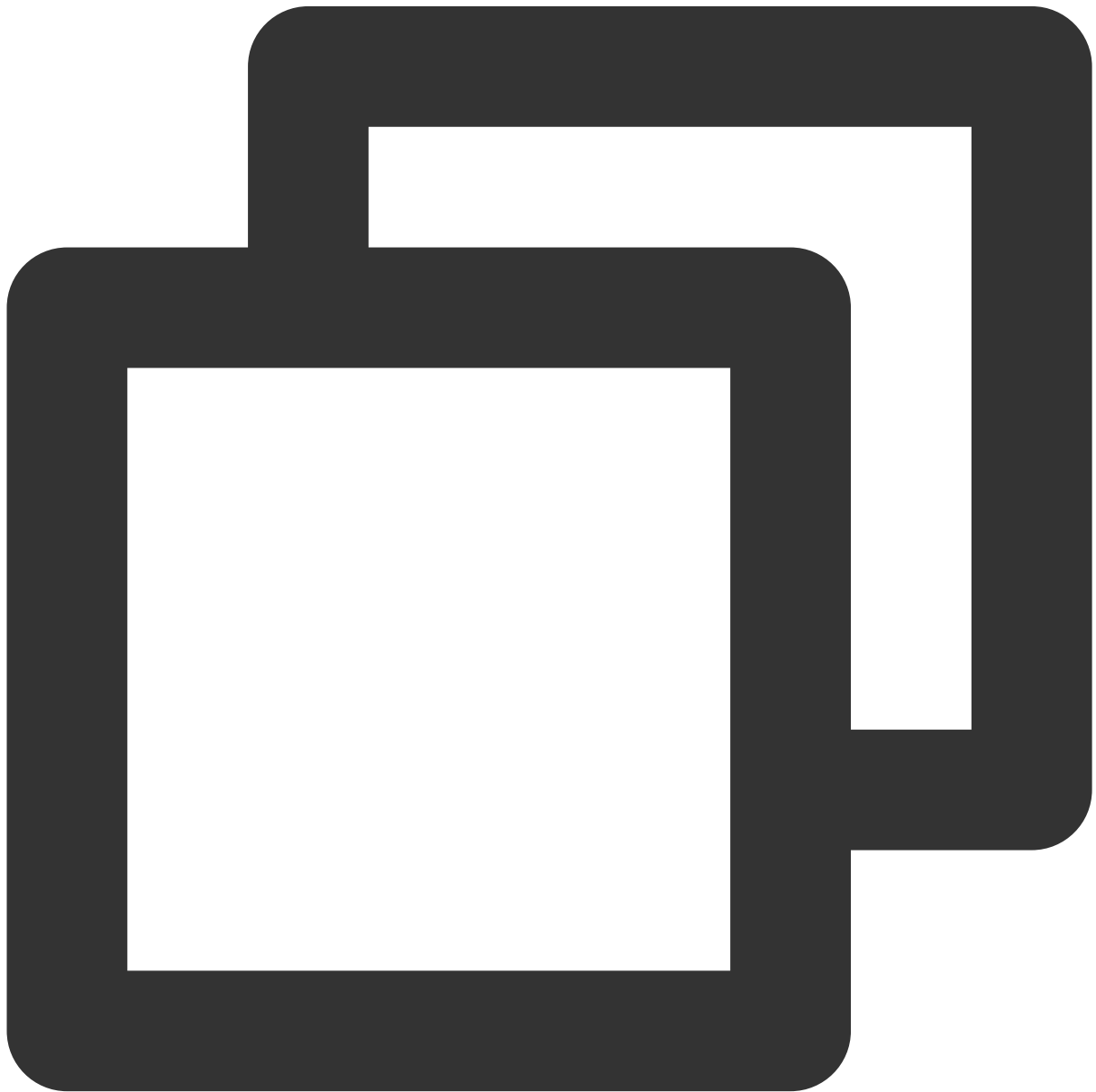
```



```
        break;

    default:
        break;
    }
}
@end
```

5. For integrating with the main app's receiving logic: Currently, the main app incorporates TCICSDK, which already supports system screen sharing logic. You only need to configure the App Group. In addition, set the App Group before entering the class.



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

// Set AppGroup using the KVC method.
[roomConfig setValue:@"group.com.xx.xxxx" forKey:@"appGroup"];
```

Notes

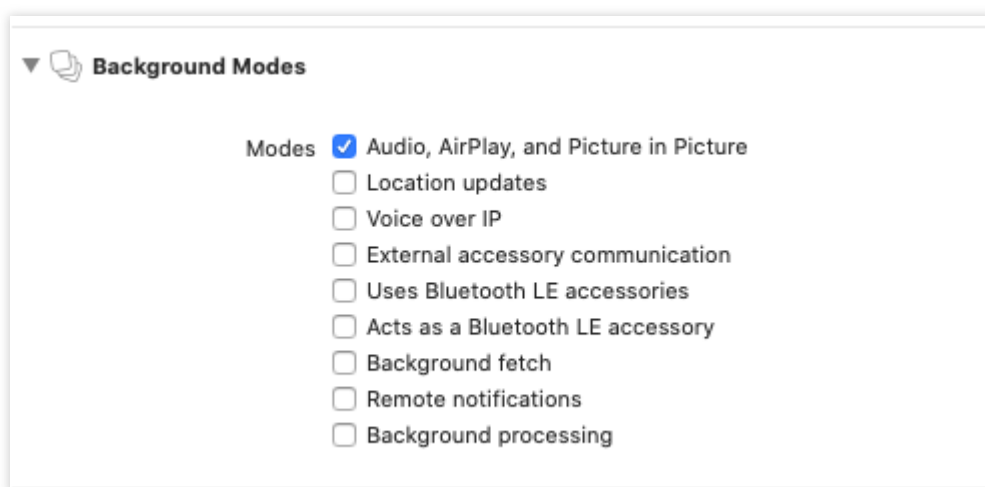
1. Currently, TCICSDK supports a trigger button for screen sharing. For details, refer to [TRTC Documentation](#) (optional). However, this feature has limitations.

1.1 The **trigger button for screen sharing** is only supported on iOS12 and above. In addition, the project created must not depend on the **Scene Lifecycle**. If the code already supports `Scenedelegate`, refer to [Xcode 11 Remove Scenedelegate](#) to remove it. For example, in the Demo, the pop-up effect is as follows. Click **Start live streaming** to begin.

1.2 For iOS11 devices, it is necessary to guide users to trigger from the remote control by **pressing and holding screen recording** and selecting the business's own `Broadcast Upload Extension` to trigger, as exemplified below with **Tencent Meeting**:

2. The `Upload Extension` created should have its `Deployment target` set to `iOS 11.0` (since `Replay Kit` is only supported from `iOS11`), and ensure that the physical device used for debugging is also on `iOS11` or later.

3. To support system-level screen sharing in the main app, `Background Modes` needs to be added.



Other Issues

We suggest integrating Tencent Bugly when using LCICSDK, which helps to rapidly identify and resolve anomalies, keep track of product operations dynamics, and promptly follow up on user feedback. Refer to the [Tencent Bugly official website](#) for the integration guide.

Refer to [Develop a demo](#).

Windows and macOS

Last updated : 2024-06-28 10:09:17

To provide cross-platform desktop applications, the development for Windows and macOS utilizes Electron. This article mainly describes how to quickly integrate Tencent Cloud LCIC Electron SDK into your project.

Prerequisites

You have completed the procedures for [Sign up for a Tencent Cloud account](#) and [Identity Verification](#).

Step 1: Create New Application

1. Log in to the [Low-code Interactive Classroom](#), and select **Quick Start** from the left navigation bar.
2. By default, you enter the "Create application" interface, where you can select "**Create new application**" for the application type. Enter the application name, such as TestLCIC.

If you have already created an application, you can click "Select an existing application" for the application type.

Note:

Each account can get a trial version application for free. If there are no trial applications under the current account, you can directly create one to experience. To create a commercial application, you can create the corresponding version of the application on the [Purchase Page](#) according to business needs.

3. Based on actual business requirements, add or edit tags, and then click **Create**.

1 Create application > 2 Generate Class Parameters > 3 Success

Create/Select Application

Type

☒ Create new application ☐ Select an existing application

Application

Enter an application name

0/15

Add tags (optional)

Tags are used to classify and manage resources from different dimensions. If the existing label does not meet your requirements, please go to [Tag r](#)
Once the tags are updated, please [Click here to refresh tab](#)

Tags

Please select a tab key

Please select a tag value

+ add tags

Create

Reset

Note:

Application names may only contain underscores, digits, or letters.

Tags are used to identify and organize your resources in Tencent Cloud. For instance, a corporation with several business units, each of which has one or more LCIC applications, can use tags to label departmental information on LCIC applications. Tags are optional, and you can add or edit them based on your actual business needs.

Step 2: Enter SDKAppId and Key (SecretKey)

1. Choose [Application Management](#) > **Settings** to obtain the SDKAppId.
2. Enter the [Cloud Access Management](#) obtain the key. If there is no key, you need to create a new one in **API Key Management**. For details, refer to [Key Management](#).

Step 3: Obtain Parameters Required to Enter the Classroom

1. By calling the TencentCloud API interface [RegisterUser](#) to register users, you can obtain the corresponding user ID (`userid`) information.
2. By calling the TencentCloud API interface [LoginUser](#) to log in, you can obtain the user authentication `token` information.
3. By using the Application Programming Interface [CreateRoom](#) to create a classroom, you can obtain the Classroom ID (`classid`) information.
4. Specify the version of the classroom you need to integrate: latest.
5. The parameters `scene` , `debugjs` , and `debugcss` are optional and only need to be set when you customize the UI. For more information, refer to [Custom UI Integration](#). `debugjs` and `debugcss` are used only for debugging custom layouts and components, and are only accessible through addresses `localhost` or `127.0.0.1` . Do not use these parameters in the release phase.
6. `lng` , `location` , and `layout` are also optional parameters. You can determine whether to enter these parameters based on actual business needs. If not, they will use the default values. The `layout` parameter is only effective when the classroom layout is set to video + document layout (`videodoc`).

Field	Type	Meaning	Remarks	Mandatory	
userid	String	Username	It is obtained through the RegisterUser interface.	Yes	

classid	String	Classroom ID	It is obtained through the return of creating the CreateRoom interface.	Yes	
token	String	Backend authentication parameters	It is obtained through the LoginUser interface.	Yes	
version	String	Version number of the classroom	(This parameter will be officially deprecated from tcic-electron-sdk version 1.9.0) The corresponding version can be selected from the release logs.	No	
scene	String	Scene name	It is used to distinguish different custom layouts. It is configured through the SetAppCustomContent interface.	No	
role	String	Role for entering the classroom, default is empty	Optional parameter supervisor (Patrol Classes/Content Review) is only available to registered users who can patrol classes within the application.	No	
debugjs	String	JS link of the custom defined UI	It is obtained through the custom defined UI integration.	No	
debugcss	String	CSS link of the custom defined UI	It is obtained through the custom defined UI integration.	No	
lng	String	Language parameter, which is zh-CN by default	Currently zh-CN and en-US are supported.	No	
location	Boolean	Whether to report geographical location information	It is set to false by default, indicating no reporting.	No	
layout	String	Page layout	It is set to top (top) by default. Currently, it supports double row layout (double), right layout (right), left layout (left), and three-part layout (three).	No	

Step 4: Enter the Classroom

For different business needs, we offer the following two integration methods.

Method 1: URL Concatenation

This capability allows the client to be launched through a browser. The clients are classified into Tencent Cloud-provided LCIC-Demo Client and the Customized Client. The difference lies in the client icon, name, and initial page.

Directly Launching the Client Through a Browser

1. Download and install the client.

[Windows 64-bit](#)

[Windows 32-bit](#)

[macOS](#)

2. The client is launched through a link. When you access or navigate to the URL

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

`classid=${classId}&userid=${userId}&token=${token}` in a browser, the browser will request to open the LCIC-Demo Client.

Transfer Page Invocation Method (Recommended)

By integrating LCIC Electron SDK quickly and easily, you can jump to the transfer page on the business side. If the recognition is successful, it will directly jump. If it fails, downloading the client is prioritized. If you cancel or opt not to download the client, the web version of the class page is opened through redirection.

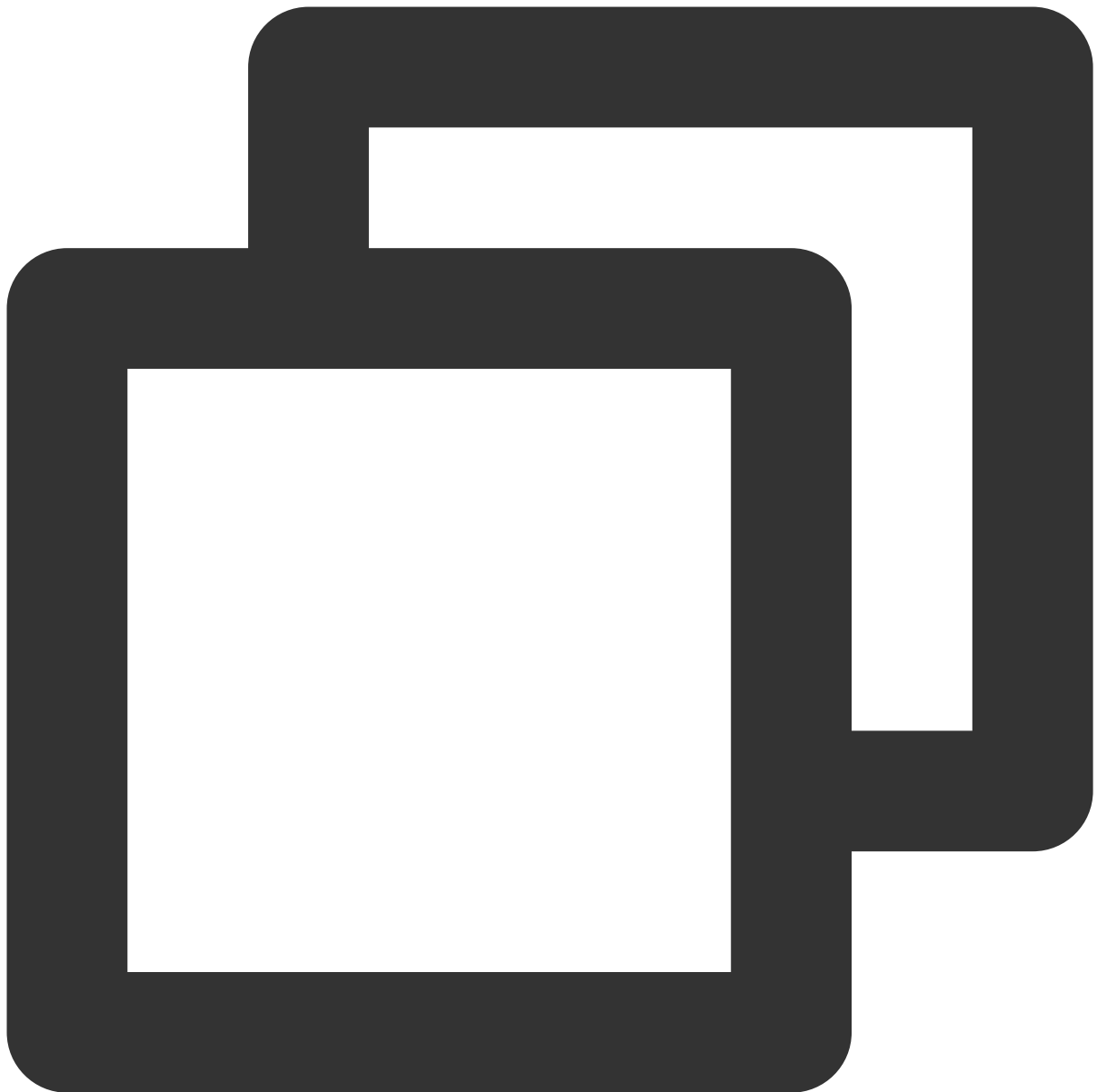
Specific Process:

Before entering the classroom, it is necessary to jump to a transfer page first, where the user jump logic is handled. By obtaining the class parameters and appending them to this URL

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

`classid=${classId}&userid=${userId}&token=${token}`, the client application can be invoked.

Download [ElectronProtocolCheck.js](#) and integrate it into your own project. The sample code in the transfer page is as follows:



```
// For ElectronProtocolCheck file code, see the GitHub example
import ElectronProtocolCheck from './ElectronProtocolCheck';

// The ID required to enter the classroom
const classId = 368507569;
// The ID of the user currently entering the classroom
const userId = "JIUzI1NiIsIn123456";
// The token needs to be dynamically obtained from the backend interface to prevent
const token = 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhhd
const url = `tcic://class.qcloudclass.com/latest/class.html?classid=${classId}&us
console.log(`callClient->start: ${url}`);
```

```
// Invoke client
ElectronProtocolCheck(
  url,
  successCallback: (res) => {
    // Call successful
    console.log('callClient->success! ');
  },
  failCallback: (res) => {
    console.log('callClient->failed! ');
    // If the call fails, it is recommended to pop up a window that suggests the

    // If the user clicks to download --> Change the pop-up window information, a

    // If the user cancels the download or closes the pop-up window, let the user
    // You can also add a timeout pop-up window. Generally, if the user does not
  },
  unsupportedCallback: () {
    // For browsers that are not supported, you can use a web link to load.
  });
```

We provide a complete Demo. Click [LCIC-Electron-Demo](#) to view.

Note:

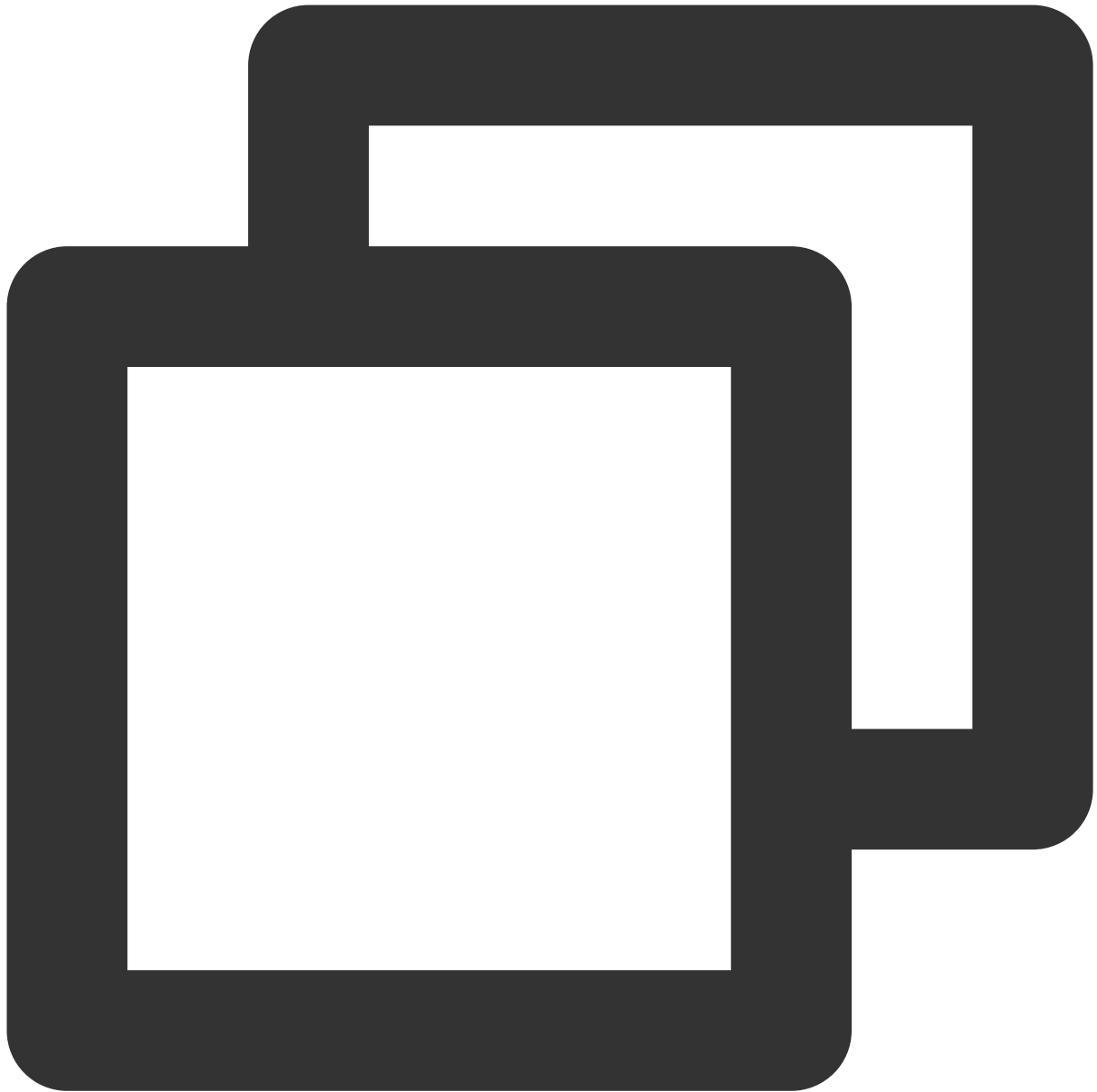
In the concatenated URL, if the pass-in `userid` matches the `teacherid` specified for the current classroom, the current user is a teacher. If it matches the current classroom's assistant ID (`assistantid`), the user is an assistant. Otherwise, the user is a student.

Method 2: SDK Integration

1. Due to some dependency libraries, ensure that you use the following base library versions.

Development framework	Version
Electron	10.4.3+
Node	10.14.1+

2. Use the npm command to install the SDK package in your project.

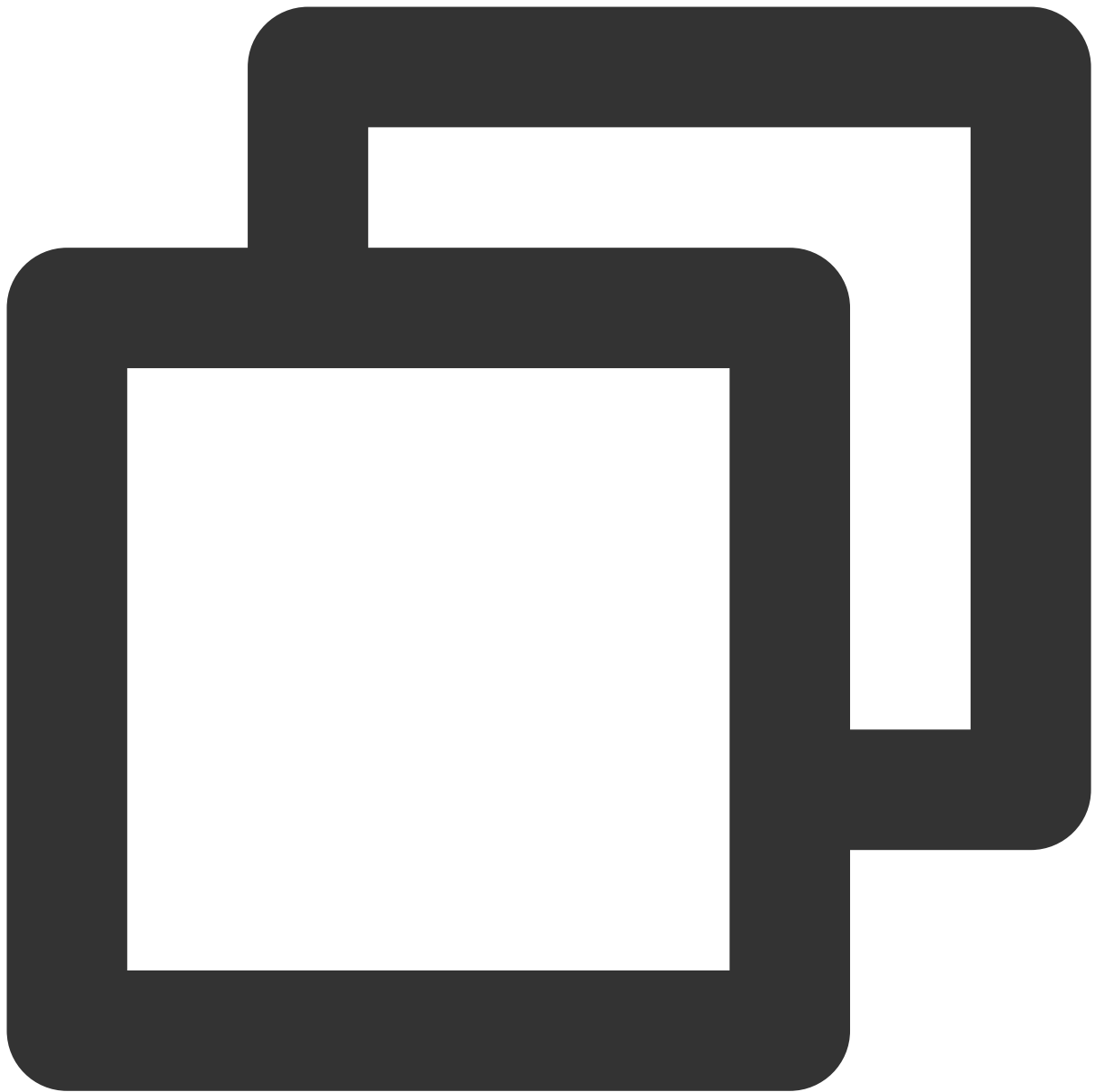


```
npm install tcic-electron-sdk@latest
```

Note:

The latest version of the TCIC Electron SDK can be found at [tcic-electron-sdk](#).

3. After importing the module into the project script, call the initialization interface and pass in the previously obtained parameters to launch the in-class page.



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

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

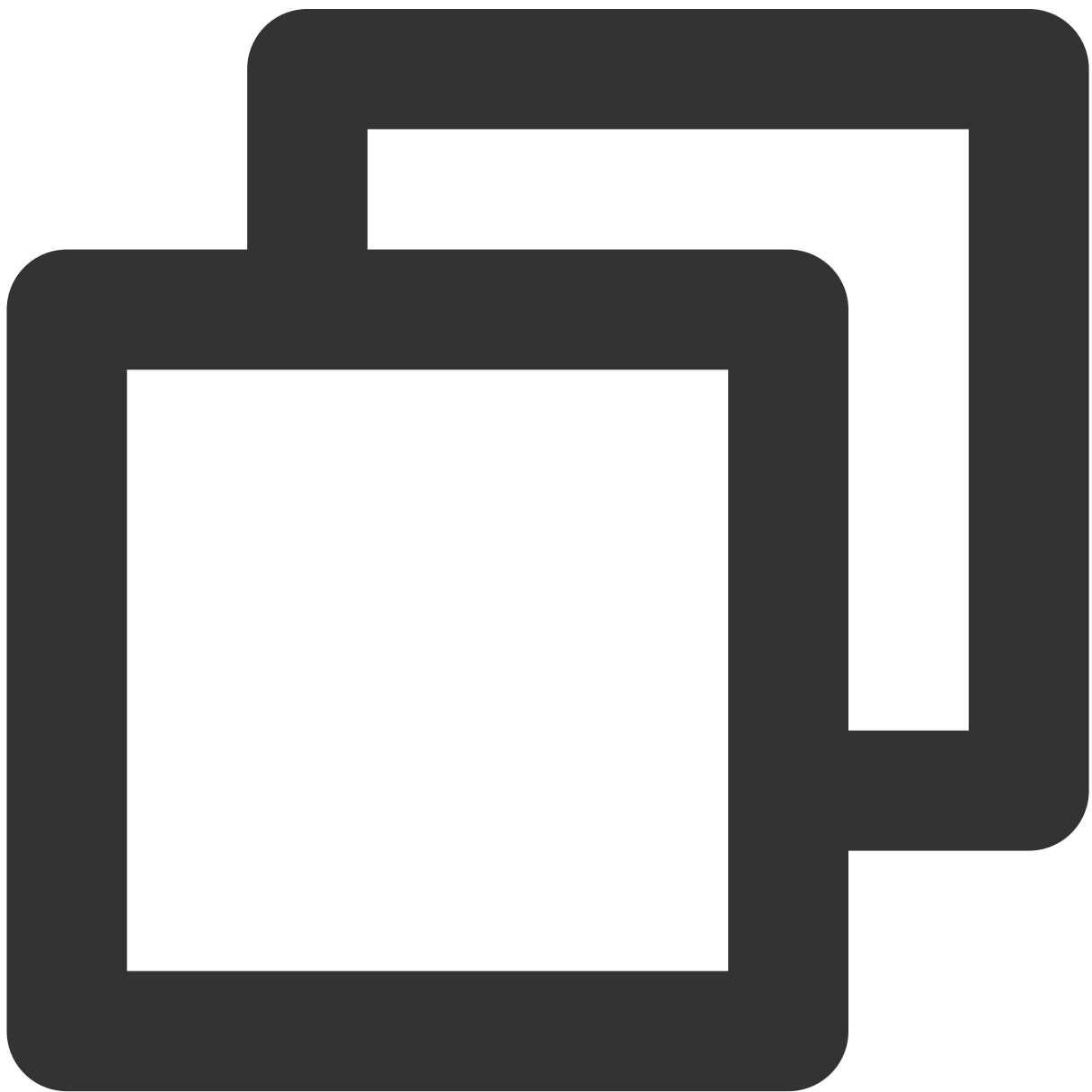
Advanced Features

Assembly Line Packaging Integration

If you need the generated client to have its own brand logo and application name while being able to quickly integrate existing business systems, you can use the assembly line packaging method for integration.

1. Configure the Business Logic

After packaging, you can enter the class through the `window` global variable's `joinClass` method on the business side. If the course entry logic has been implemented through URL or SDK integration on the business side, see the following example for compatibility.

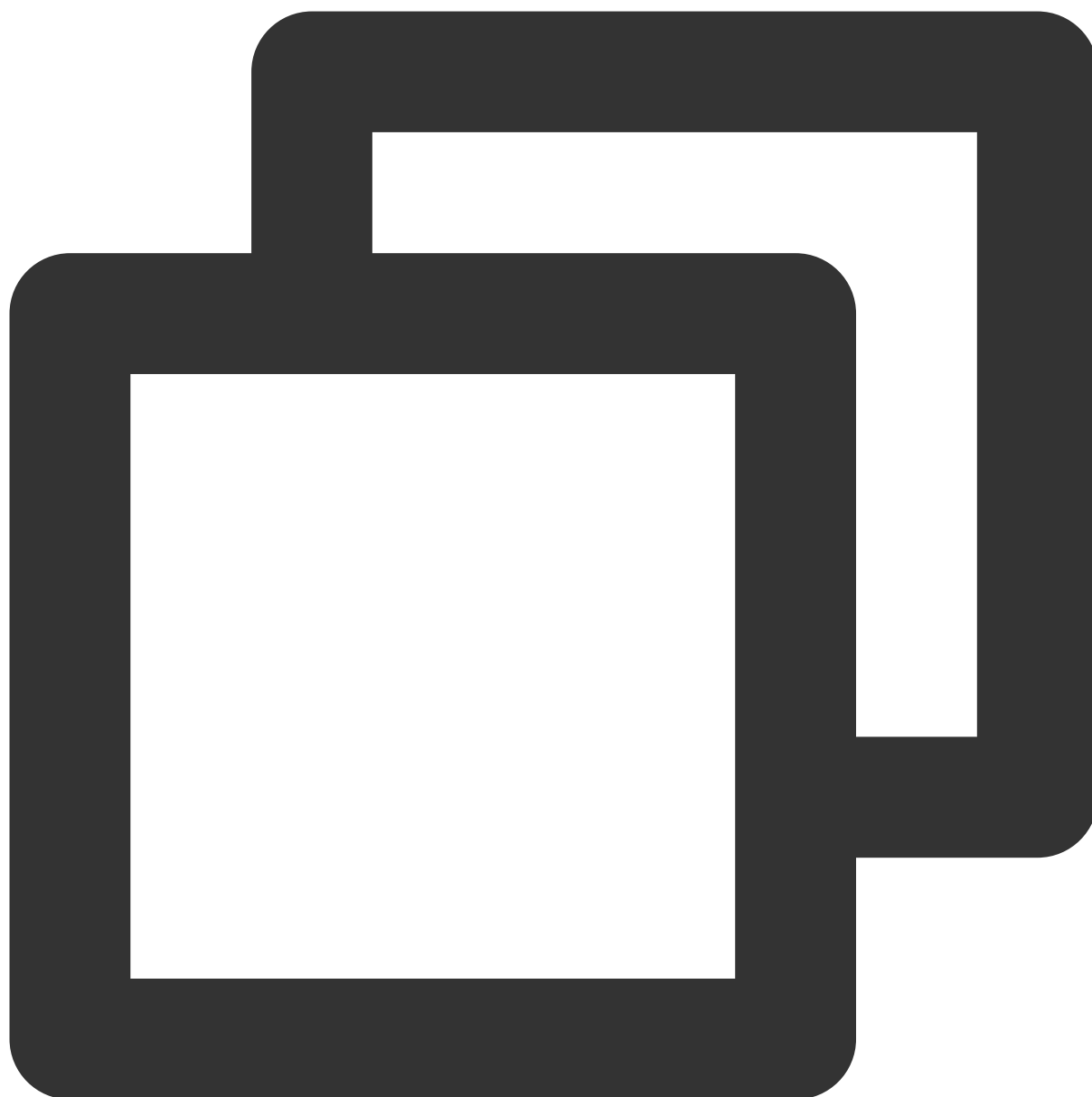


```
const options = {
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhlhdCI6MTY3O
};
// Logic method for clicking to enter the classroom
window.joinClass(options);

// How to maintain compatibility with the existing logic? Refer to the code logic b

// Check whether the method exists.
if (window.joinClass) {
  window.joinClass(options);
} else {
  // Original logic for entering the classroom
}
```

The current client should be closed on the business side through the `window` global variable's `closeWin` method.



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

2. Prepare the corresponding Bill of Materials, as detailed below:

Field	Meaning	Mandatory
AppName	App name, such as XX Classroom	Yes
Logo	Application Logo, which should be an ico/png image with specifications of 256x256	Yes

URL	Business URL	Yes
-----	--------------	-----

Note:

The Business URL should be a complete URL already in a logged-in status. If not logged in, it should automatically redirect to the login entry. The configured Business URL should not be a login page. Otherwise, users will need to log in every time they open it.

3. Send a packaging application email in the following format and attach the above materials as an attachment. Once the information and materials are confirmed to be correct, we will complete the packaging within 1 working day.

Note:

Recipient: mediaservices@tencent.com

Topic

Client Packaging Application

Content

Company Name: xxx Limited Company

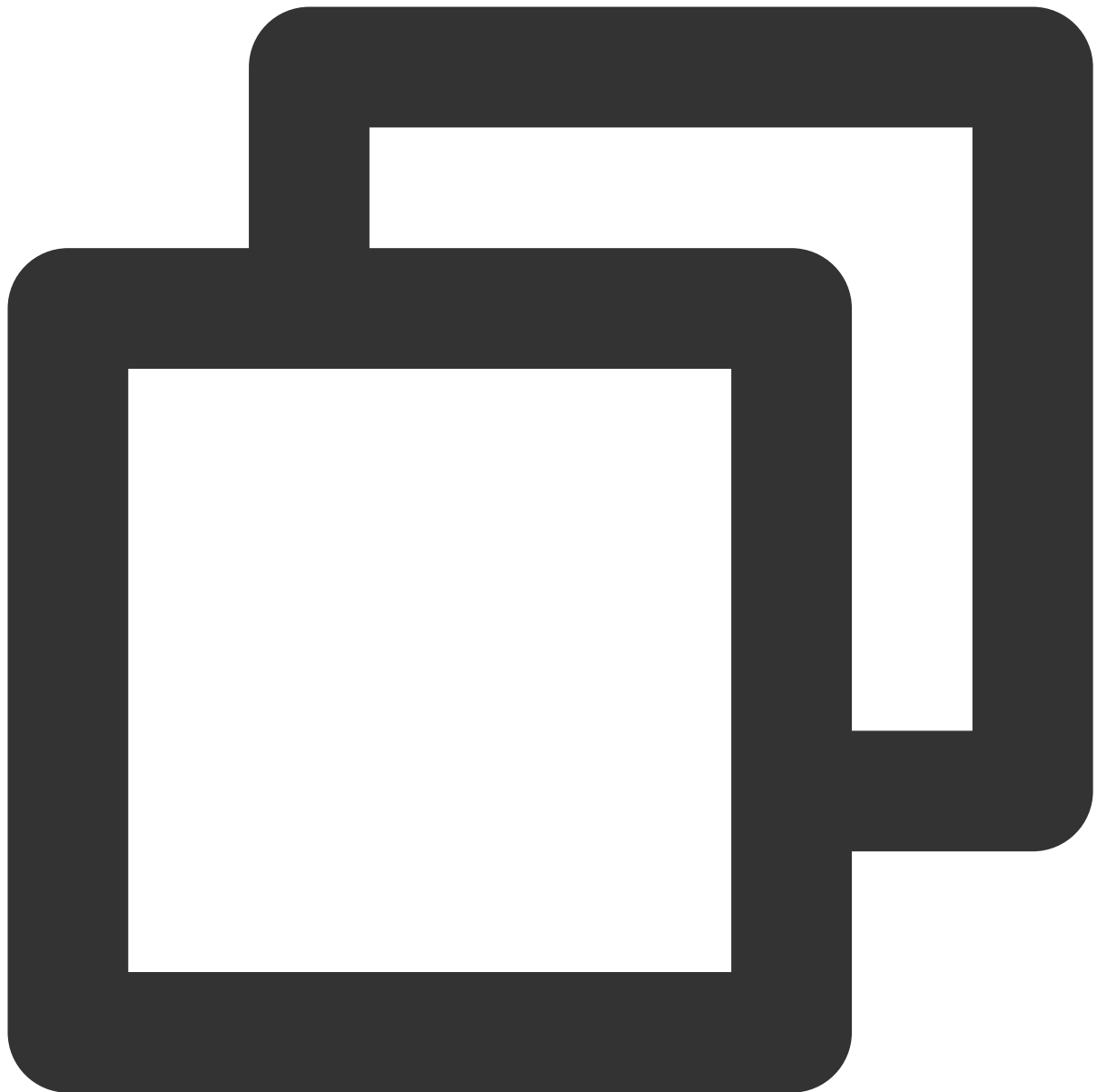
Individual Name:

Contact Information:

Packaging Materials (Attachment)

Custom UI Integration

To meet the diverse needs of different customers, LCIC Electron currently also offers an integration plan for Custom UI. You can define the layout and style of the course on the business side. Through [customizing the interface](#), you can get the JS and CSS links on the business side. Just concatenate the `debugjs` and `debugcss` parameters to the above link (**these parameters are for debugging purposes only**), as shown in the following code:



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

// SDK Integration Method
TCIC.initialize({
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImhhdCI6MTY3O
  debugjs: '%!s(<nil>)',
  debugcss: '%!s(<nil>)',
})
```

```
// Pipeline Packaging Method
const options = {
  classId: '368507569',
  userId: '123456',
  token: 'yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2ODAwNzQwMjEsImklhdCI6MTY3O
  debugjs: '%!s(<nil>)',
  debugcss: '%!s(<nil>)',
};
window.joinClass(options);
```

Once the custom JS and CSS debugging is completed, you can use the TencentCloud API interface

[SetAppCustomContent](#) or [Console > Application Configuration > Settings](#) to bind the scene with the custom JS, CSS links. By appending the `scene` parameter to the URL or adding it to the corresponding entry when you enter the classroom, the layout and components of the specific scene will be loaded. When dealing with multiple class types and layouts, you can switch scenes using this parameter on the business side.

Custom Business Domain

On the in-class page, if you need to hide the classroom domain and only display the business domain on the business side, you can create a custom business domain through the [CDN Console \(CDN\)](#) and set it to origin from the classroom domain. For detailed process, refer to [Custom Business Domain](#).

Other Related Documents

[LCIC API](#)

[Custom UI Integration](#)