

Tencent Effect SDK

SDK Integration Guide (No UI)

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

SDK Integration Guide (No UI)

Integrating Tencent Effect SDK

iOS

Android

Integrating Capabilities

Gesture Detection

Facial Keypoint Detection

Virtual Background

Facial Features

Android

Facial Expressions

iOS

Android

Body Keypoints

iOS

Android

Converting Audio to Expressions

Converting Audio to Expression

iOS

SDK Integration Guide (No UI)

Integrating Tencent Effect SDK

iOS

Last updated : 2023-07-21 15:05:54

Prerequisites

Environment requirements

Xcode 11 or later (download from App Store or [here](#))

Recommended runtime environment:

Device requirements: iPhone 5 or later. iPhone 6 and older models support up to 720p for the front camera.

System requirements: iOS 10.0 or later.

Importing the SDK

You can use CocoaPods or download and import the SDK manually into your project.

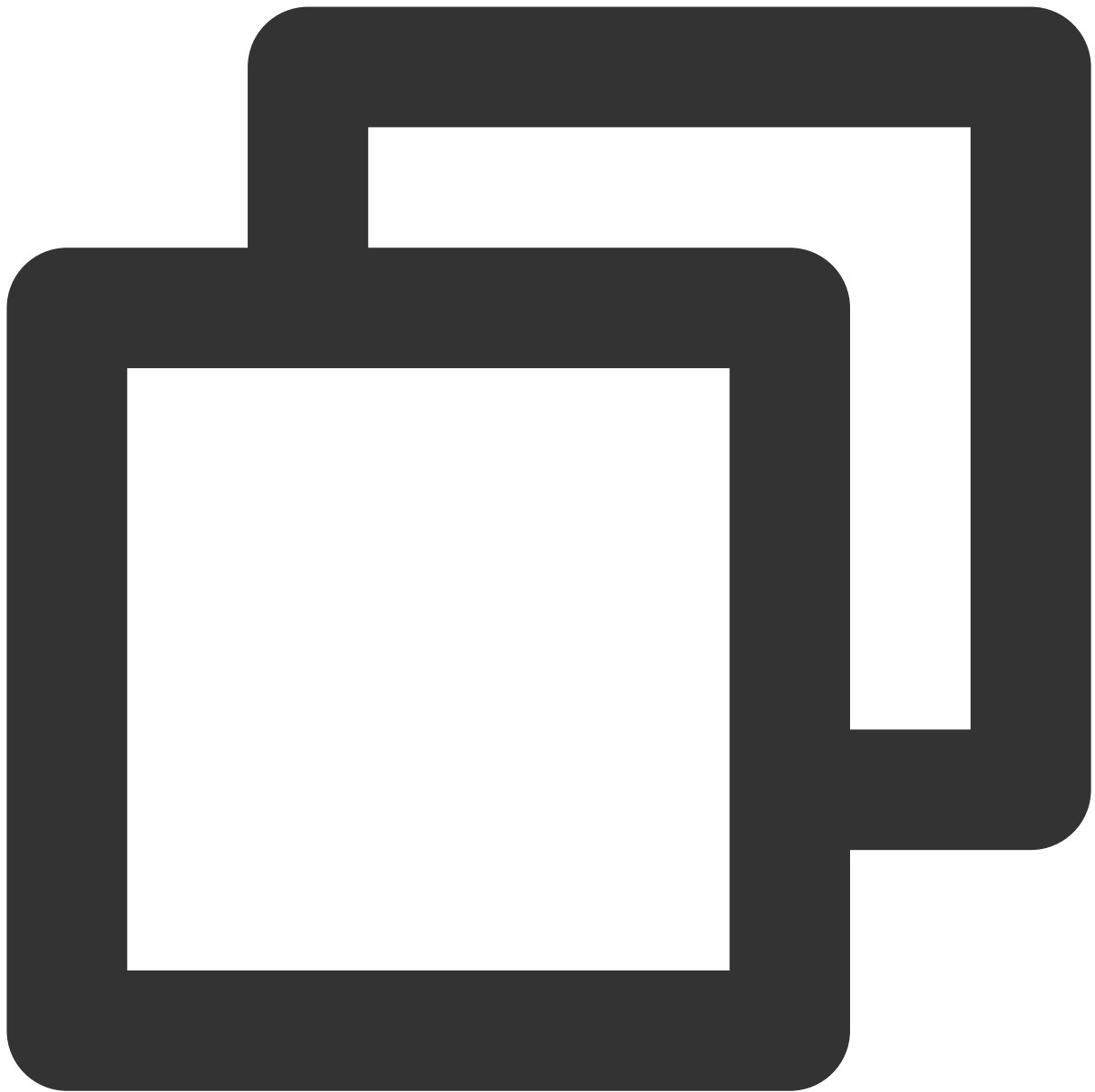
CocoaPods

Manual import

Dynamic download and integration

1. Install CocoaPods

Enter the following command in a terminal (you need to install Ruby on your Mac first):



```
sudo gem install cocoapods
```

2. Create a Podfile

Go to the directory of your project and enter the following command to create a Podfile in the directory.



```
pod init
```

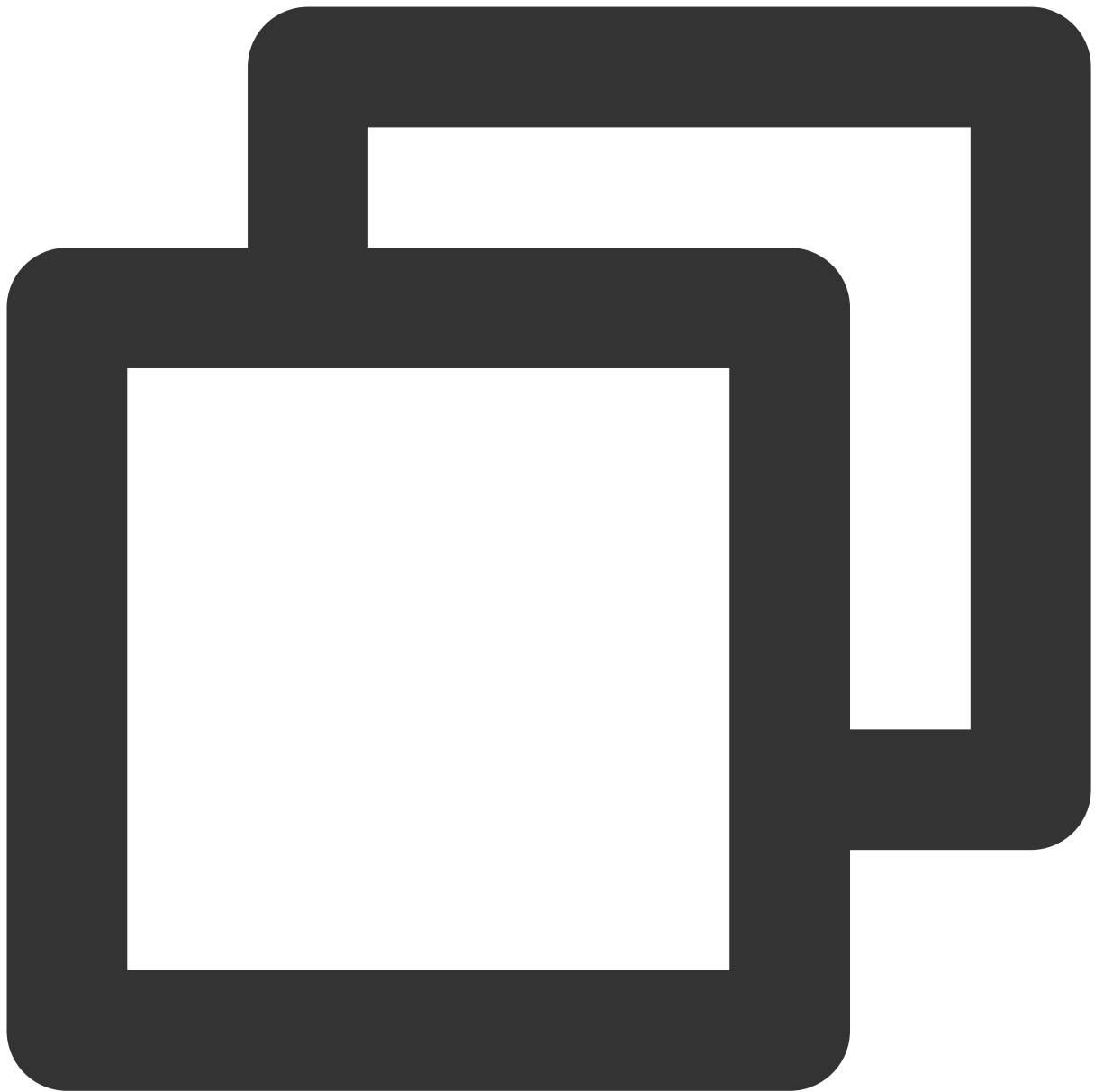
3. Edit the Podfile.

Choose an edition for your project and edit the Podfile:

XMagic version before 3.0.1:

- **XMagic Standard**

Edit the Podfile as follows:



```
platform :ios, '8.0'

target 'App' do
  pod 'XMagic'
end
```

XMagic Lite

The installation package of XMagic Lite is smaller than XMagic Standard. It supports only Basic A1 - 00, Basic A1 - 01, and Advanced S1 - 00. Edit the Podfile as follows:



```
platform :ios, '8.0'

target 'App' do
  pod 'XMagic_Smart'
end
```

XMagic version 3.0.1 and later:

Choose the appropriate version based on your project package and edit the Podfile file:



```
#Please install the corresponding library with 'pod install' based on your package
#For example: if your package is of type 'all', then you only need to use pod 'Tenc
#For example: if your package is of type 'S1-04', then you only need to use pod 'Te
pod 'TencentEffect_All'
#pod 'TencentEffect_A1-00'
#pod 'TencentEffect_A1-01'
#pod 'TencentEffect_A1-02'
#pod 'TencentEffect_A1-03'
#pod 'TencentEffect_A1-04'
#pod 'TencentEffect_A1-05'
#pod 'TencentEffect_A1-06'
```

```
#pod 'TencentEffect_S1-00'  
#pod 'TencentEffect_S1-01'  
#pod 'TencentEffect_S1-02'  
#pod 'TencentEffect_S1-03'  
#pod 'TencentEffect_S1-04'  
#pod 'TencentEffect_S1-05'  
#pod 'TencentEffect_S1-06'  
#pod 'TencentEffect_S1-07'  
#pod 'TencentEffect_X1-01'  
#pod 'TencentEffect_X1-02'
```

4. Update the local repository and install the SDK

Enter the following command in a terminal window to update the local repository and install the SDK:



```
pod install
```

An XCWORKSPACE project file integrated with the SDK will be generated. Double-click to open the file.

5. Add effect resources to your project

Download the [SDK and effect resources](#) for the Tencent Effect package you use, decompress the file, and add **all the bundle files** except **LightCore.bundle**, **Light3DPlugin.bundle**, **LightBodyPlugin.bundle**, **LightHandPlugin.bundle**, **LightSegmentPlugin.bundle**, and **audio2exp.bundle** in the **resources** folder to your project.

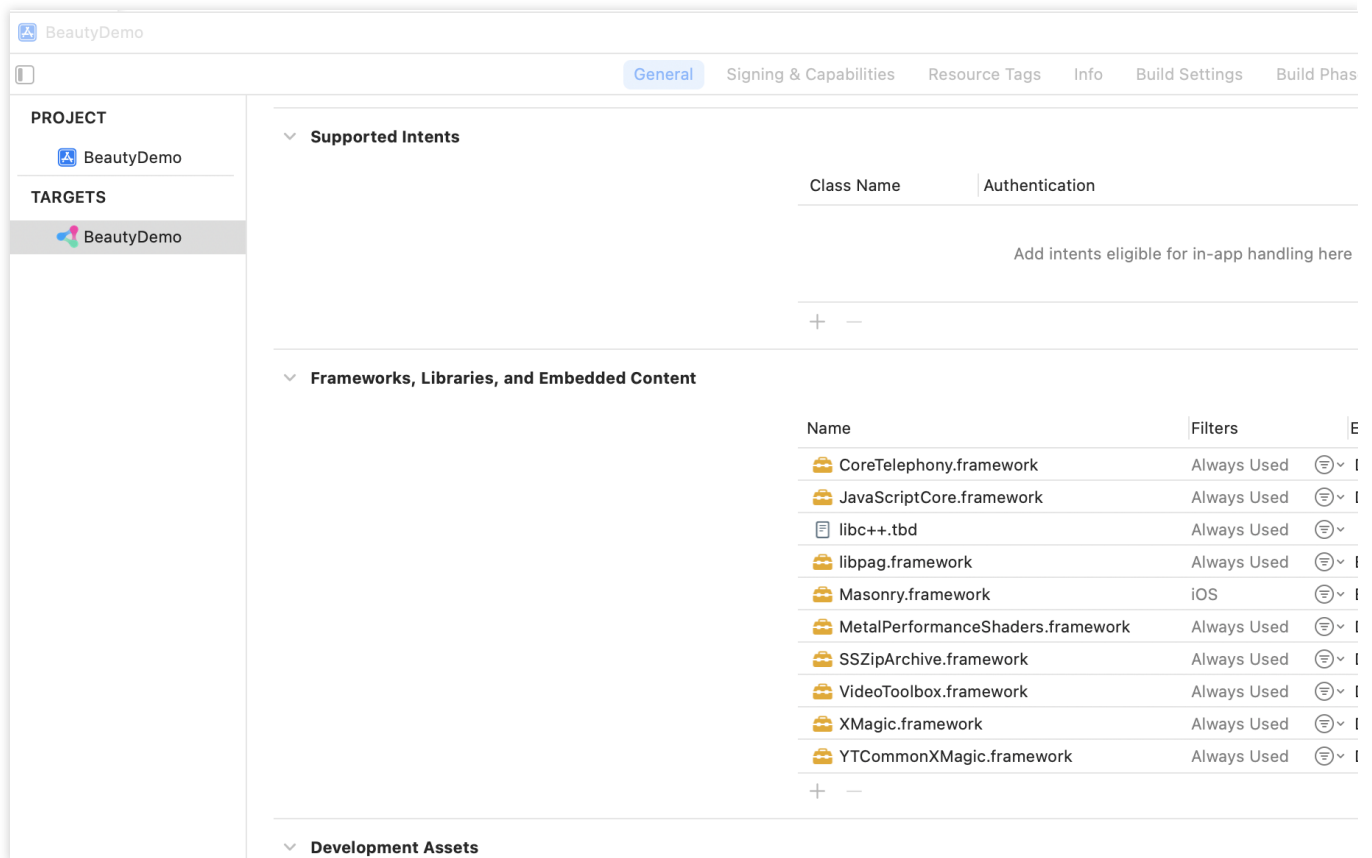
Under the **Build Settings** tab, add `-ObjC` to **Other Linker Flags**.

6. Change the bundle ID to the bundle ID bound to your license.

1. Download the [SDK and effect resources](#) and decompress the file. The SDK is in the `frameworks` folder, and the bundle resources are in `resources`.

2. If your SDK version is earlier than 2.5.1:

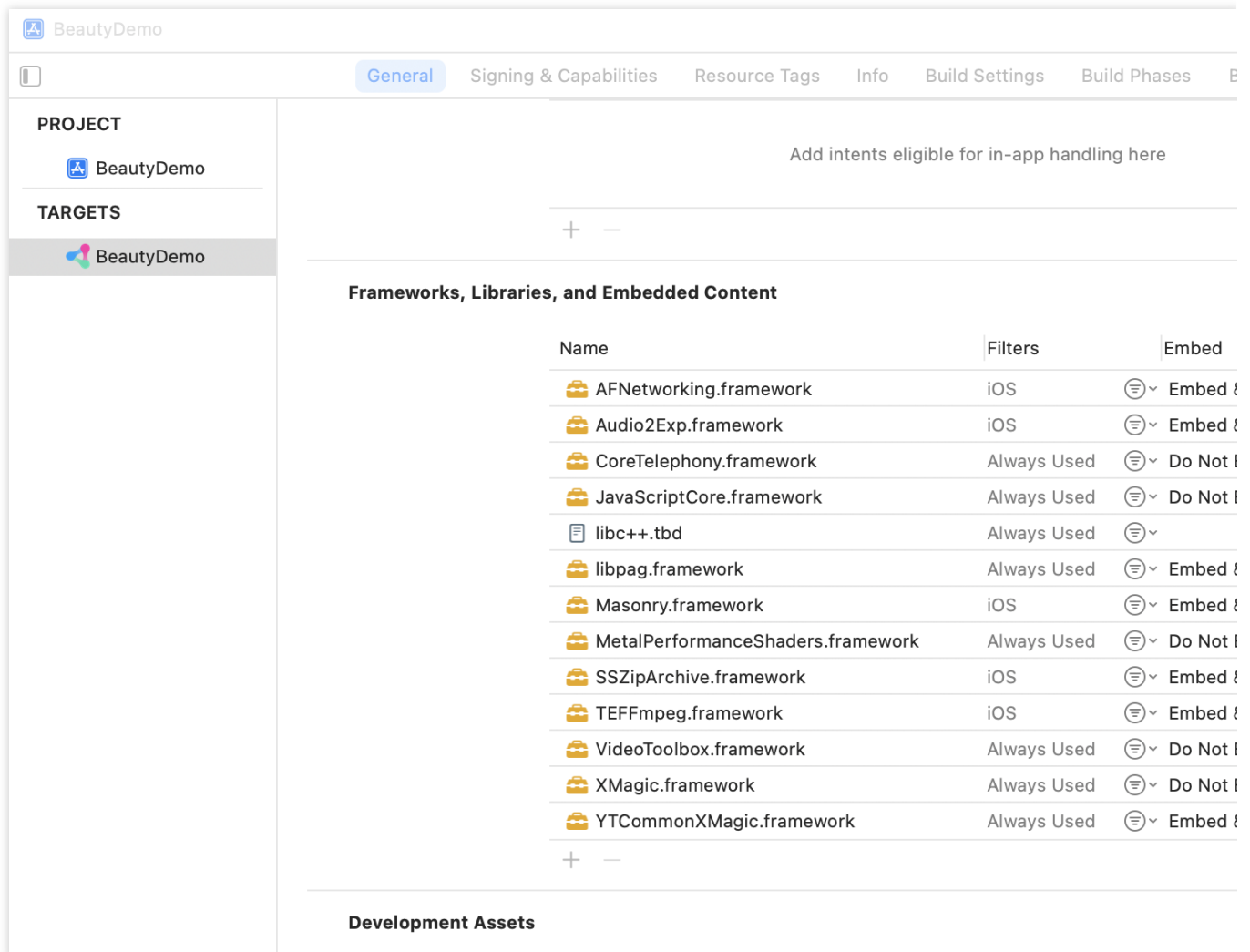
Open your Xcode project and add the frameworks in the `frameworks` folder to your project: Choose the target to run, select the **General** tab, expand **Frameworks, Libraries, and Embedded Content**, and click **+** to add the frameworks downloaded, including `XMagic.framework`, `YTCommonXMagic.framework`, and `libpag.framework`, as well as `MetalPerformanceShaders.framework`, `CoreTelephony.framework`, `JavaScriptCore.framework`, `VideoToolbox.framework`, and `libc++.tbd`. You can also add `Masonry.framework` (control layout) and `SSZipArchive` (file decompression) if necessary.



If your SDK version is 2.5.1 or later:

Open your Xcode project and add the frameworks in the `frameworks` folder to your project: Choose the target to run, select the **General** tab, expand **Frameworks, Libraries, and Embedded Content**, and click **+** to add the frameworks downloaded, including `XMagic.framework`, `YTCommonXMagic.framework`, `libpag.framework`, `Audio2Exp.framework`, and `TEFFmpeg.framework` (Renamed to `TECodec.framework` after version 3.0.0.) , as well as

`MetalPerformanceShaders.framework` , `CoreTelephony.framework` ,
`JavaScriptCore.framework` ,
`VideoToolbox.framework` , and `libc++.tbd` . If necessary, you can also add `Masonry.framework`
 (control layout) and
`SSZipArchive` (file decompression).



3. Add the effect resources in the `resources` folder to your project.

4. Under the **Build Settings** tab, add `-ObjC` to **Other Linker Flags**.

5. Change the bundle ID to the bundle ID bound to your license.

To reduce the SDK package size, you can dynamically download the necessary module resources and animated effect resources (`MotionRes` , not available in some basic editions of the SDK) from a URL and, after download, pass the path of the resources to the SDK.

You can use your existing download service, but we recommend you use the download logic of the demo. For detailed directions on implementing dynamic download, see [Reducing SDK Size](#).

Configuring permissions

Add permission descriptions in the `Info.plist` file. If you don't do so, the application will crash on iOS 10. Grant the application camera access in **Privacy - Camera Usage Description**.

Directions

Step 1. Authenticate

1. Apply for a license and get the `LicenseURL` and `LicenseKEY` .

Note

Under normal circumstances, the authentication process can be completed by connecting your application to the internet one time, so you **don't need** to put the license file in the project directory. However, if your application needs to use the SDK features without ever connecting to the internet, you can download the license file and put it in the project directory. In this case, the license file must be named `v_cube.license` .

2. Set the URL and key in the initialization code of your business module to download the license. Avoid downloading it just before use. You can also trigger the download in the `didFinishLaunchingWithOptions` method of `AppDelegate` (the values of `LicenseURL` and `LicenseKey` are generated when you bound the license in the console).

If your SDK version is earlier than 2.5.1, you can find `TELICENSECheck.h` in `XMagic.framework` ; if your SDK version is 2.5.1 or later, `TELICENSECheck.h` is in `YTCommonXMagic.framework` .



```
[TELICENSECheck setTELICENSE:LicenseURL key:LicenseKey completion:^(NSInteger authr
if (authresult == TELICENSECheckOk) {
    NSLog(@"Authentication successful");
} else {
    NSLog(@"Authentication failed");
}
}];
```

Authentication error codes:

Error Codes	Description
-------------	-------------

0	Successful.
-1	The input parameter is invalid; for example, the <code>URL</code> or <code>KEY</code> is empty.
-3	Download failed. Check the network settings.
-4	Unable to obtain any Tencent Effect authentication information from the local system, which may be caused by an I/O failure.
-5	The VCUBE TEMP license file is empty, which may be caused by an I/O failure.
-6	The JSON field in the <code>v_cube.license</code> file is incorrect. Please contact Tencent Cloud team for help.
-7	Signature verification failed. Please contact Tencent Cloud team for help.
-8	Decryption failed. Please contact Tencent Cloud team for help.
-9	The JSON field in <code>TELicense</code> is incorrect. Please contact Tencent Cloud team for help.
-10	The Tencent Effect authentication information parsed online is empty. Please contact Tencent Cloud team for help.
-11	Failed to write Tencent Effect SDK authentication information to the local file, which may be caused by an I/O failure.
-12	Download failed, and failed to parse local assets.
-13	Authentication failed.
Others	Please contact Tencent Cloud team for help.

Step 2. Load the SDK (`xMagic.framework`)

The following is the process of using the Tencent Effect SDK:

1. Load effect resources.



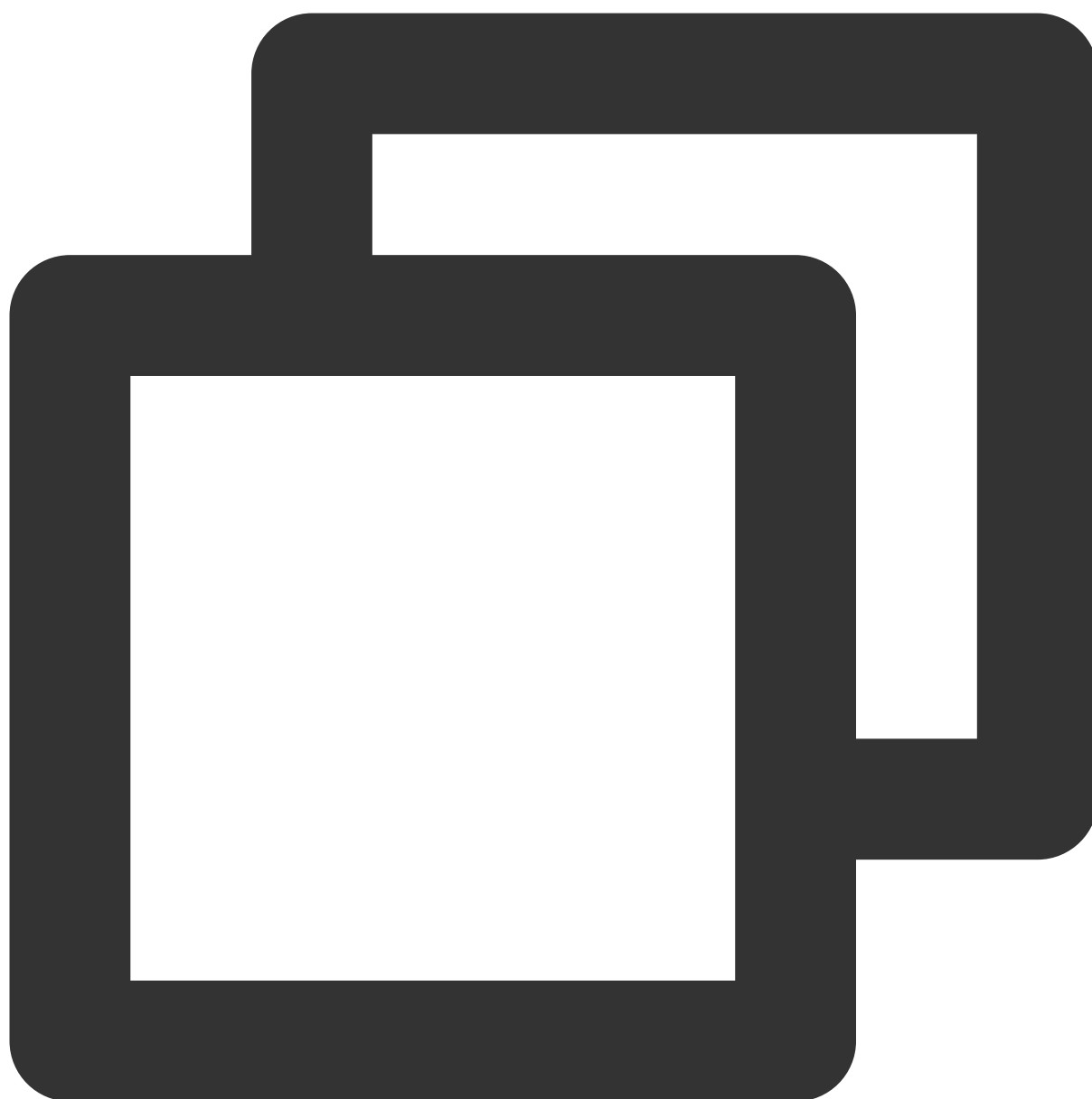
```
NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
    @"root_path":[[NSBundle mainBundle] bundlePath]
};
```

2. Initialize the Tencent Effect SDK.



```
initWithRenderSize:assetsDict: (XMagic)  
self.beautyKit = [[XMagic alloc] initWithRenderSize:previewSize assetsDict:assetsDi
```

3. The SDK processes each frame of data and returns the results.



```
process: (XMagic)
```

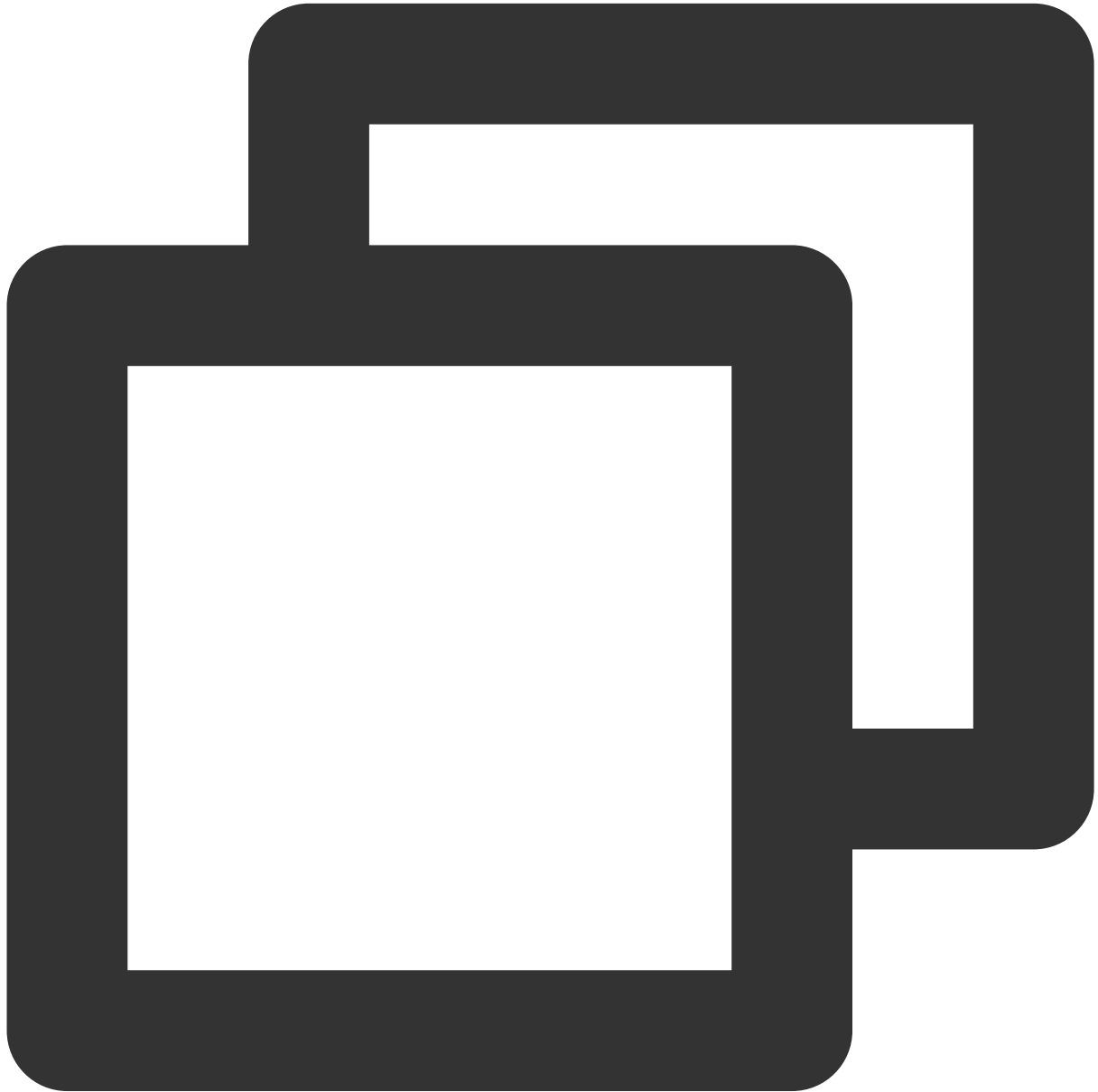


```
// Pass in frame data via the camera callback
- (void)captureOutput:(AVCaptureOutput *)captureOutput didOutputSampleBuffer:(CMSampleBufferRef)inputSampleBuffer {
    // Get the raw data and process the rendering information for each frame
    - (void)mycaptureOutput:(AVCaptureOutput *)captureOutput didOutputSampleBuffer:(CMSampleBufferRef)inputSampleBuffer {
        // Use the CPU to process the data
        - (YTPProcessOutput*)processDataWithCpuFuc:(CMSampleBufferRef)inputSampleBuffer;

        // Use the GPU to process the data
        - (YTPProcessOutput*)processDataWithGpuFuc:(CMSampleBufferRef)inputSampleBuffer;
```

```
// The data processing API of the Tencent Effect SDK
/// @param input: Input the data to be processed
/// @return: Output the processed data
- (YTProcessOutput* _Nonnull)process:(YTProcessInput * _Nonnull)input;
```

4. Release the Tencent Effect SDK.



```
deinit (XMagic)
// Call this API when you need to release the resources of the SDK
[self.beautyKit deinit]
```

Note

After completing the above steps, you can control the display timing and other device environment parameters as needed.

FAQs

Question 1. What should I do in case of the compilation error "unexpected service error: build aborted due to an internal error: unable to write manifest to-xxxx-manifest.xcbuild': mkdir(/data, S_IRWXU | S_IRWXG | S_IRWXO): Read-only file system (30):"?

1. Go to **File > Project settings > Build System** and select **Legacy Build System**.
2. For Xcode 13.0++, you need to select **File > Workspace Settings > Do not show a diagnostic issue about build system deprecation**.

Question 2. What should I do if the error "Xcode 12.X compilation: Building for iOS Simulator, but the linked and embedded framework '.framework'..." occurs when I compile my iOS project after importing the resources?

Go to **Build Settings > Build Options**, change **Validate Workspace** to **Yes**, and click **Run**.

Note

If you change **Validate Workspace** back to **No** after compilation, you can still run your project successfully.

Question 3. What should I do if the filter settings don't take effect?

Check the value you set (value range: 0-100). The effect may not be obvious if the value is too small.

Question 4. What should I do if a `dSYM` generation error occurs when I compile the iOS demo project?



```
PhaseScriptExecution CMake\\ PostBuild\\ Rules build/XMagicDemo.build/Debug-iphoneo
cd /Users/zhenli/Downloads/xmagic_s106
/bin/sh -c /Users/zhenli/Downloads/xmagic_s106/build/XMagicDemo.build/Debug-iph

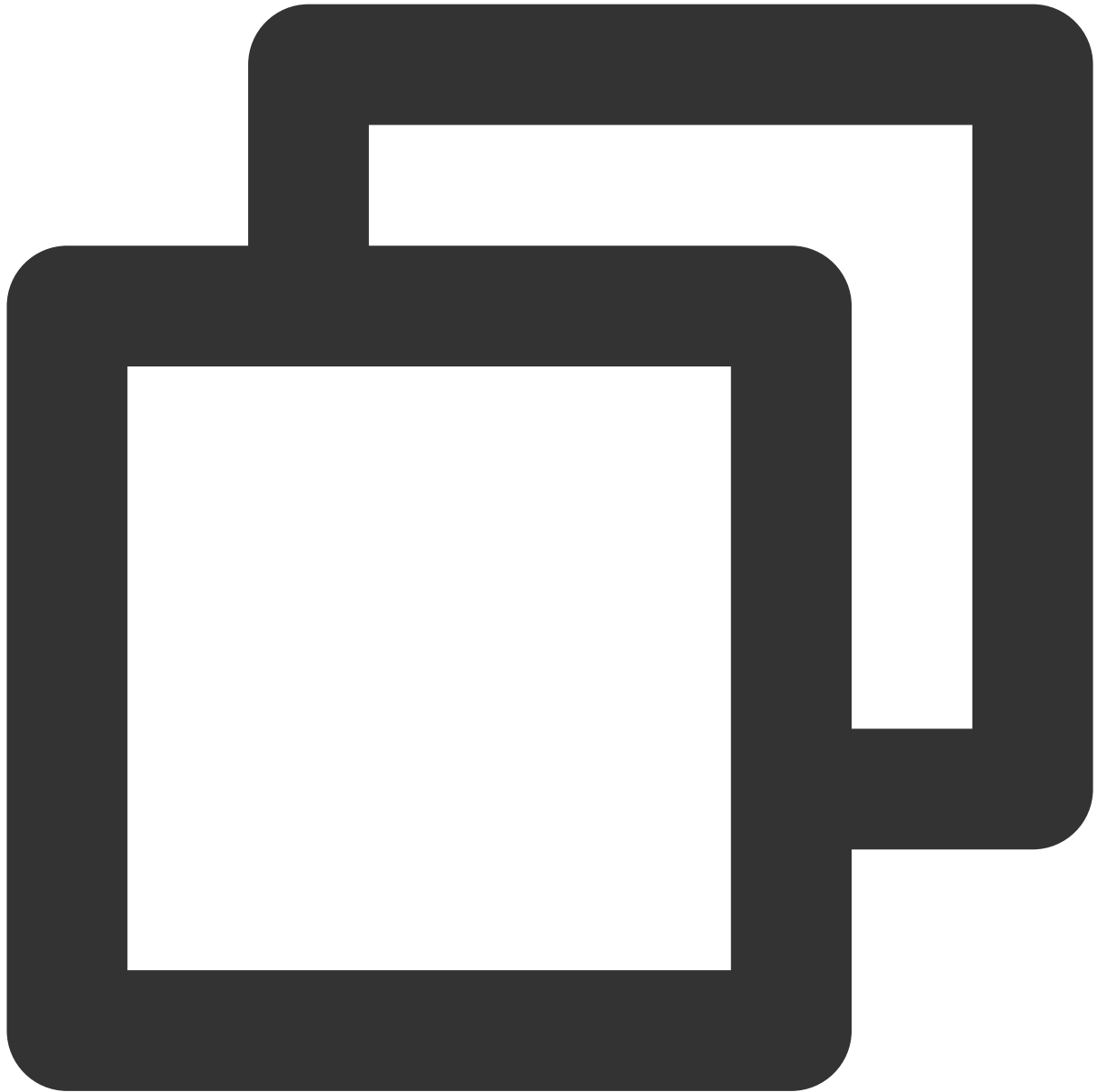
Command /bin/sh failed with exit code 1
```

Cause: Failed to sign `libpag.framework` and `Masonry.framework` again.

Solution:

1.1 Open `demo/copy_framework.sh` .

1.2 Use the following command to check the absolute path of the local `cmake` . Replace `$(which cmake)` with the absolute path of `cmake` .



```
which cmake
```

1.3 Replace all `Apple Development:` with your own signature.

Android

Last updated : 2024-08-08 18:08:43

Developer Environment Requirements

Android 4.4 (SDK API level 19) or later versions. Android 7.0 (SDK API level 24) or later versions are recommended.
Android Studio 3.5 or later versions.

Demo Project: TEBeauty_API_Example

Clone the [demo project](#) from GitHub, where the TEBeautyDemo is a demo project with UI, and the TEBeauty_API_Example is a demo project without UI.

Follow the instructions in the TEBeauty_API_Example/README Document to run TEBeauty_API_Example, and then refer to this document for detailed steps to integrate the SDK with UI.

Integration Method

Note: In the demo project on GitHub, the SDK is integrated using Maven.

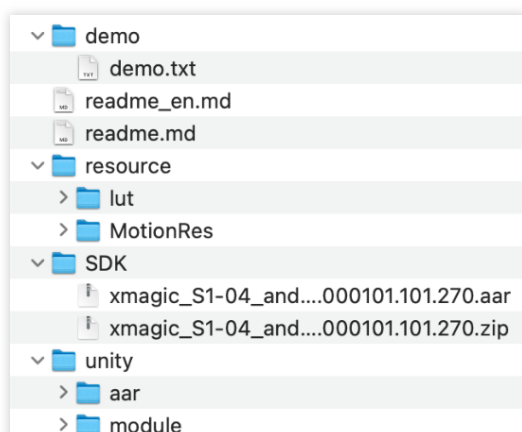
Manual Integration (Built-in Resources)

Manual Integration (Dynamic Resource Download)

Maven Integration

Downloading SDK

[Download the SDK](#) and decompress it. The directory structure is as follows:



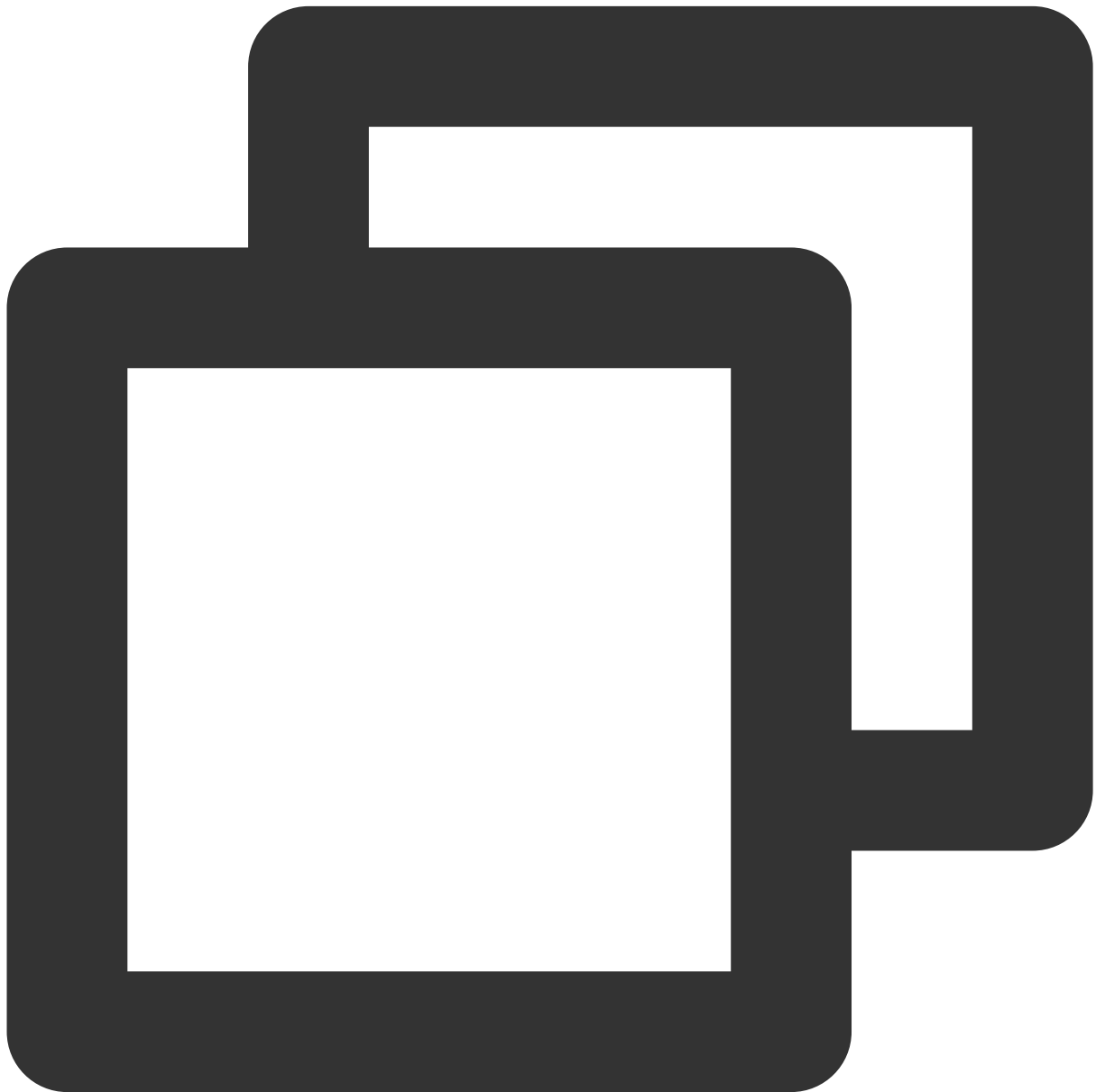
Integration

Copy the `xmagic-xxxx.aar` file from the `SDK` folder to your project's `libs` directory.

Copy the `lut` from the `resource` folder to the project's `../src/main/assets` directory. If there are resources in the `MotionRes` folder under `resource`, also copy this folder to the `../src/main/assets` directory.

Importing Method

Open the app module's `build.gradle` and add the dependency reference:



```
android{  
    ...  
}
```

```
defaultConfig {
    applicationId "Modify it with the package name bound to the authorized lice
    ....
}
packagingOptions {
    pickFirst '**/libc++_shared.so'
}
}

dependencies{
    ...
    compile fileTree(dir: 'libs', include: ['*.jar','*.aar'])//add *.aar
}
```

Note:

Note:

The following dependencies need to be added to the project:



```
dependencies{
    implementation 'com.google.code.gson:gson:2.8.2'
    //Needs to be added for v2.6.0 to 3.1.0.2.
    implementation 'androidx.exifinterface:exifinterface:1.3.3'
    //Needs to be added for v3.5.0 or later versions.
    implementation 'com.tencent.tav:libpag:4.3.33-noffavc'
}
```

If you want to use another version of pag, please click [here](#) to view.

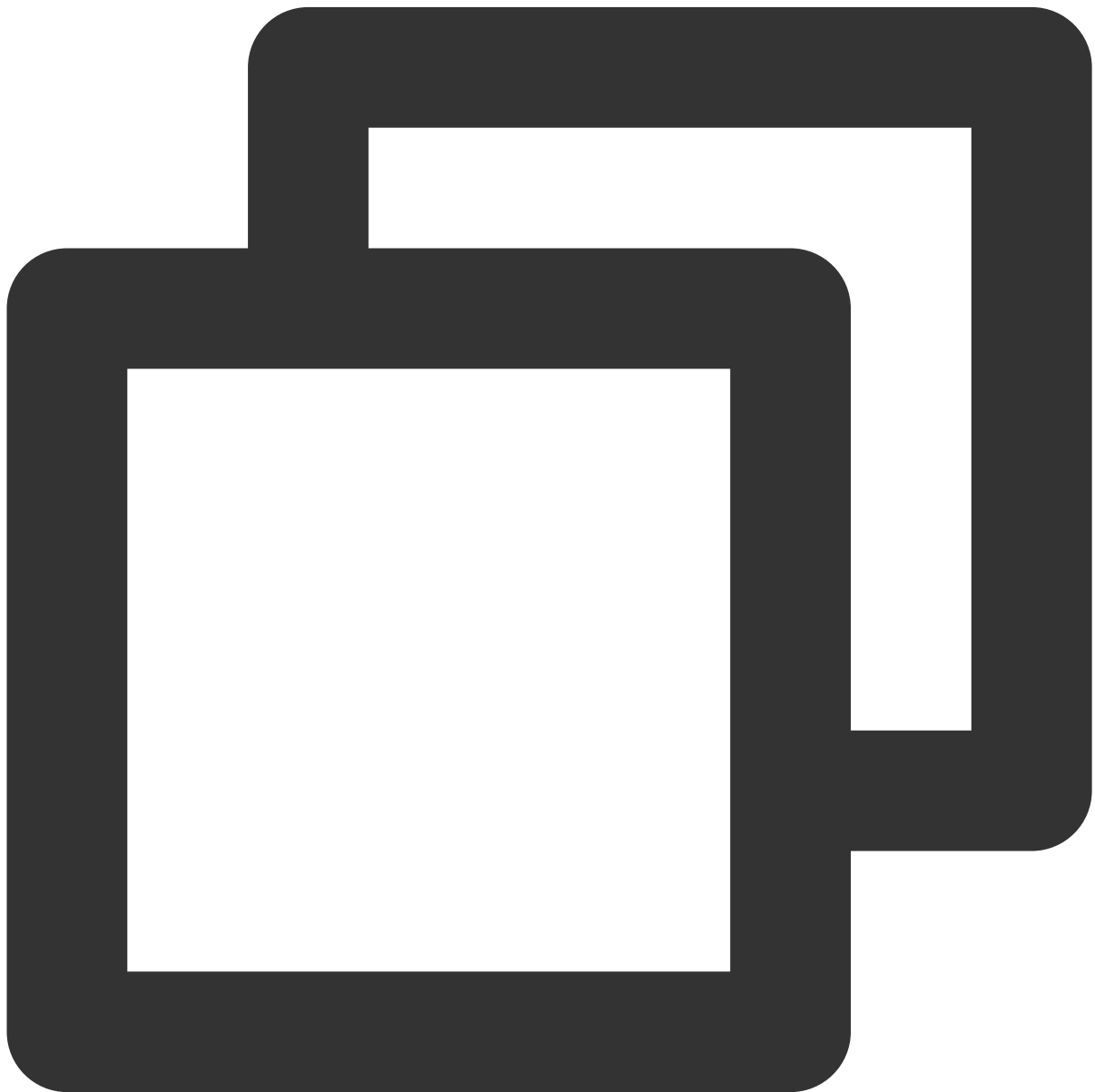
Dynamically Downloading Assets, .so Libraries, and Dynamic Effect Resources Guide

To reduce the package size, you can change the download mode of the assets resources, .so libraries, and dynamic effect resources MotionRes (some basic SDKs do not have dynamic effect resources) required by the SDK to the online mode. After successful download, provide the paths of the above files to the SDK through setting.

You can use your existing download service, but we recommend you use the download logic of the demo. For detailed directions on implementing dynamic download, see [Reducing SDK Size \(Android\)](#).

The Tencent Effect SDK has been released to the mavenCentral repository, and you can configure Gradle to download updates automatically.

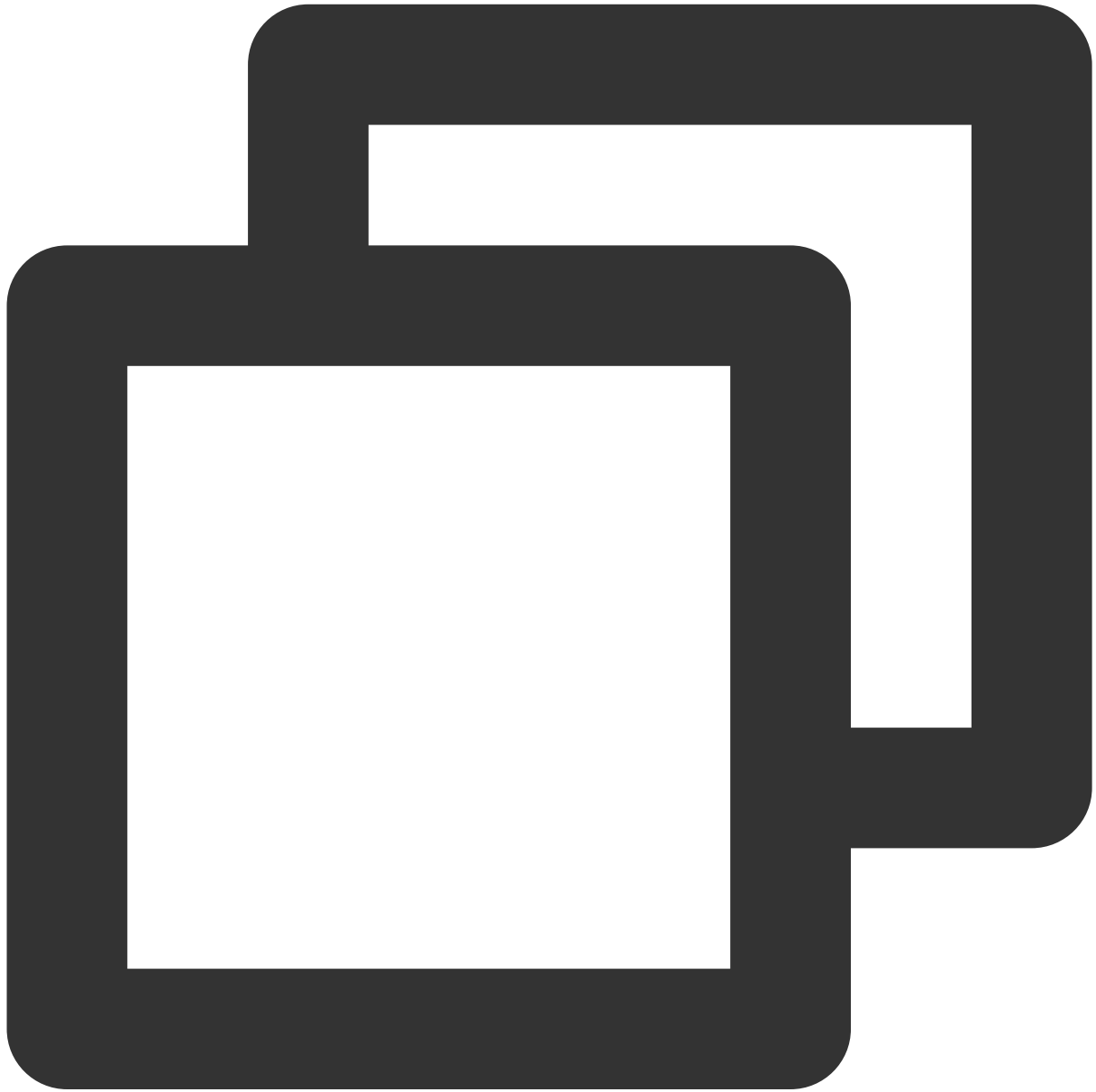
1. Add the Tencent Effect SDK dependency in the dependencies section.



```
dependencies {
```

```
//For example, the S1-04 package is as follows:  
implementation 'com.tencent.mediacloud:TencentEffect_S1-04:version number'  
//The version number can be found on the Version History page on the official website  
//But please note: using latest.release will always keep your SDK version up to date
```

2. In defaultConfig, specify the CPU architecture for the app.



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

Note

Currently, the Tencent Effect SDK supports armeabi-v7a and arm64-v8a.

3. click



<2> to automatically download the SDK and integrate it into the project.

4. If your package includes dynamic effect and filter features, you need to download the corresponding resources from the [SDK Download Page](#) and place the dynamic effect and filter materials in the following directories under your project:

Dynamic effect: `../assets/MotionRes`

Filter: `../assets/lut`

Maven Address for Each Package

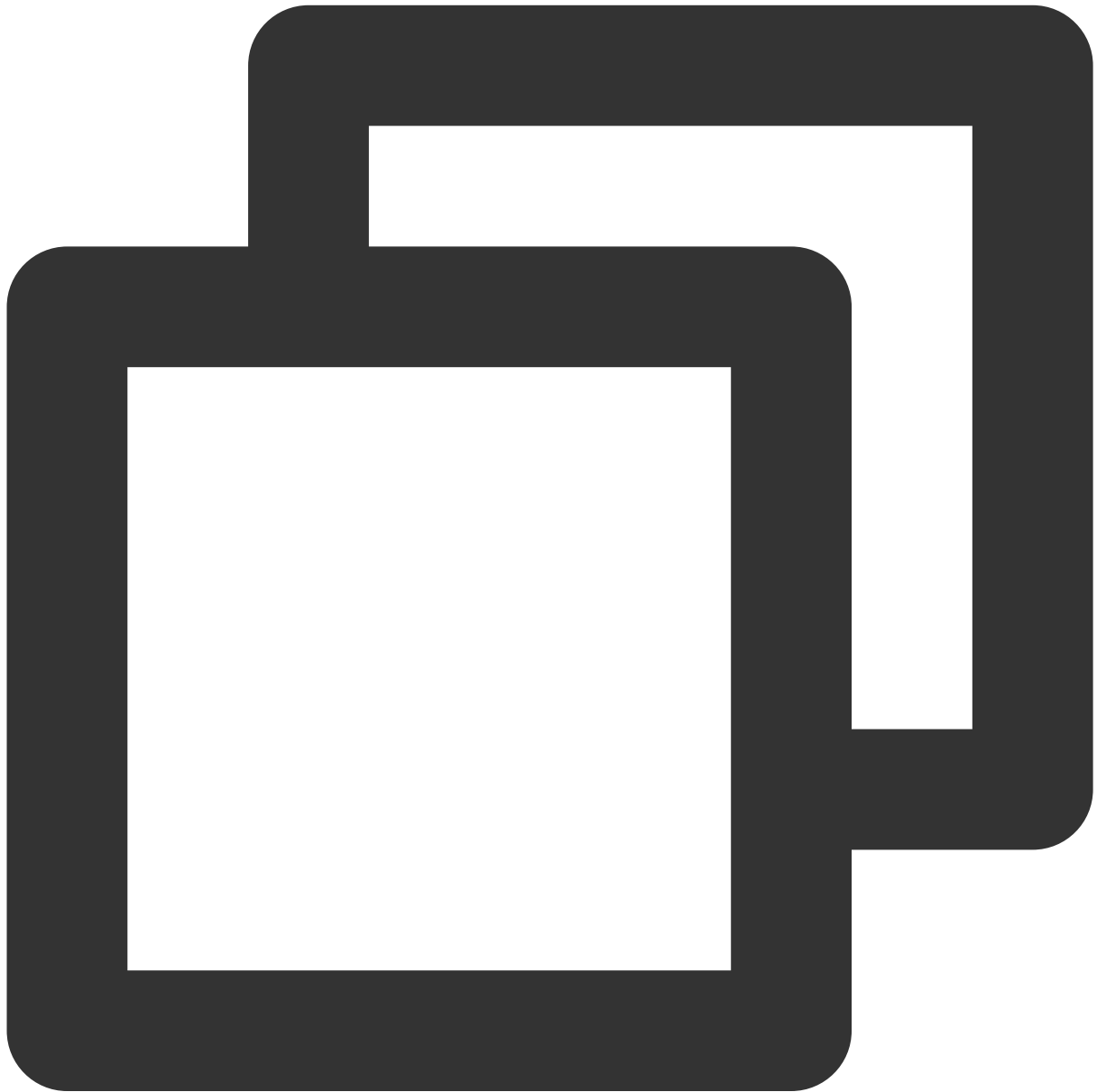
Version	Maven Address
A1 - 01	implementation 'com.tencent.mediacloud:TencentEffect_A1-01:version number'
A1 - 02	implementation 'com.tencent.mediacloud:TencentEffect_A1-02:version number'
A1 - 03	implementation 'com.tencent.mediacloud:TencentEffect_A1-03:version number'
A1 - 04	implementation 'com.tencent.mediacloud:TencentEffect_A1-04:version number'
A1 - 05	implementation 'com.tencent.mediacloud:TencentEffect_A1-05:version number'
A1 - 06	implementation 'com.tencent.mediacloud:TencentEffect_A1-06:version number'
S1 - 00	implementation 'com.tencent.mediacloud:TencentEffect_S1-00:version number'
S1 - 01	implementation 'com.tencent.mediacloud:TencentEffect_S1-01:version number'
S1 - 02	implementation 'com.tencent.mediacloud:TencentEffect_S1-02:version number'
S1 - 03	implementation 'com.tencent.mediacloud:TencentEffect_S1-03:version number'
S1 - 04	implementation 'com.tencent.mediacloud:TencentEffect_S1-04:version number'

SDK Usage Process

Step 1: Authenticate

1. Apply for authorization to obtain the license URL and license key. Please refer to the [License Guide](#).

2. Set the URL and key in the initialization code of the related business module to **trigger license download to avoid downloading temporarily before use**. For example, in our demo project, the download is triggered in the onCreate method of the application. However, we do not recommend triggering it here in your project because network permission might not be available or there could be a high rate of network failure at this time. Please choose a more appropriate time to trigger the license download.



```
//If you only want to trigger the download or update the license without caring about  
TELICENSECheck.getInstance().setXMagicLicense(context, URL, KEY, null);
```

3. Perform authentication before actually using the beauty feature:



```
TELICENSECHECK.getInstance().setTELICENSE(context, URL, KEY, new TELICENSECHECKLIST

@Override
public void onLicenseCheckFinish(int errorCode, String msg) {
    //Note: This callback may not be on the calling thread.
    if (errorCode == TELICENSECHECK.ERROR_OK) {
        // Authentication succeeded.
    } else {
        // Authentication failed.
    }
}
```

```
});
```

Note:

Under normal circumstances, as long as the app can connect to the internet normally and the license URL can be accessed normally in the user's region, the authentication process can be completed using the above code, and the license information will be cached locally. Therefore, you do not need to embed the license file into the project. But in special situations, the app may consistently fail to connect to the internet or be unable to access the license URL, making authentication impossible. To address this, you can open your license URL in a browser, download the license file, place it in the project's `src/main/assets` directory, and name it `v_cube.license`. This way, even if online authentication fails, local authentication can be performed.

If using the built-in license scheme, ensure that the license file in the package is always the latest version. For example, after renewing the license, download the updated license file and place it in the package.

Authentication error code description:

Error Code	Description
0	Success
-1	Invalid input parameters, such as empty URL or KEY
-3	Download failure. Please check your network settings.
-4	TE authorization information read locally is empty, possibly due to I/O failure.
-5	Content of the VCUBE TEMP license file is empty, possibly due to I/O failure.
-6	JSON fields in the v_cube.license file are incorrect. Please contact the Tencent Cloud team to deal with it.
-7	Signature verification failed. Please contact the Tencent Cloud team to deal with it.
-8	Decryption failed. Please contact the Tencent Cloud team to deal with it.
-9	JSON fields in the TELicense field are incorrect. Please contact the Tencent Cloud team to deal with it.
-10	The TE authorization information parsed from the network is empty. Please contact the Tencent Cloud team to deal with it.
-11	Failed to write TE authorization information to a local file, possibly due to I/O failure.
-12	Download failed, and parsing local assets also failed.
-13	Authentication failed. Please check if the .so file is in the package or if the .so path is set correctly.
3004/3005	Invalid authorization. Please contact the Tencent Cloud team to deal with it.

3015	Bundle ID/Package Name mismatch. Check if the Bundle ID/Package Name used by your app is consistent with the applied one, and ensure you are using the correct license file.
3018	The license file has expired. You need to apply for a renewal from Tencent Cloud.
Others	Please contact the Tencent Cloud team to deal with it.

Step 2: Copy resources

The resource files mentioned here consist of two parts:

The SDK model files, located in the `assets` directory of the SDK's AAR package.

Filter and dynamic effect resource files, located in the `assets` directory of the demo project, named `lut` and `MotionRes` respectively.

Before using the beauty effect, you need to copy the above resources to the app's private directory. If the SDK version is not updated, you only need to copy it once. After a successful copy, you can record it in the SharedPreferences of the app, so that you do not need to copy it again next time. For details, refer to `TEMenuActivity.java` in the demo project.



```
String resPath = new File(getFilesDir(), AppConfig.getInstance().getBeautyFileDirName());
if (!resPath.endsWith(File.separator)) {
    resPath = resPath + File.separator;
}
AppConfig.resPathForSDK = resPath;
AppConfig.lutFilterPath = resPath + "light_material/lut";
AppConfig.motionResPath = resPath + "MotionRes";

new Thread(() -> {
    Context context = getApplicationContext();
```

```
int addResult = XmagicApi.addAiModeFilesFromAssets(context, AppConfig.resPathFo
Log.d(TAG, "copyRes, add ai model files result = " + addResult);

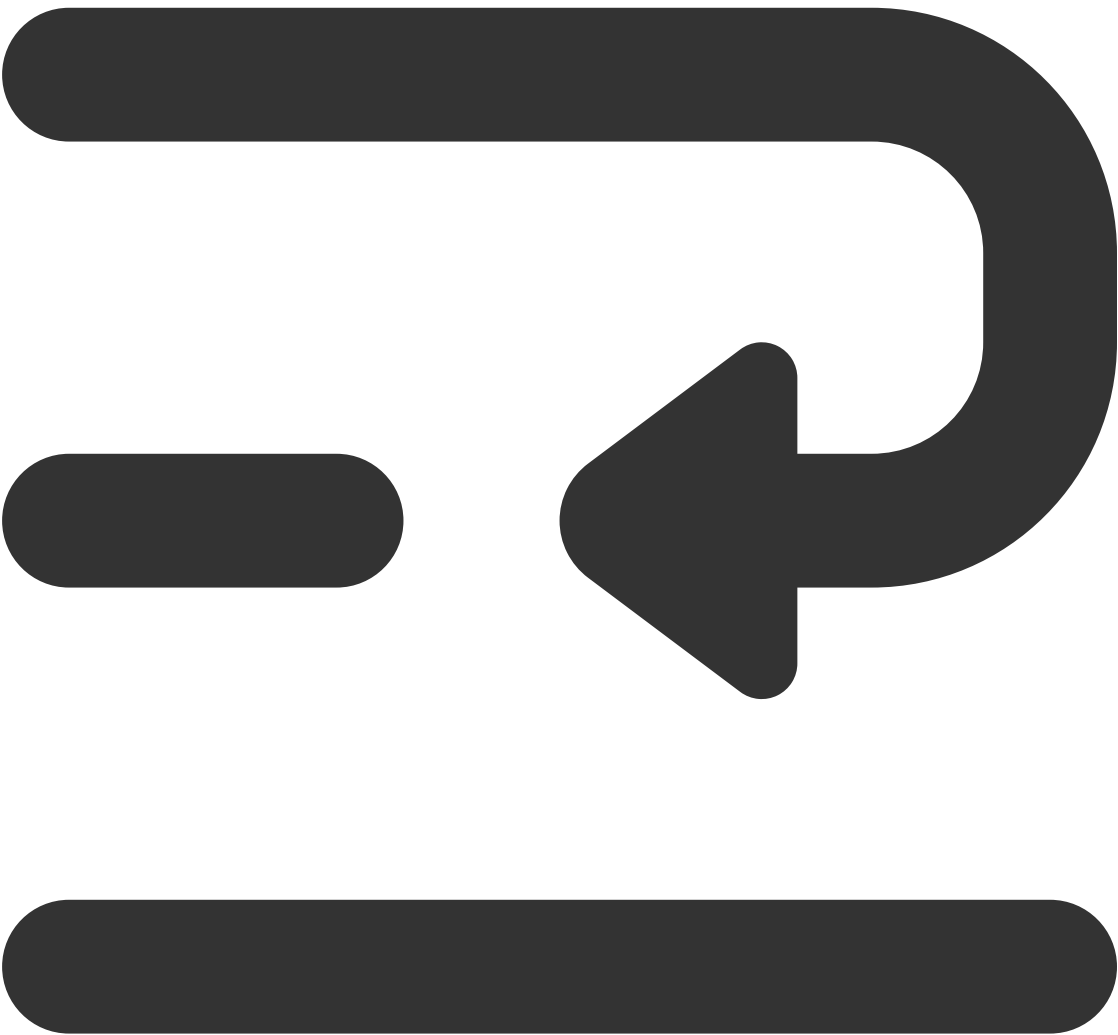
String lutDirNameInAsset = "lut";
boolean result = FileUtil.copyAssets(context, lutDirNameInAsset, AppConfig.lutF
Log.d(TAG, "copyRes, copy lut, result = " + result);

String motionResDirNameInAsset = "MotionRes";
boolean result2 = FileUtil.copyAssets(context, motionResDirNameInAsset, AppConf
Log.d(TAG, "copyRes, copy motion res, result = " + result2);
}).start();
```

Step 3: Initialize and use the SDK

1. (Optional) Quickly implement cameras.

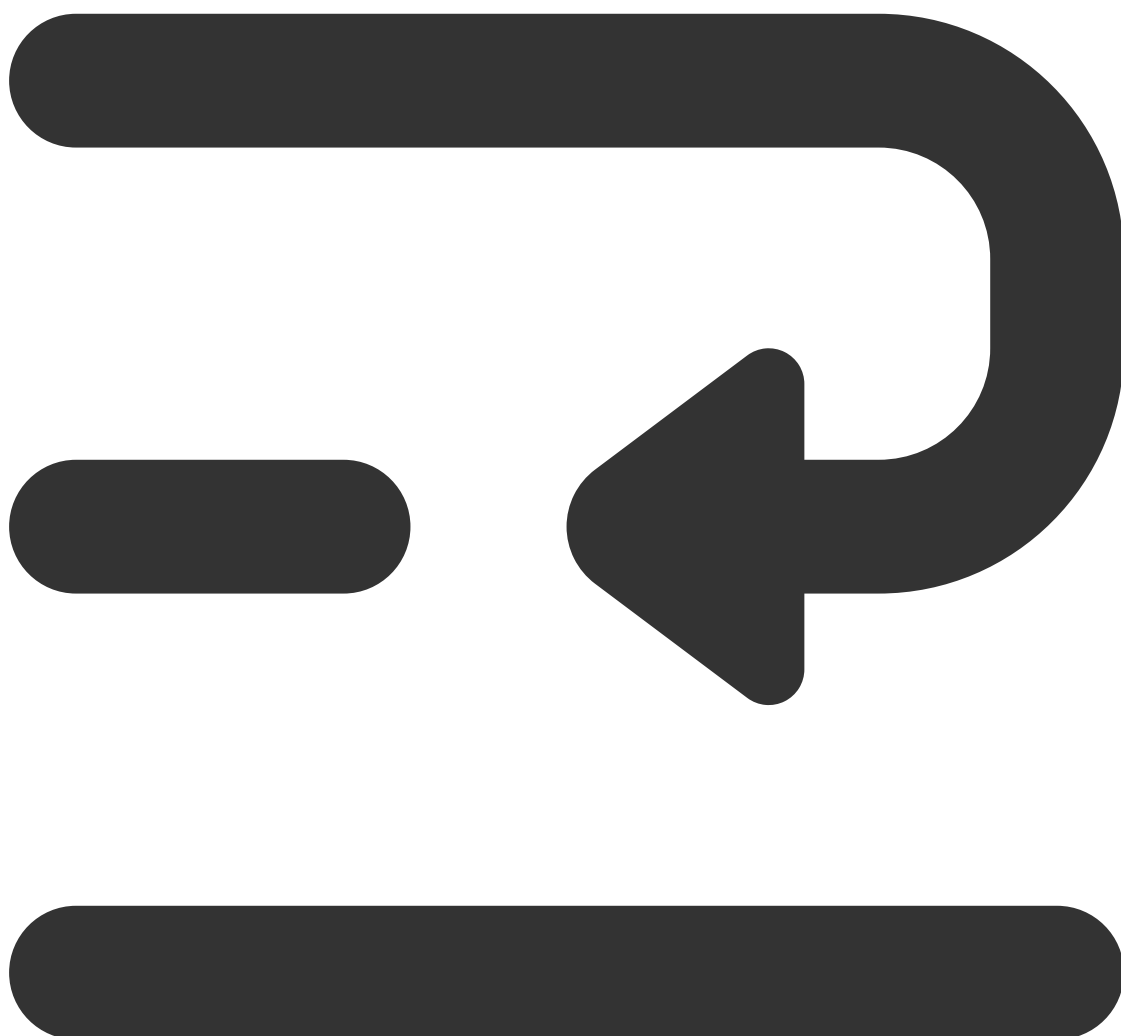
We assume you have already implemented the camera application, can start the camera normally, and can call back the camera's SurfaceTexture texture information to Activity for beauty effect processing, as shown below:





```
@Override
public int onCustomProcessTexture(int textureId, int textureWidth, int textureHeight) {
    // The beauty effect SDK processes the textureId here, adds beauty effects and so on.
    return 0;
}
```

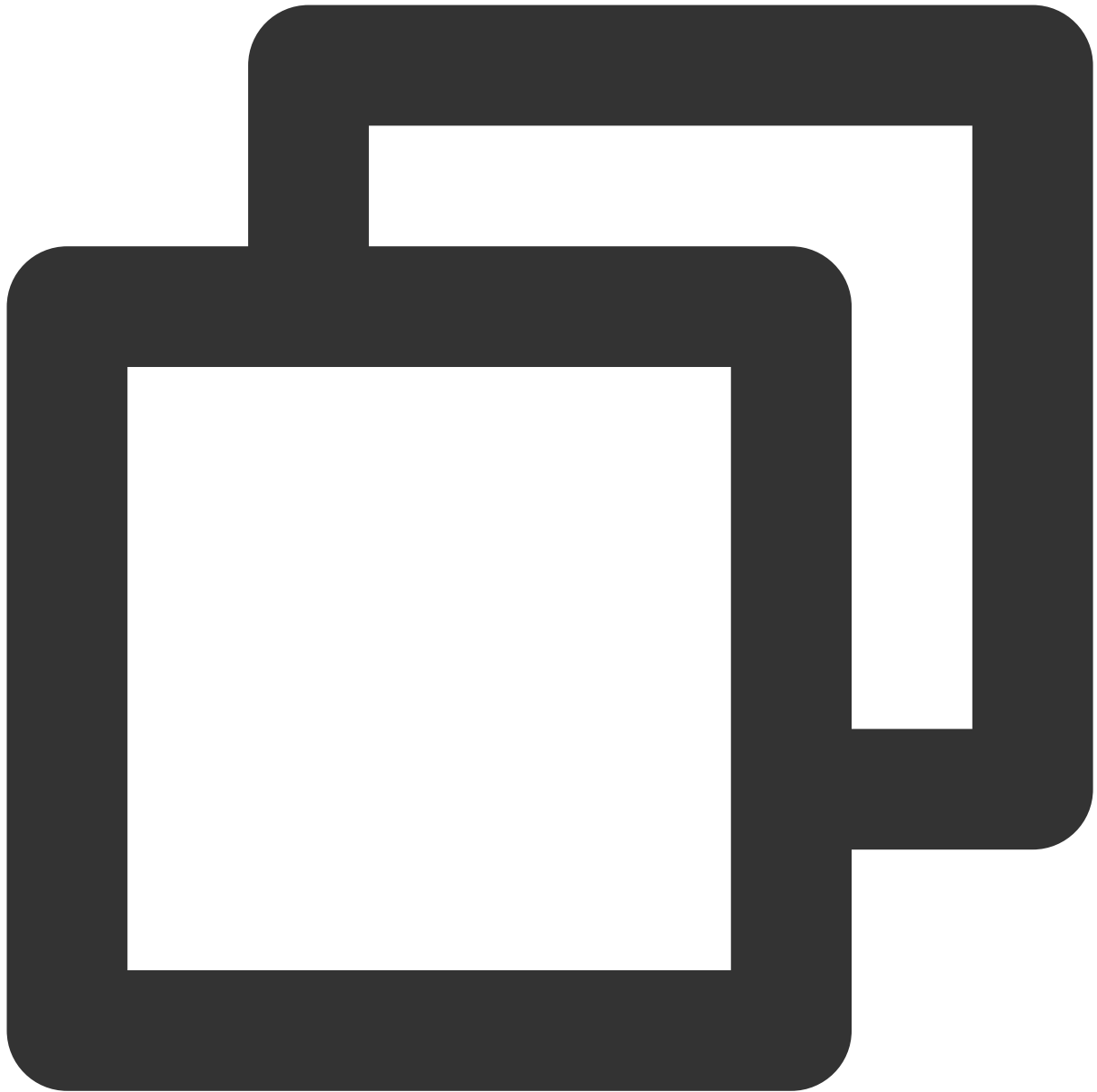
If you have not yet implemented the camera application, you can refer to `TECameraBaseActivity.java` in the demo project and use the `GLCameraXView` component to add it to your Activity's layout for quick camera preview:





```
<com.tencent.demo.camera.camerax.GLCameraXView  
    android:id="@+id/te_camera_layout_camerax_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:back_camera="false"  
    app:surface_view="false"  
    app:transparent="true" />
```

2. Initialize the beauty effect SDK. In the demo project, it is initialized in the Activity's `onCreate()` method:



```
//AppConfig.resPathForSDK is the resource path determined during the resource copy  
mXmagicApi = new XmagicApi(this, AppConfig.resPathForSDK);
```

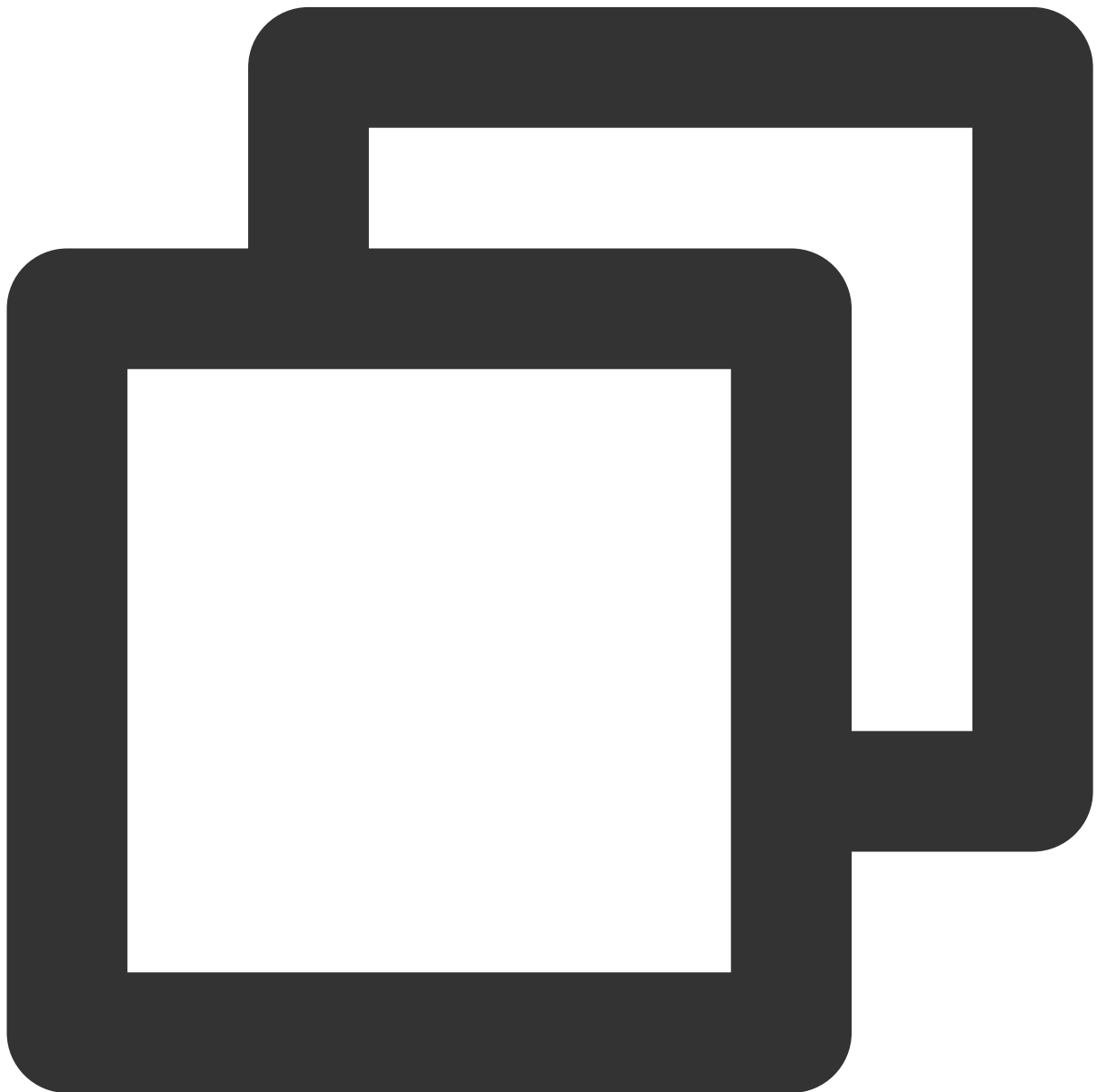
Parameter

Parameter	Meaning
Context context	Context
String resDir	Resource file directory. For details, refer to Step 2 .

OnXmagicPropertyErrorListener
errorListener

Optional. Callback function implementation class, for handling some error codes during the SDK initialization and usage process. For the meanings of error codes, refer to [API Documentation](#)

3. (Optional) Add a material prompt callback function (method callback might run in a child thread). Some materials will prompt the user to nod, extend the hand, and make a heart gesture. This callback is used to display similar prompts.



```
mXmagicApi.setTipsListener(new XmagicTipsListener() {
```

```
final XmagicToast mToast = new XmagicToast();
@Override
public void tipsNeedShow(String tips, String tipsIcon, int type, int duration) {
    mToast.show(MainActivity.this, tips, duration);
}

@Override
public void tipsNeedHide(String tips, String tipsIcon, int type) {
    mToast.dismiss();
}
});
```

4. The beauty effect SDK processes each frame of data and returns the corresponding processing results. For detailed information on the process method, see the [API Documentation](#).



```
@Override
public int onCustomProcessTexture(int textureId, int textureWidth, int textureHeight) {
    return mXmagicApi.process(textureId, textureWidth, textureHeight);
}
```

5. Set beauty effects or special effects.

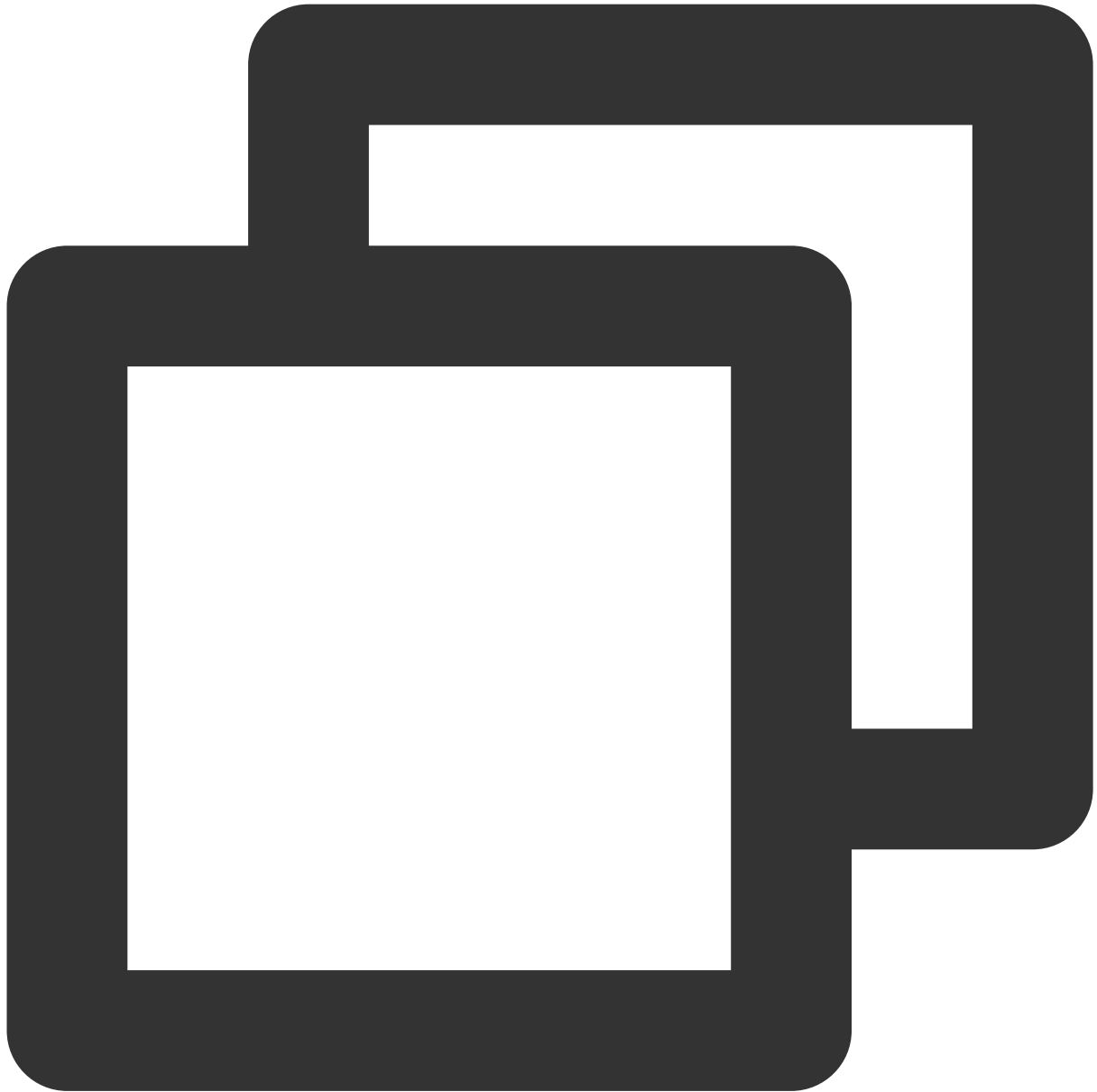
Use the `setEffect` method for **version 3.5.0 and later**. For detailed information, see the [API Documentation](#).

Use the `updateProperty` method for **version 3.3.0 and earlier**. For detailed information, see the [API Documentation](#).



```
//Use this method for version 3.5.0 and later.  
mXmagicApi.setEffect(String effectName, int effectValue, String resourcePath, Map<S  
  
// Available input parameter properties can be obtained from XmagicResParser.parseR  
// Use this method for version 3.3.0 and earlier.  
@Deprecated  
mXmagicApi.updateProperty(XmagicProperty<?> p);
```

6. Lifecycle method `onResume` : It is recommended to call it in the `Activity`'s `onResume()` method. When called, it will resume the sound in the effects.



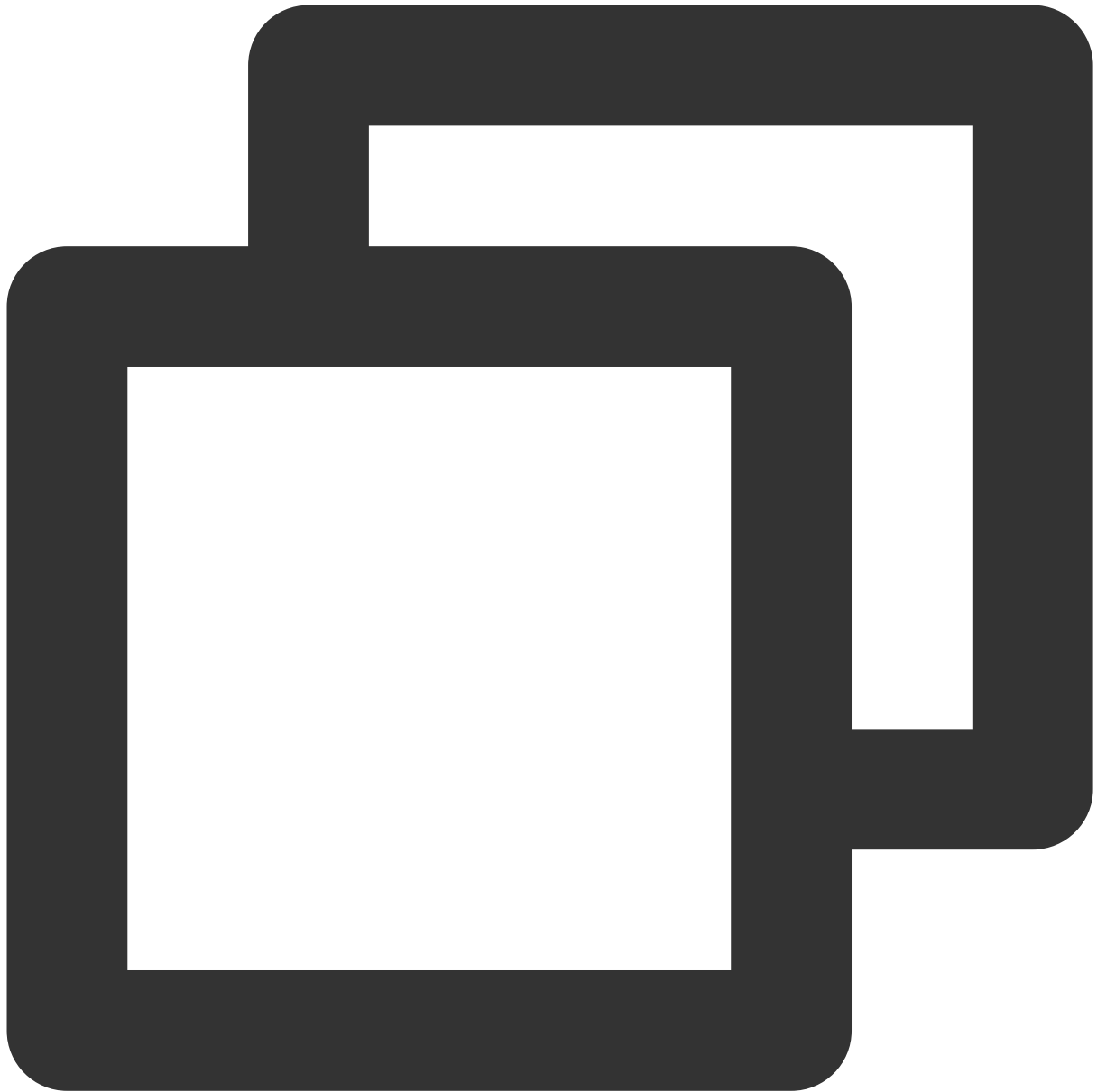
```
mXmagicApi.onResume();
```

7. Lifecycle method `onPause` : It is recommended to call it in the `Activity`'s `onPause()` method. When called, it will pause the sound in the effects.



```
mXmagicApi.onPause();
```

8. Beauty effect release SDK: It is called when the OpenGL environment is being terminated. **It needs to be called in the GL thread, and cannot be called in the main thread (Activity's onDestroy);** otherwise, it may cause resource leaks and result in a white or black screen after multiple entries and exits.

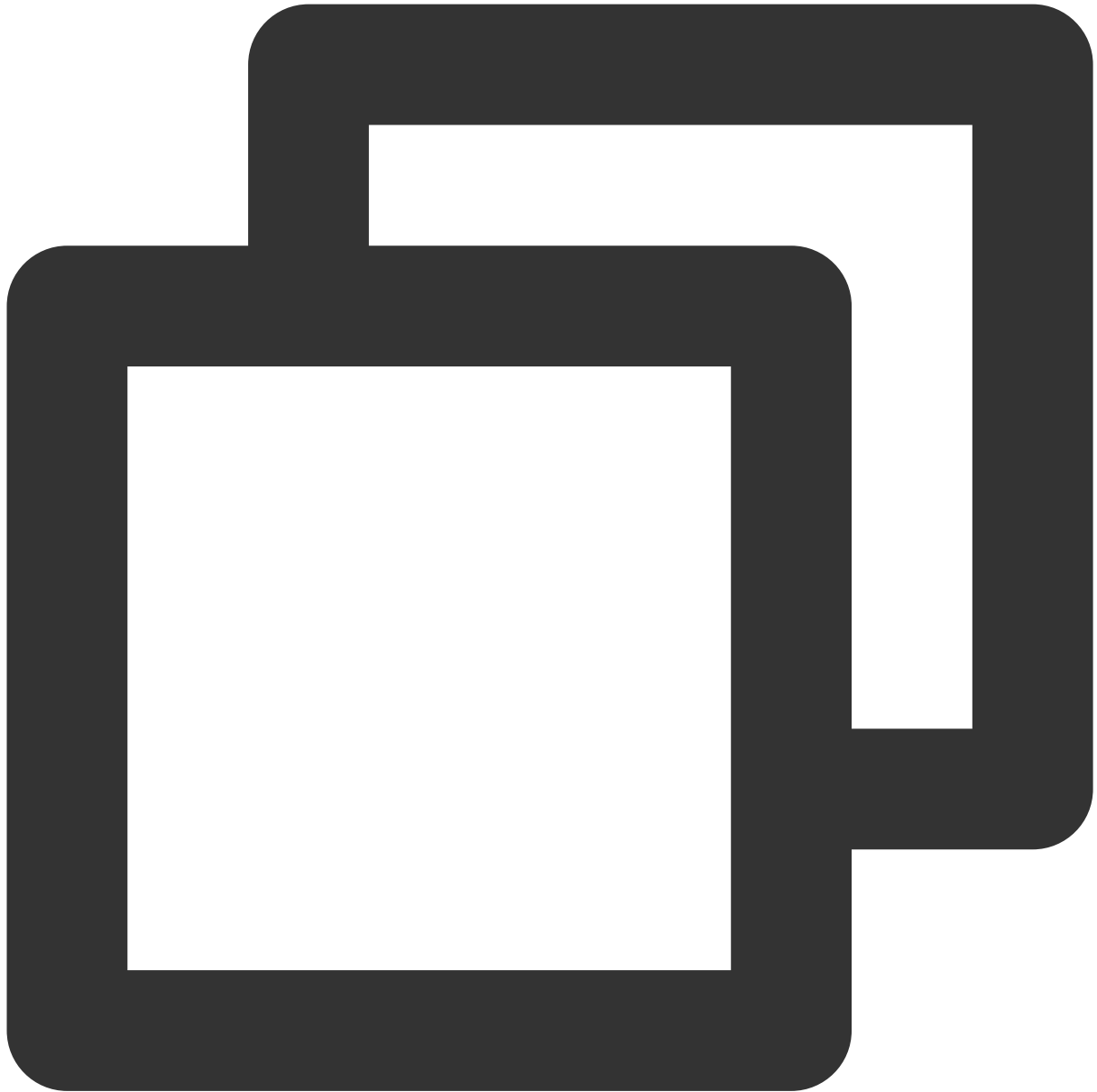


```
@Override
public void onGLContextDestroy() {
    mXmagicApi.onDestroy();
}
```

Step 4: Obfuscation configuration

If you enable compile optimization (setting `minifyEnabled` to `true`) when packaging the release, it will trim some code that is not called in the Java layer. This code may possibly be invoked by the native layer, thus causing the `no xxx method` exception.

If you have enabled such compile optimization, you should add these keep rules to prevent xmagic's code from being trimmed:



```
-keep class com.tencent.xmagic.** { *;}
-keep class org.light.** { *;}
-keep class org.libpag.** { *;}
-keep class org.extra.** { *;}
-keep class com.gyailib.**{ *;}
-keep class com.tencent.cloud.iai.lib.** { *;}
-keep class com.tencent.beacon.** { *;}
-keep class com.tencent.qimei.** { *;}
```

```
-keep class androidx.exifinterface.** { *; }
```

Attachment (SDK File Structure):

Note:

This table lists all the files used by the SDK. Some files may not be included in your package, but this will not affect the usage of that package's feature.

File Type			Description
assets	audio2exp		Avatar virtual human voice-driven model: If this feature is not used, the model is not needed.
	benchmark		Used for model adaptation.
	Light3DPlugin		Used for 3D stickers.
	LightBodyPlugin	LightBody3DModel.bundle	Used for 3D human skeleton points.
		LightBodyModel.bundle	Used for the body beauty feature.
	LightCore		SDK core model resources
	LightHandPlugin		Required for gesture stickers and hand-point capabilities.
	LightSegmentPlugin		Required for background segmentation capabilities.
	lut		Free filter resources
demo_XXX_android_XXXX			Demo project
jniLibs	libace_zplan.so		3D engine library
	libaudio2exp.so		Avatar virtual human voice-driven library: If this feature is not used, the library is not needed.
	libc++_shared.so		libc <u>shared.so is a shared library of the C standard library.</u> It provides a set of C <u>standard library functions and classes to</u>

		<u>support the development and operation of C programs.</u> It is widely used in the Android system and is an essential part of C <u>applications and libraries.</u> <u>If your project already includes a C shared library,</u> you can keep only one copy.
	liblight-sdk.so	Light SDK core library
	libpag.so	Animation file library that the light SDK depends on
	libtecodec.so	Codec library that the light SDK depends on
	libv8jni.so	JavaScript parsing library that the light SDK depends on
	libYTCommonXMagic.so	Used for license authentication
libs	xmagic-xxxx.aar	Beauty effect SDK .aar file
MotionRes	2dMotionRes	2D stickers
	3dMotionRes	3D stickers
	avatarRes	Avatar materials
	ganMotionRes	Fun stickers
	handMotionRes	Gesture stickers
	makeupRes	Makeup stickers
	segmentMotionRes	Background segmentation stickers
unity	aar	Bridging AAR required for the unity project
	module	Original project for the bridging AAR

Integrating Capabilities

Gesture Detection

Last updated : 2023-08-03 14:27:27

Overview

Input the camera's OpenGL texture and output real-time gesture detection data. You can use this data for further development.

Android Interface Description

Android Integration Guide

Integrate Tencent Effect SDK on Android, for details please refer to: [Integrate Tencent Effect SDK Independently](#).

Android Interface Invocation

1. Turn on the gesture detection feature switch (in XmagicApi.java)



```
public void setFeatureEnableDisable(String featureName, boolean enable);
```

Fill in featureName with XmagicConstant.FeatureName.HAND_DETECT, and set enable to true.

2.Set data callback (in XmagicApi.java)



```
void setAIDataListener(XmagicApi.OnAIDataListener aiDataListener)

public interface OnAIDataListener {
    void onFaceDataUpdated(List<TEFaceData> faceDataList);
    void onHandDataUpdated(List<TEHandData> handDataList);
    void onBodyDataUpdated(List<TEBodyData> bodyDataList);
    void onAIDataUpdated(String data);
}
```

onAIDataUpdated returns a JSON structured string data.

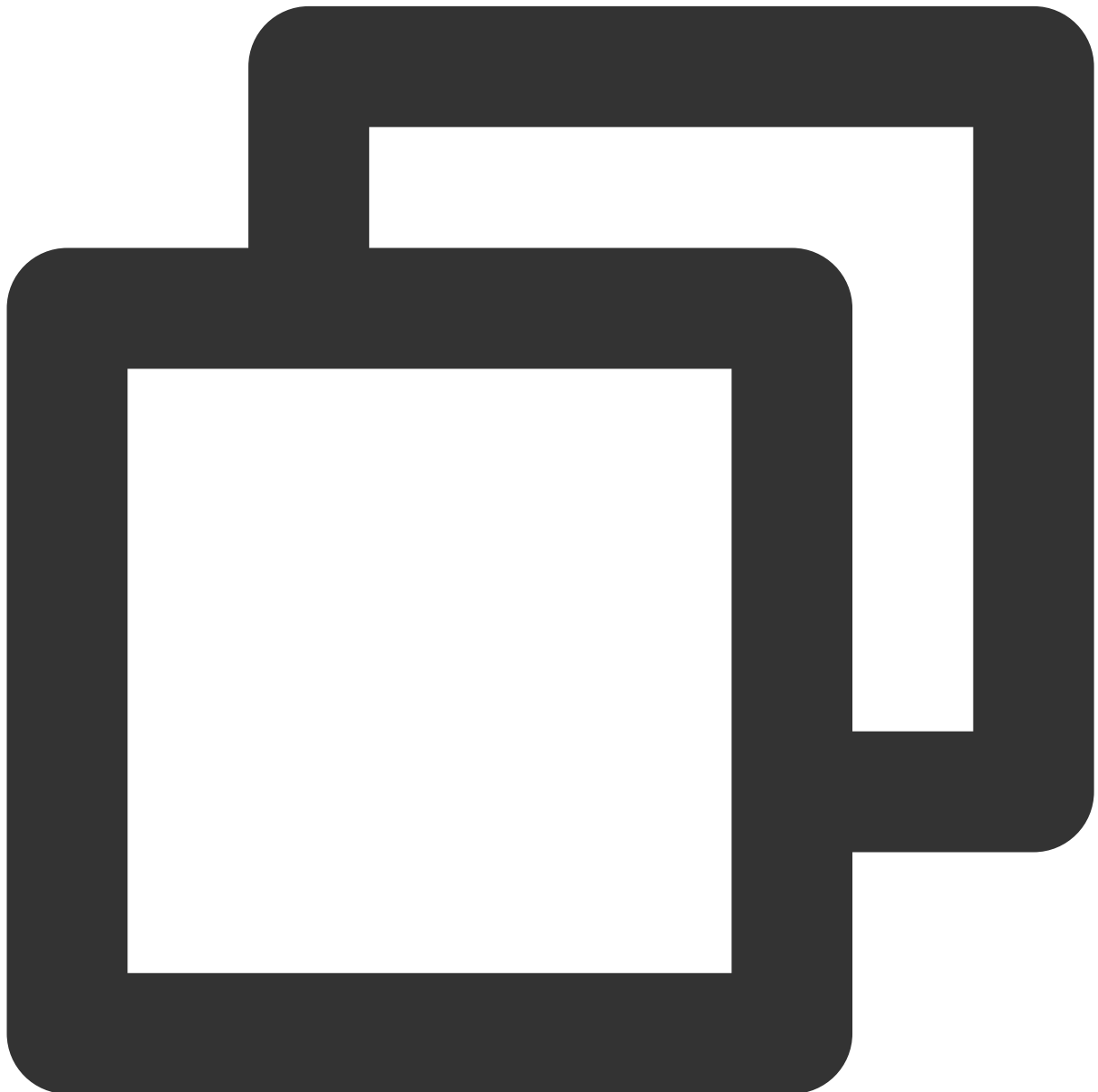
iOS Interface Description

iOS Integration Guide

Integrate Tencent Effect SDK on iOS, for details please refer to: [Integrate Tencent Effect SDK Independently](#).

iOS Interface Invocation

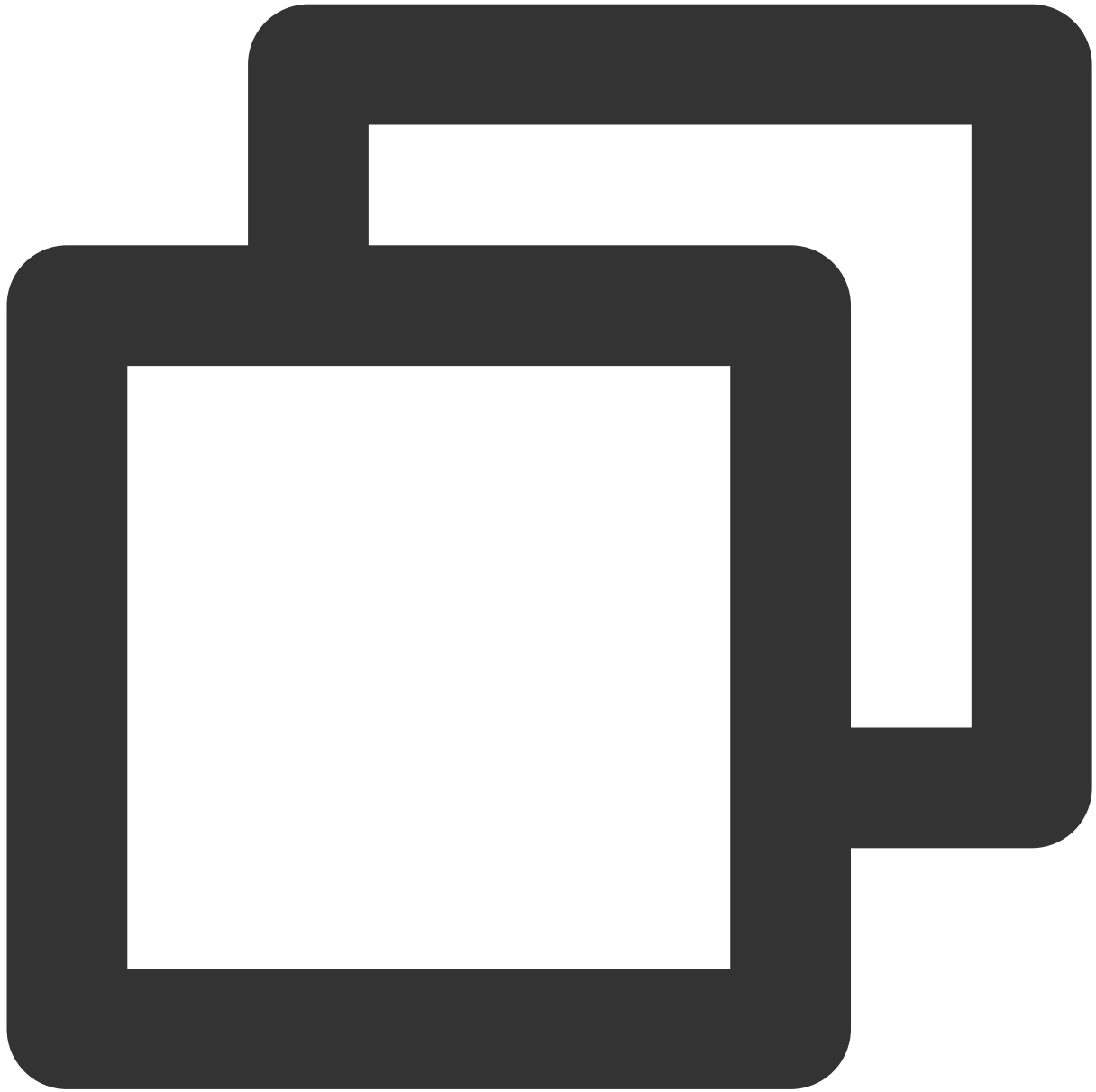
1. Turn on the gesture detection feature switch (in Xmagic.h)



```
- (void)setFeatureEnableDisable:(NSString *_Nonnull)featureName enable:(BOOL)enable
```


Fill in featureName with HAND_DETECT (can be imported from TEDefine.h), and set enable to true.

2.Set data callback (in Xmagic.h)



```
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;

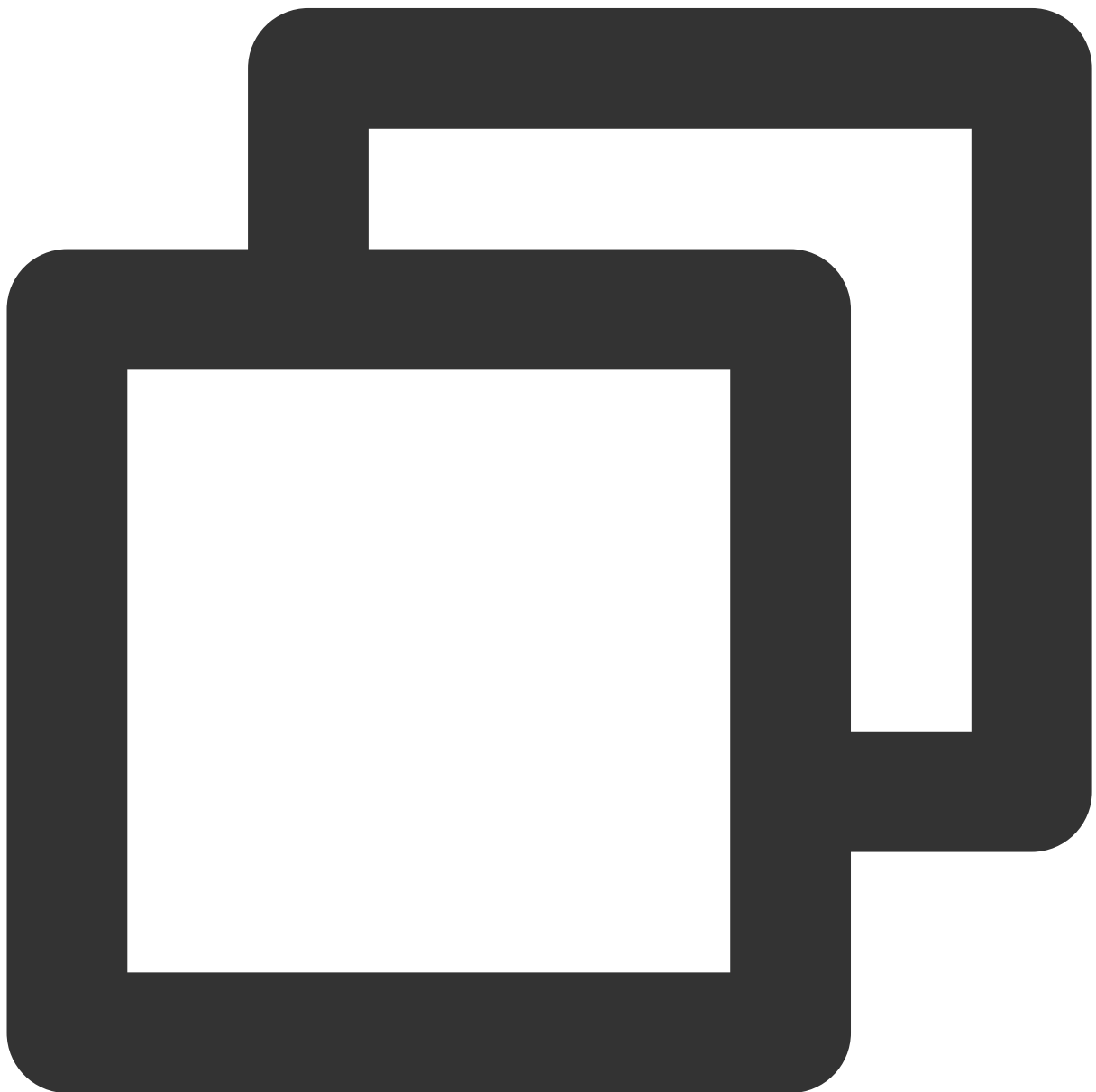
- (void)onAIEvent:(id)event
{
    NSDictionary *eventDict = (NSDictionary *)event;
    if (eventDict[@"ai_info"] != nil) {
        NSLog(@"ai_info %@", eventDict[@"ai_info"]);
    }
}
```

```
}  
}
```

eventDict["@ai_info"] is the returned JSON structured string data.

Callback JSON Data Description

In the callback JSON data, the gesture-related data is in "hand_info", and the format is as follows:

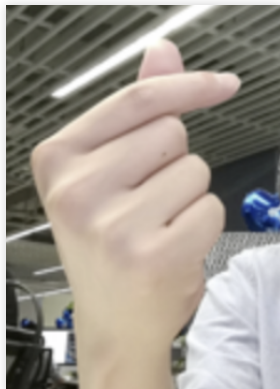
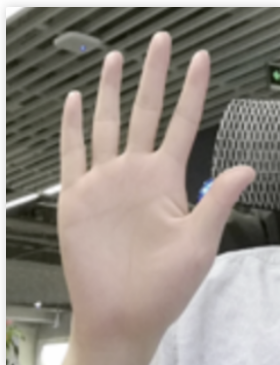


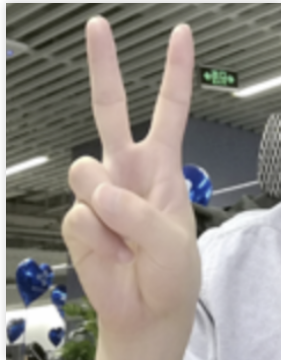
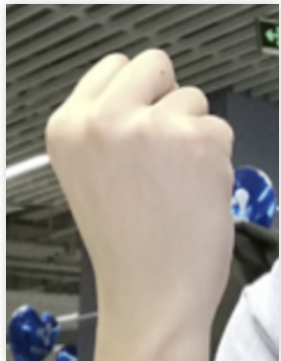
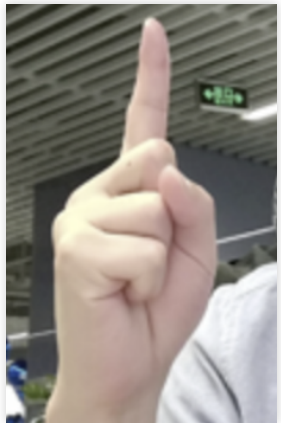
```
"hand_info": {  
  "gesture": "PAPER",  
  "hand_point_2d": [180.71888732910156, 569.2958984375, ... , 353.8714294433594,  
}
```


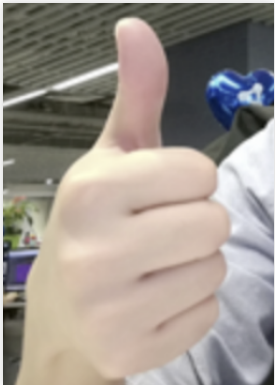
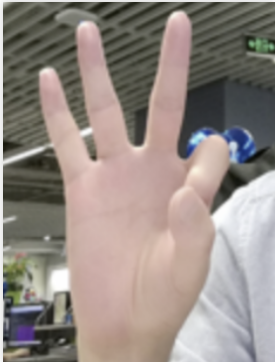
The explanations of each field in hand_info are as follows:

Field	Explanation
gesture	Gesture Type Name
hand_point_2d	Captured gesture data information

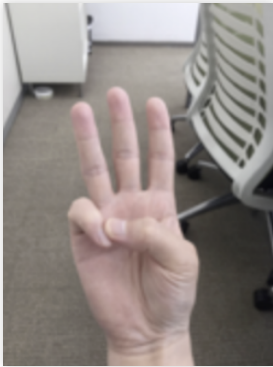
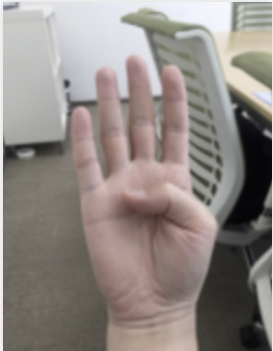
The following gestures are currently supported:

Order	Gesture	Type Name	Example Image
1	Heart	HEART	
2	Gestrue with number 5(open)	PAPER	
3	Gesture with number 2	SCISSOR	

			
4	Fist	FIST	
5	Gesture with number 1	ONE	
6	I love you	LOVE	

			
7	Thumb up	LIKE	
8	OK	OK	
9	Rock	ROCK	

			
10	Gesture with number 6	SIX	
11	Gesture with number 8	EIGHT	
12	Lift	LIFT	
13	Gesture with number 3	THREE	

			
14	Gesture with number 4	FOUR	

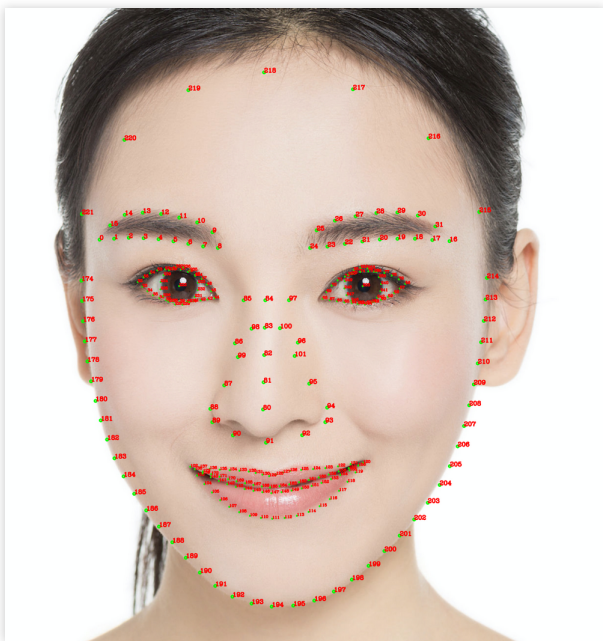
If it is an undetected gesture, the gesture type name is OTHER.

Facial Keypoint Detection

Last updated : 2023-05-18 16:30:41

Detects when a face is partially captured or concealed or when there are multiple faces; recognizes 256 facial keypoints.

Index Image for 256 Facial Keypoints



iOS API Description

Integration guide for iOS

For directions on how to integrate the Tencent Effect SDK for iOS, see the integration guide for [iOS](#).

Registering an Xmagic listener



```
/// @brief The SDK event listener  
/// @param listener The listener for SDK events, including AI events, tips, and res  
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;
```

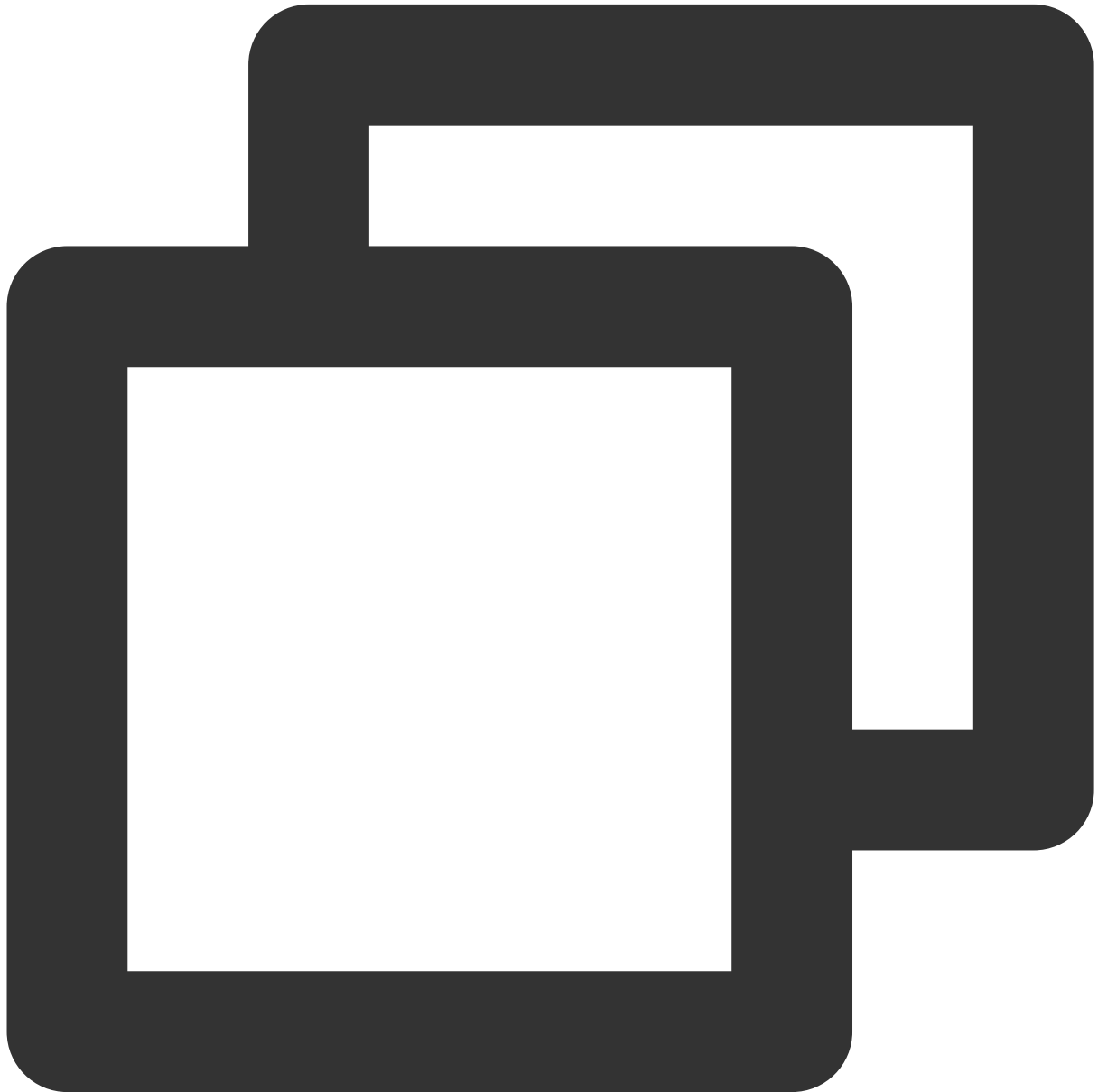
YTSDKEventListener callback description



```
#pragma mark - Event callback APIs
/// @brief SDK event callback APIs
@protocol YTSDKEventListener <NSObject>
/// @brief `YTDataUpdate` event callback
/// @param event: Callback in NSString* format
- (void)onYTDataEvent:(id _Nonnull)event;
/// @brief AI event callback
/// @param event: Callback in dict format
- (void)onAIEvent:(id _Nonnull)event;
/// @brief Tip event callback
/// @param event: Callback in dict format
```

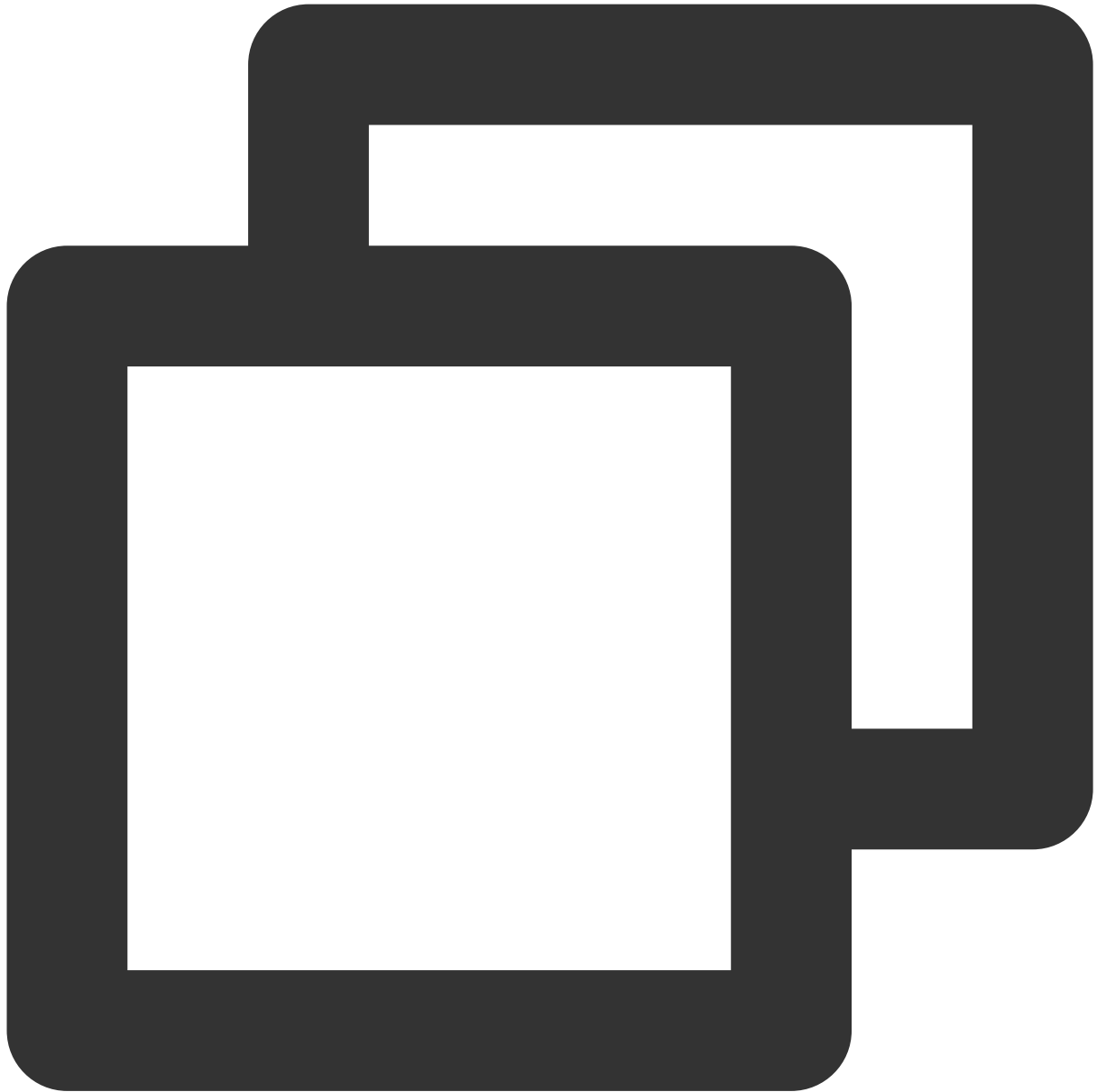
```
- (void)onTipsEvent:(id _Nonnull)event;  
/// @brief Resource pack event callback  
/// @param event: Callback in string format  
- (void)onAssetEvent:(id _Nonnull)event;  
@end
```

After the callbacks are configured successfully **on Version 2.6.0 and earlier versions**, the SDK will send a callback of facial data for each video frame.



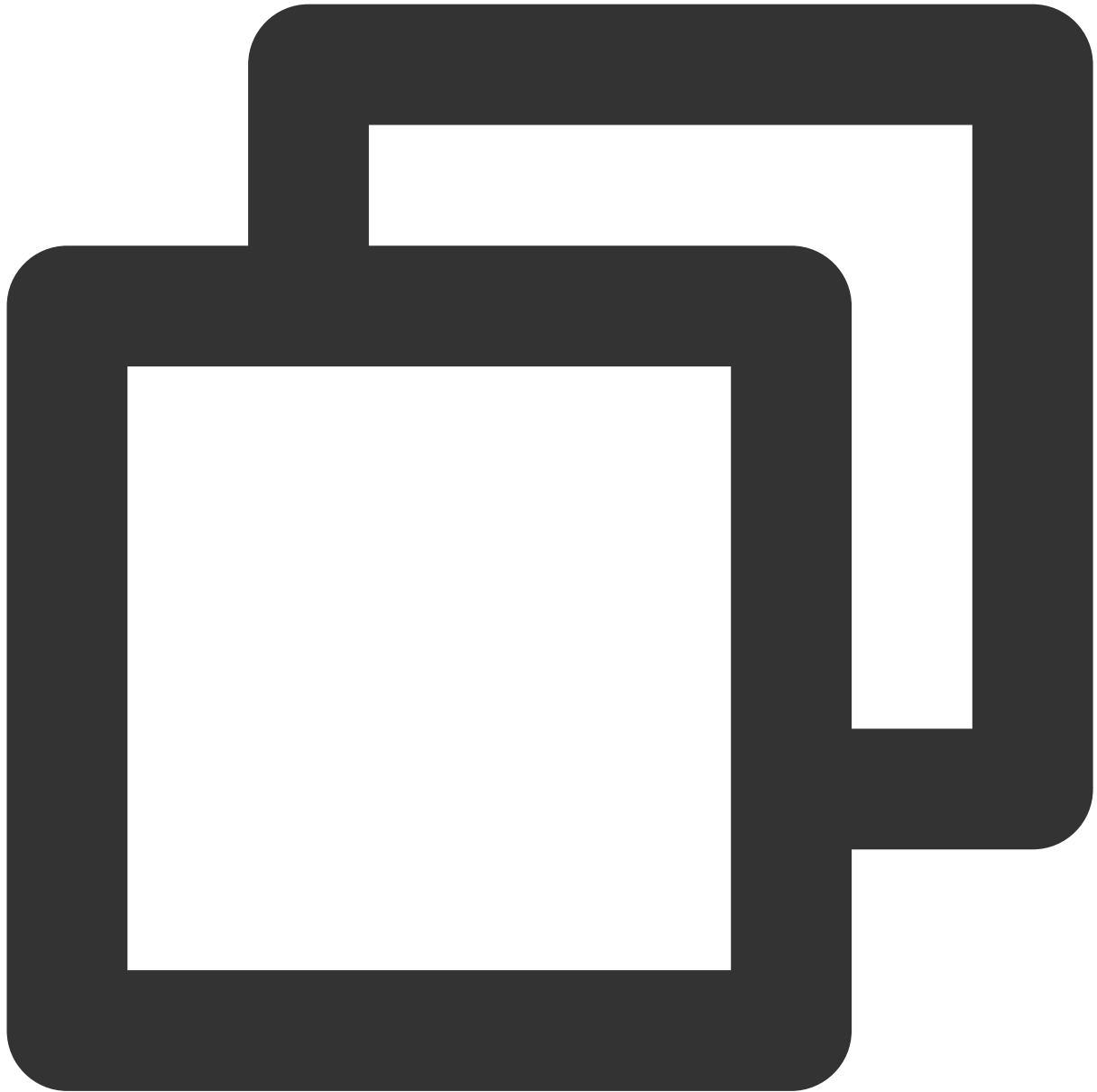
```
- (void)onYTDataEvent:(id _Nonnull)event;
```

After the callbacks are configured successfully **on Version 3.0.0**, the SDK will send a callback of facial data for each video frame.



```
- (void)onAIEvent:(id _Nonnull)event;
//在onAIEvent方法中可通过下边方法可以获取到数据
NSDictionary *eventDict = (NSDictionary *)event;
if (eventDict[@"ai_info"] != nil) {
    NSLog(@"ai_info %@",eventDict[@"ai_info"]);
}
```

The data returned is in JSON format and includes the following fields (for details about the 256 facial keypoints, see the illustration above):



```
/// @note The list of field descriptions
/**
| Field | Type | Value Range | Description |
| :---- | :---- | :---- | :---- |
| trace_id | int | [1,INF) | The face ID
| face_256_point | float | [0,screenWidth] or [0,screenHeight] | 512
| face_256_visible | float | [0,1] | Visibility
| out_of_screen | bool | true/false | Whether the
```

```
| left_eye_high_vis_ratio      | float | [0,1] | The  
| right_eye_high_vis_ratio     | float | [0,1] | The  
| left_eyebrow_high_vis_ratio  | float | [0,1] | The percent  
| right_eyebrow_high_vis_ratio | float | [0,1] | The percent  
| mouth_high_vis_ratio         | float | [0,1] | The  
**/  
- (void)onYTDataEvent:(id _Nonnull)event;
```

Android API Description

Integration guide for Android

For directions on how to integrate the Tencent Effect SDK for Android, see the integration guide for [Android](#).

Registering an Xmagic listener

This API is used to configure the callback of facial keypoints and other data.

Version 2.6.0 and earlier versions use the following method.



```
void setYTDataListener(XmagicApi.XmagicYTDataListener ytDataListener)
```

```
public interface XmagicYTDataListener {  
    void onYTDataUpdate(String data)  
}
```

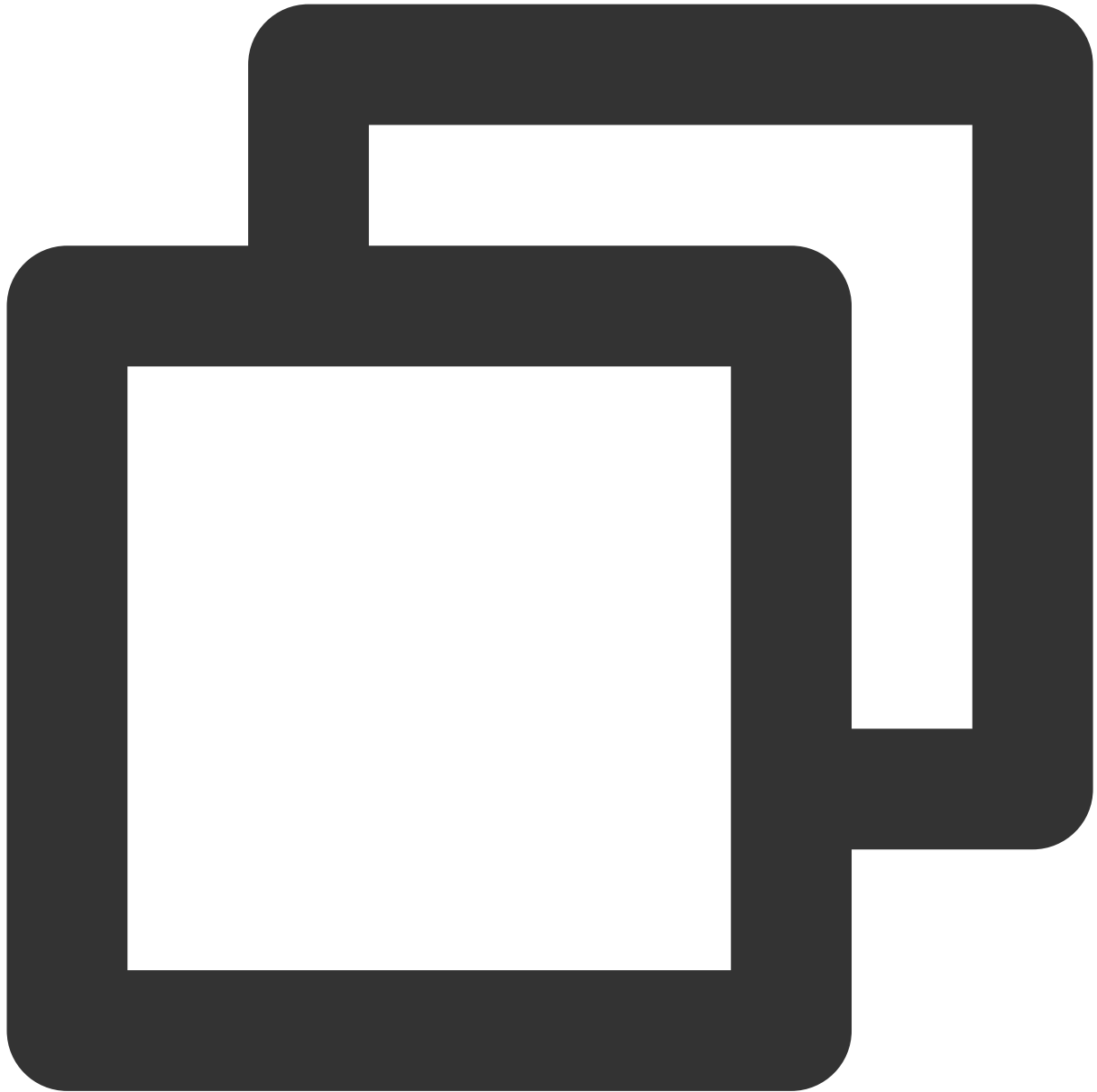
Version 3.0.0 uses the following method.



```
void setAIDataListener(XmagicApi.OnAIDataListener aiDataListener)

public interface OnAIDataListener {
    void onFaceDataUpdated(List<TEFaceData> faceDataList);
    void onHandDataUpdated(List<TEHandData> handDataList);
    void onBodyDataUpdated(List<TEBodyData> bodyDataList);
    void onAIDataUpdated(String data); //This method is a new method added in versi
}
```


`onYTDataUpdate` and `onAIDataUpdated` returns a JSON string structure that contains the information of up to 5 faces:



```
{
  "face_info": [{
    "trace_id": 5,
    "face_256_point": [
      180.0,
      112.2,
      ...
    ],
  },
]
```

```

    "face_256_visible": [
        0.85,
        ...
    ],
    "out_of_screen": true,
    "left_eye_high_vis_ratio": 1.0,
    "right_eye_high_vis_ratio": 1.0,
    "left_eyebrow_high_vis_ratio": 1.0,
    "right_eyebrow_high_vis_ratio": 1.0,
    "mouth_high_vis_ratio": 1.0
},
...
]
}

```

Fields

Field	Type	Value Range	Description
trace_id	int	[1,INF)	The face ID. If the faces obtained continuously from a video stream have the same face ID, they belong to the same person.
face_256_point	float	[0,screenWidth] or [0,screenHeight]	512 values in total for 256 facial keypoints. (0,0) is the top-left corner of the screen.
face_256_visible	float	[0,1]	Visibility of the 256 facial keypoints.
out_of_screen	bool	true/false	Whether the face is outside of the screen view.
left_eye_high_vis_ratio	float	[0,1]	The percentage of keypoints with high visibility for the left eye.
right_eye_high_vis_ratio	float	[0,1]	The percentage of keypoints with high visibility for the right eye.
left_eyebrow_high_vis_ratio	float	[0,1]	The percentage of keypoints with high visibility for the left eyebrow.
right_eyebrow_high_vis_ratio	float	[0,1]	The percentage of keypoints with high visibility for the right eyebrow.
mouth_high_vis_ratio	float	[0,1]	The percentage of keypoints with high visibility for the mouth.

Parameters

Parameters	Description
XmagicApi.XmagicYTDataListener ytDataListener	Callback function implementation class.

Virtual Background

Last updated : 2023-06-01 09:59:48

Accurately removes the background in real time and applies a virtual background (customizable):

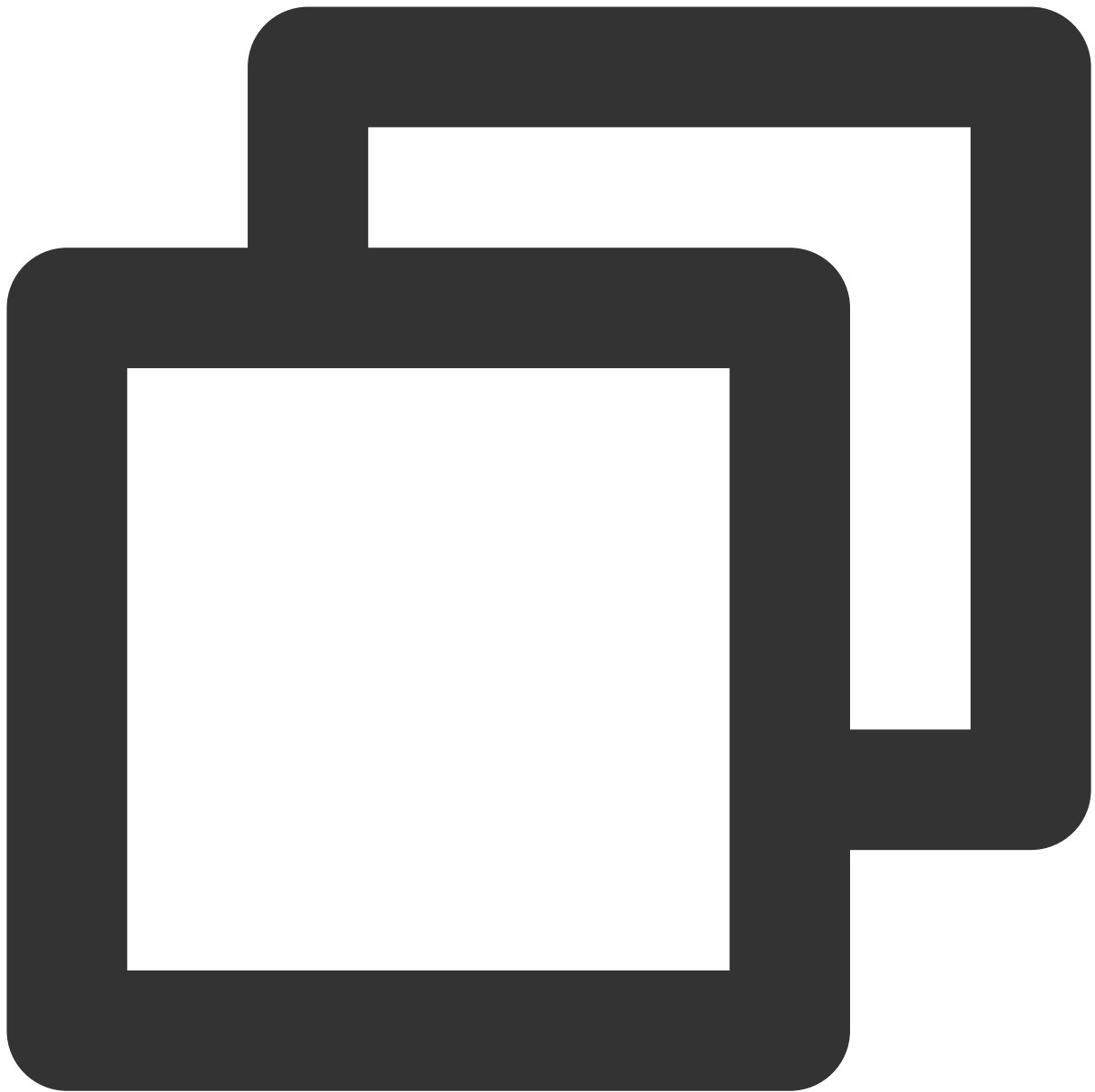


iOS API Use

Integration guide for iOS

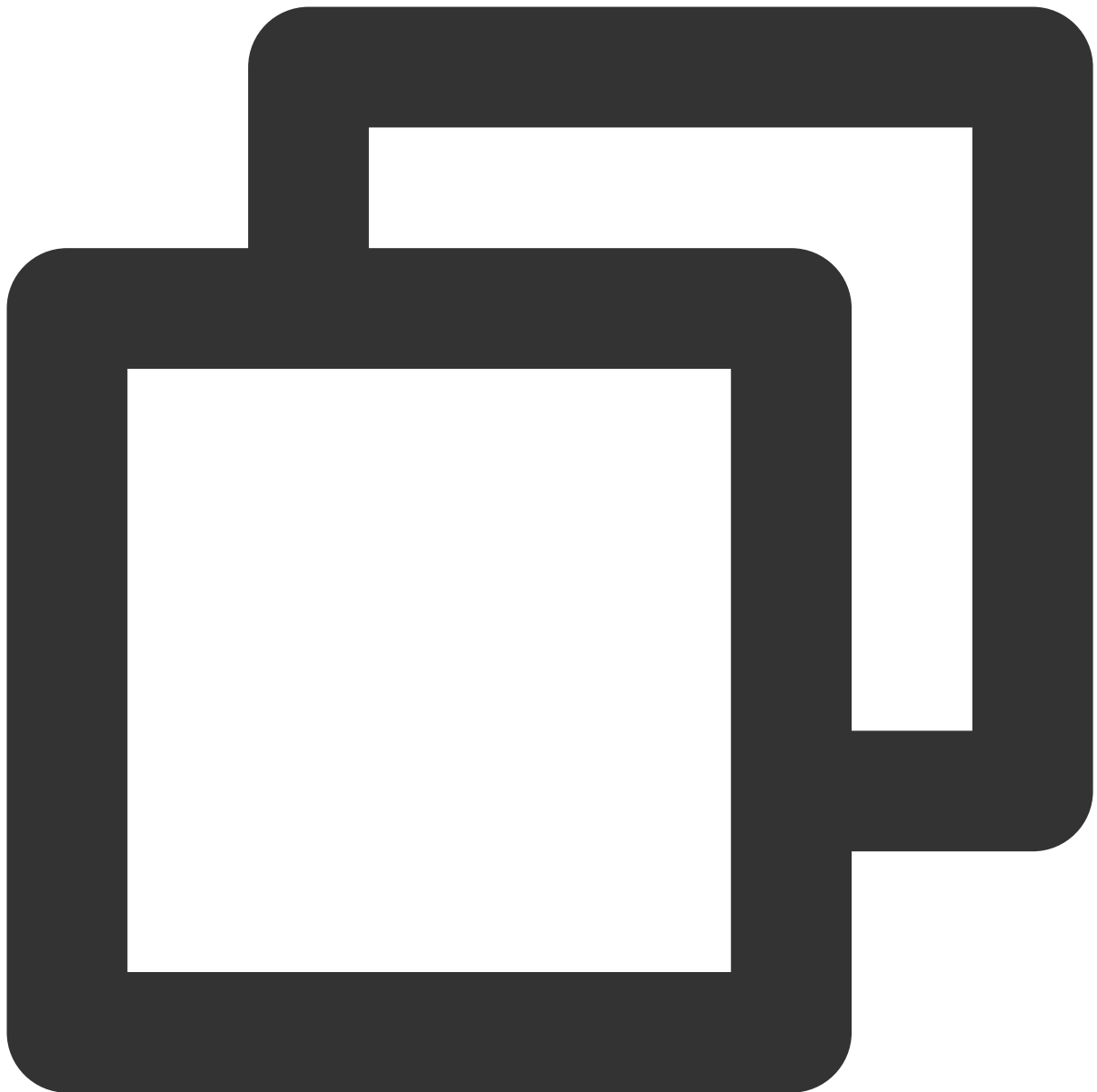
For directions on how to integrate the Tencent Effect SDK for iOS, see the integration guide for [iOS](#).

Setting a virtual background



```
NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotion" ofType:@"png"];
NSString *propertyType = @"motion"; //Set the effect type
NSString *propertyName = @"video_segmentation_blur_75"; //Specify the effect name
NSString *propertyValue = motionSegResPath; //Set the path of the animated effect
NSDictionary *dic = @{@"bgName":@"BgSegmentation.bg.png", @"bgType":@0, @"timeOffse":@0.5};
[self.beautyKit configPropertyWithType:propertyType withName:propertyName withData:dic];
```

Setting a custom background



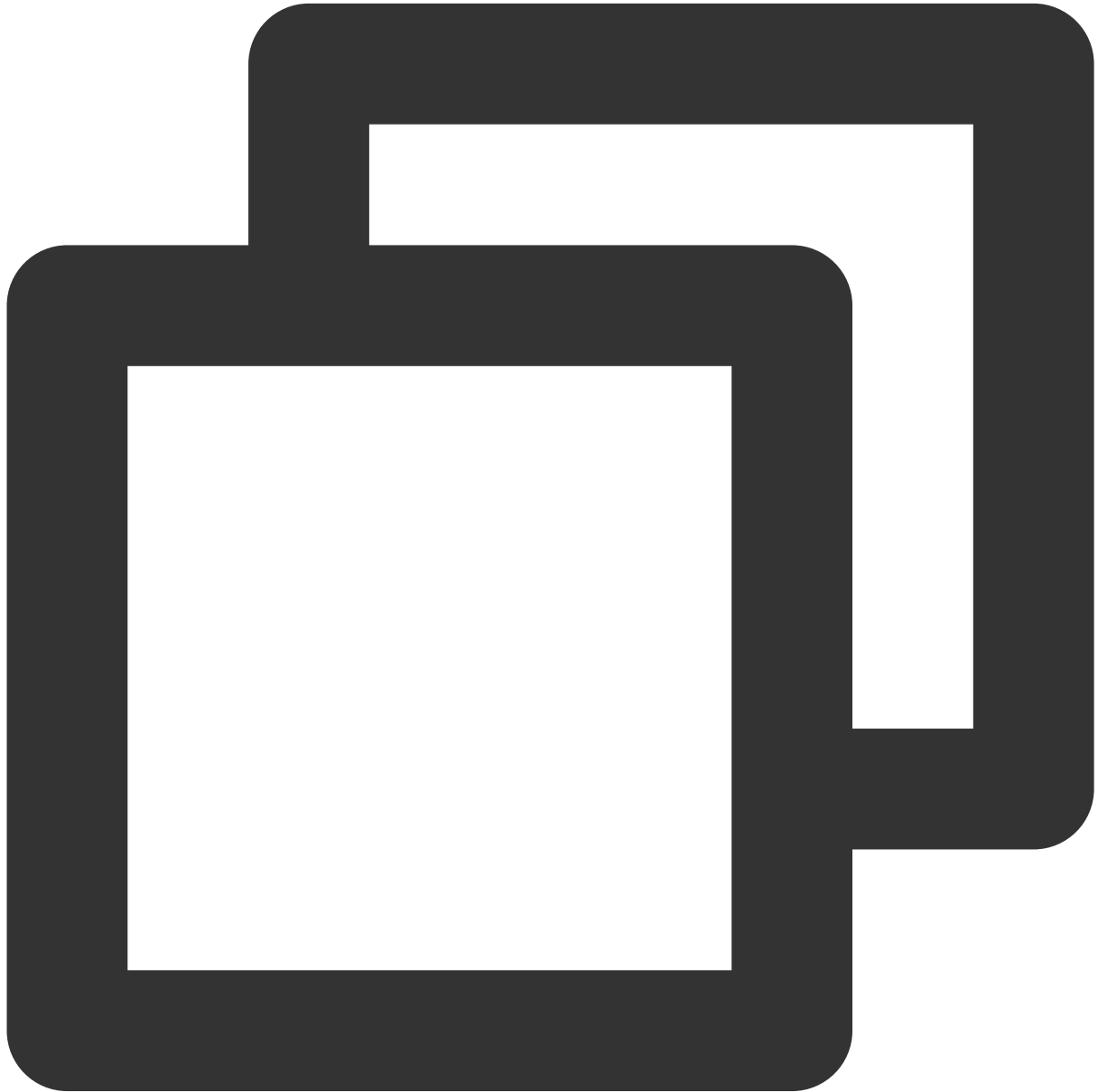
```
NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotion" ofType:@"png"];
NSString *propertyType = @"motion"; //Set the effect type
NSString *propertyName = @"video_empty_segmentation"; //Specify the effect name
NSString *propertyValue = motionSegResPath; //Set the path of the animated effect
NSString *imagePath = @"/var/mobile/Containers/Data/Application/06B00BBC-9060-450F-8C00-000000000000/Images/segmentMotion.png";
int bgType = 0; //The background type. 0: image; 1: video
int timeOffset = 0; //The duration. If an image is used as the background, its value should be 0.
NSDictionary *dic = @{@"bgName": imagePath, @"bgType": @(bgType), @"timeOffset": @(timeOffset)};
[self.beautyKit configPropertyWithType:propertyType withName:propertyName withData:dic];
```

Android API Use

Integration guide for Android

For directions on how to integrate the Tencent Effect SDK for Android, see the integration guide for [Android](#).

Attribute setting API

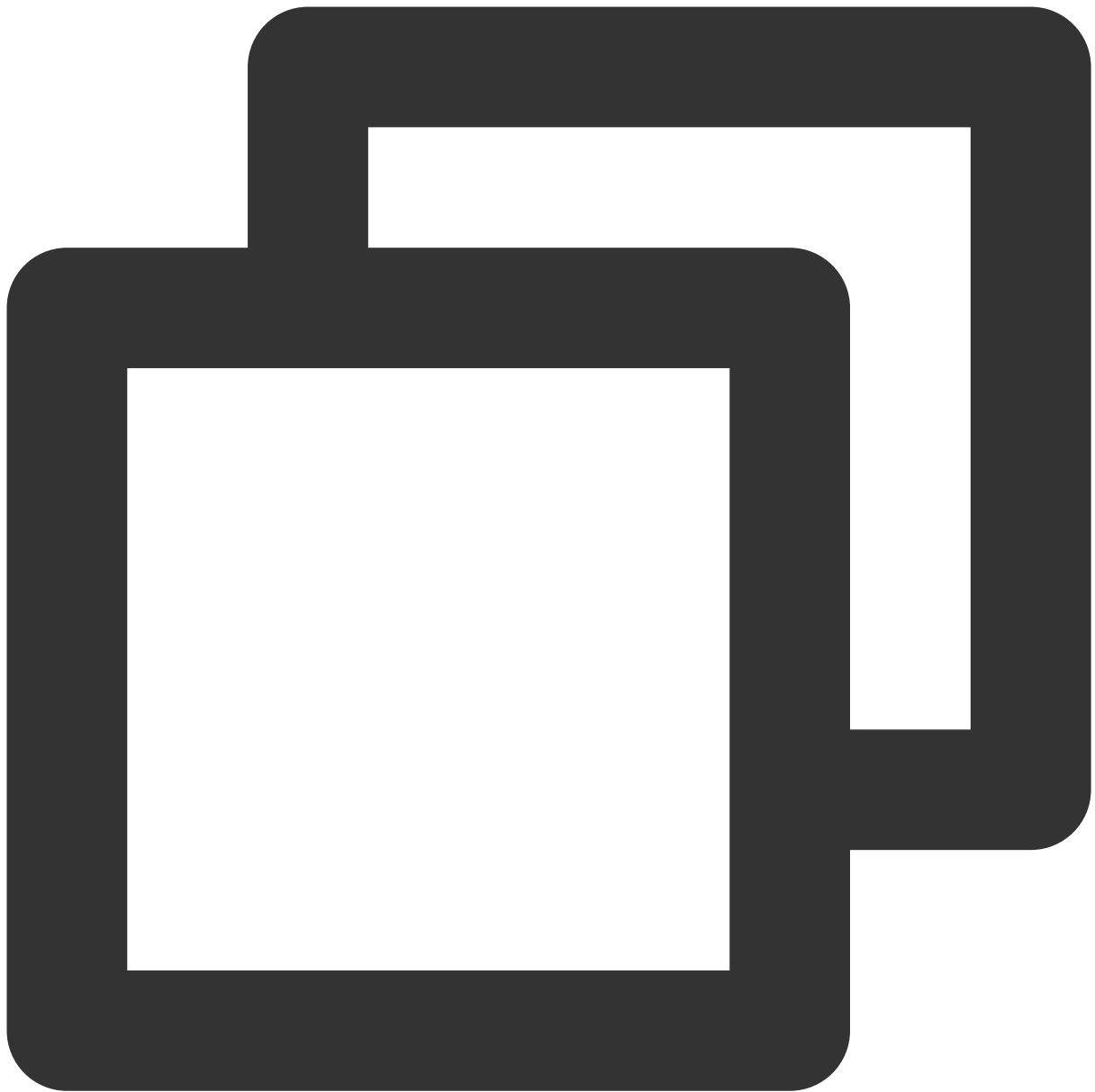


```
void updateProperty(XmagicProperty<?> p)
```

Keying parameters:

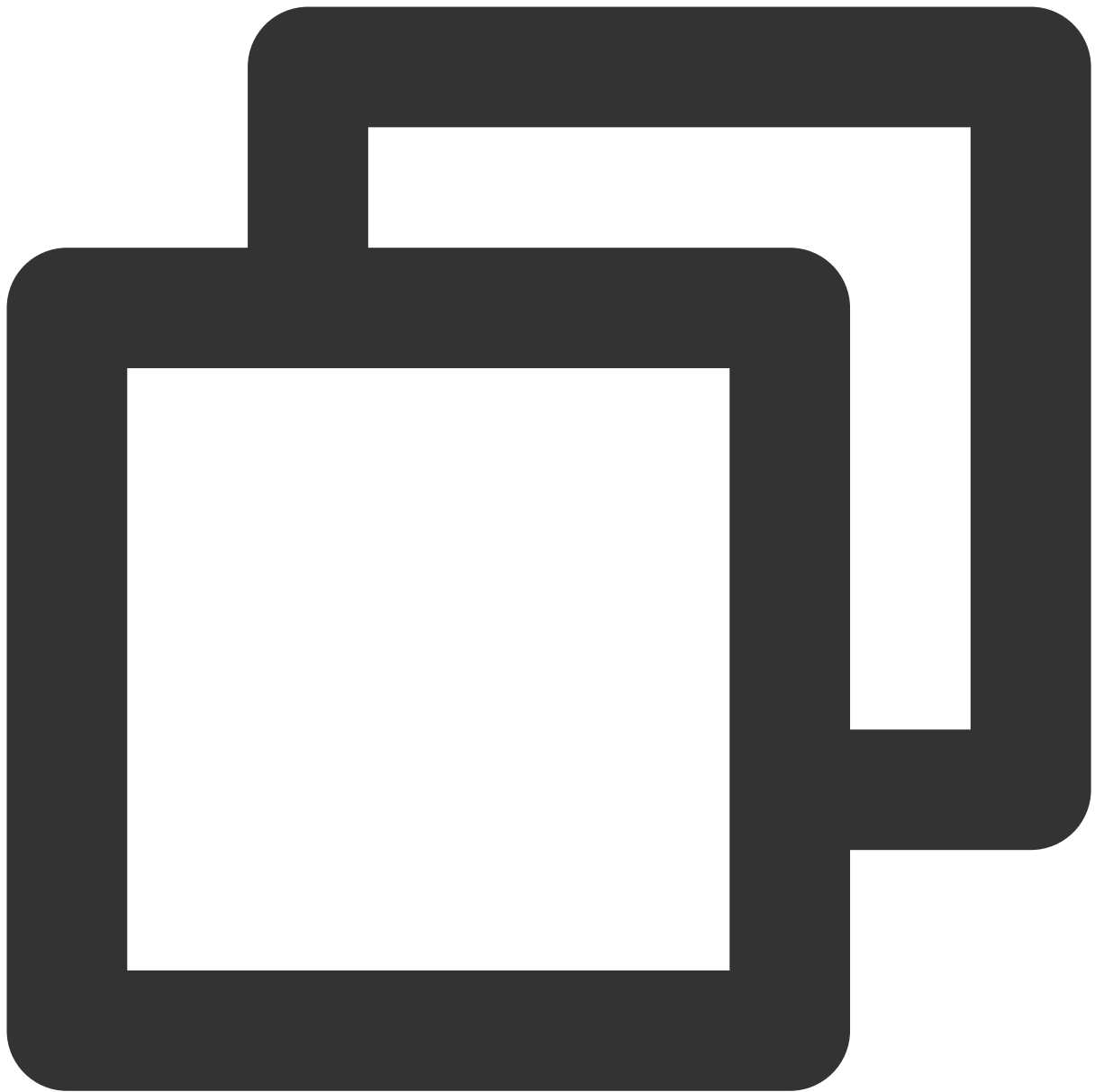
Attribute Field	Description
category	Category.SEGMENTATION
ID	The resource folder name , which is required, such as <code>video_segmentation_blur_45</code> . <code>XmagicProperty.ID_NONE</code> indicates there is no ID. A custom keying ID must be <code>XmagicConstant.SegmentationId.CUSTOM_SEG_ID</code> .
resPath	Required. For more information, see the demo.
effkey	null (except for custom backgrounds). The value of a custom background is the selected resource path.
effValue	null

Setting a virtual background



```
// Initialize `XmagicProperty`  
XmagicProperty xmagicProperty = new XmagicProperty(Category.SEGMENTATION, "video_seg  
// Set the attributes  
xmagicApi.updateProperty(xmagicProperty)
```

Setting a custom background



```
XmagicProperty xmagicProperty = new XmagicProperty(Category.SEGMENTATION,XmagicCons  
xmagicApi.updateProperty(xmagicProperty)
```

Facial Features

Android

Last updated : 2023-06-01 09:53:18

Function Description

Input a photo containing a face and output facial feature information of the person, including eyes, eyebrows, hairstyle, skin color, gender, age, etc. This interface requires internet connection, and the SDK will upload the photo to the server for parsing.

Integration Guide

First, you need to integrate Tencent Effect SDK, please refer to Independent Integration of [Tencent Effect for details](#).

Interface Description

XMagicApi.java



```
public void getFaceFeatureFromPhoto(Bitmap bitmap, FaceFeatureListener listener);
```

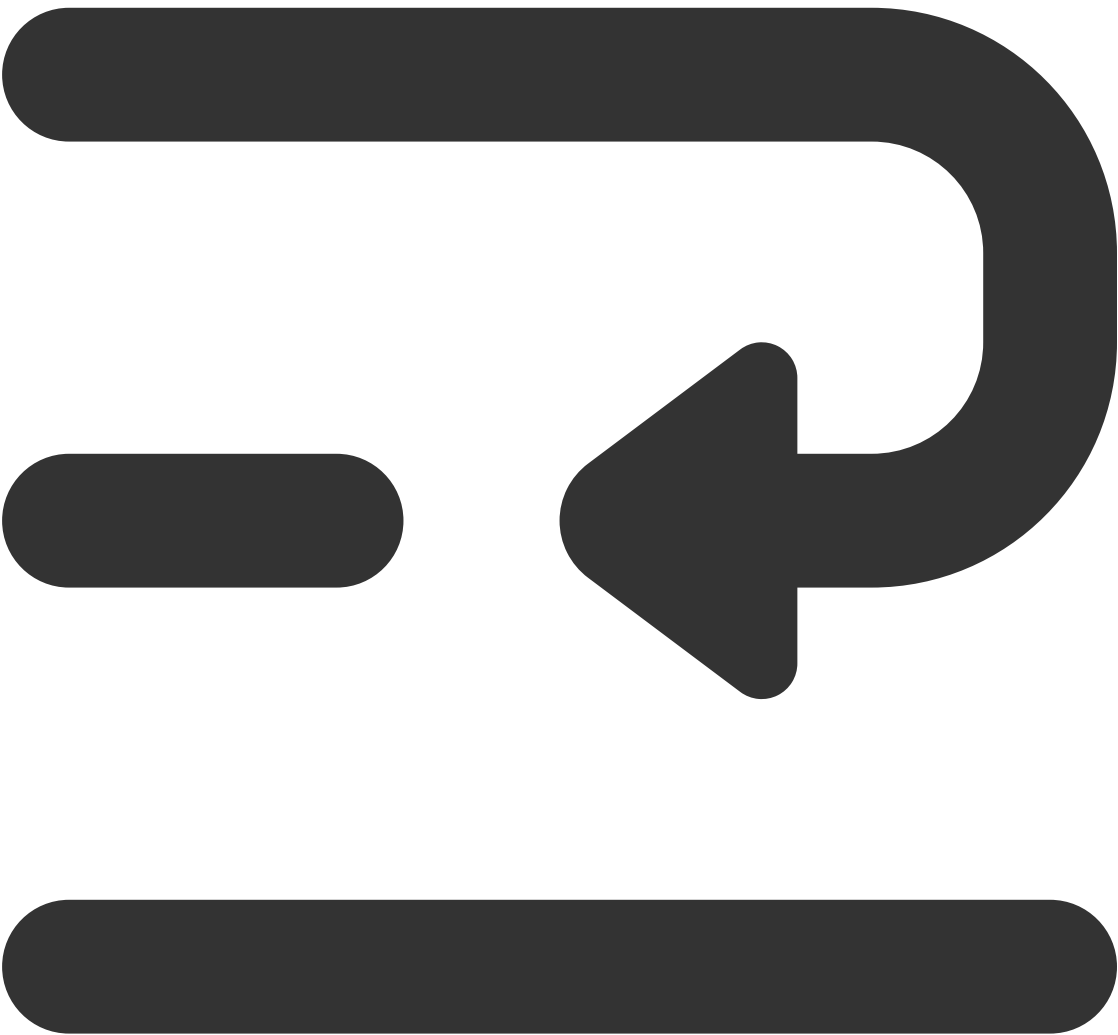
Parameter bitmap: Please try to make the face in the center of the picture. It is recommended that the picture only contains one face. If there are multiple faces, the SDK will randomly select one. It is recommended that the short side of the photo be greater than or equal to 500px. A small size will affect the recognition effect.

Parameter FaceFeatureListener, returns the recognition result.



```
public interface FaceFeatureListener {  
    void onError(int errCode, String msg);  
    void onFinish(FaceDetailAttributesInfo faceInfo);  
}
```

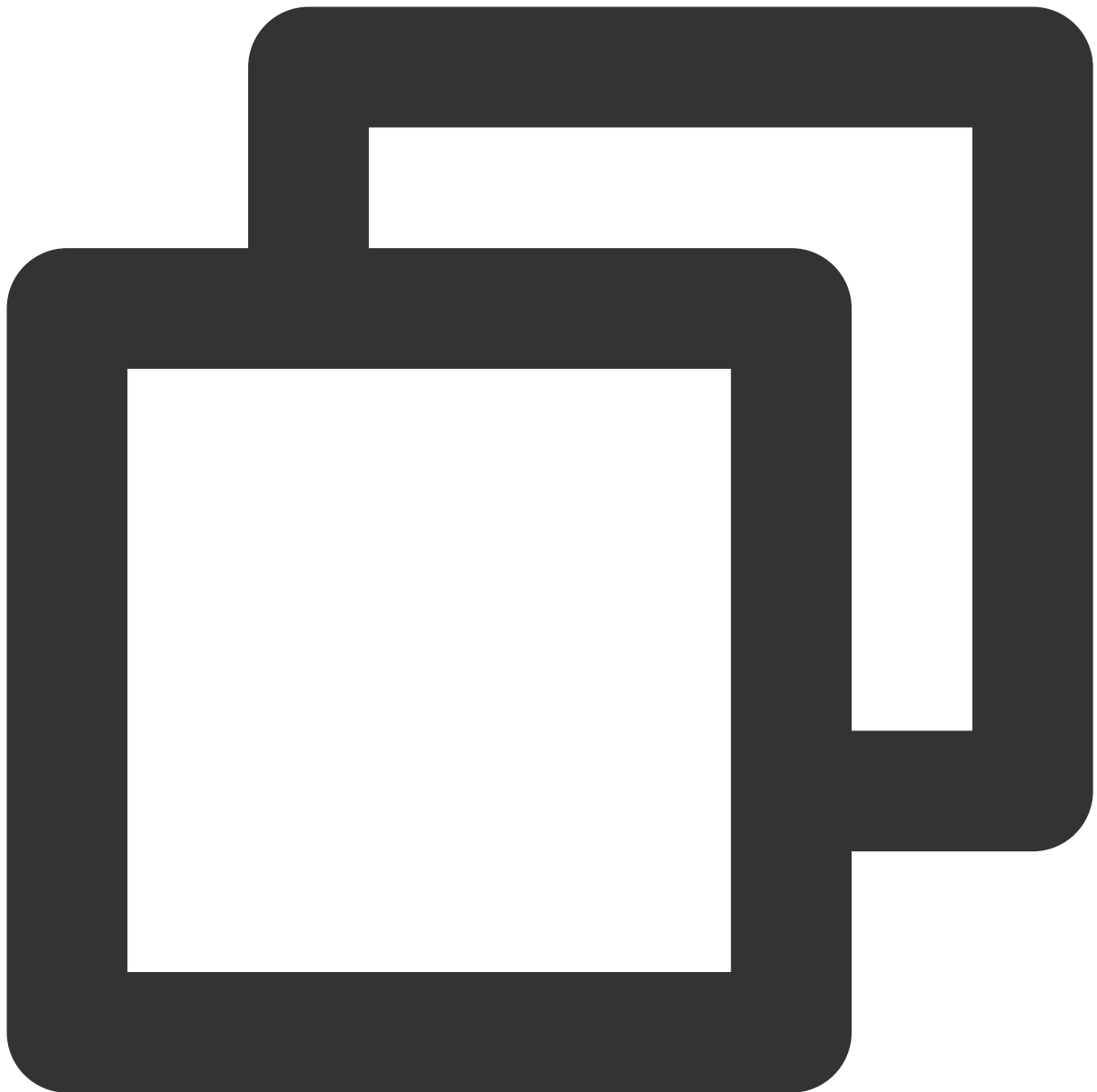
onError callback: This interface will be called back when the parsing fails, and the error codes are as follows.





```
public static final int ERROR_NO_AUTH = 1; // No permission
public static final int ERROR_RES_INVALID = 5; // Invalid Avatar material path
public static final int ERROR_PHOTO_INVALID = 10; // Failed to read the photo
public static final int ERROR_NETWORK_REQUEST_FAILED = 20; // Network request failed
public static final int ERROR_DATA_PARSE_FAILED = 30; // Failed to parse network request
public static final int ERROR_ANALYZE_FAILED = 40; // Face analysis failed
public static final int ERROR_AVATAR_SOURCE_DATA_EMPTY = 50; // Failed to load Avatar
```

onFinish callback: This interface will be called back when the parsing is successful, and FaceDetailAttributesInfo is explained as follows.



```
public static class FaceDetailAttributesInfo {
    public int age;    //[0,100]
    public int emotion;    //0: natural, 1: happy, 2: surprised, 3:
    public Eye eye;    // Eye information
    public Eyebrow eyebrow;    // Eyebrow information
    public int gender;    // Gender information. -1: not recognized, 0
    public Hair hair;    // Hairstyle information
    public Hat hat;    // Hat information
    public int mask;    // Whether there is a mask. -1: not recogniz
    public int moustache;    // Moustache information. -1: not recogn
```



```
        public int nose;        // Nose information. -1: not recognized, 0:  
        public int shape;       // Face shape information. -1: not recognize  
        public int skin;        // Skin color information. -1: not recognize  
        public int smile;       // Smile degree, [0,100].  
    }  
  
    public static class Eye {  
        public int eyelidType;   // Whether double eyelids are recognized  
  
        public int eyeSize;      // Eye size. -1: not recognized, 0: smal  
  
        public int glass;        // Whether wearing glasses is recognized  
  
        public int eyeOpen;      // Recognize the opening and closing sta  
    }  
  
    public static class Eyebrow {  
        public int eyebrowLength; // Eyebrow length. 0: short eyebrows  
  
        public int eyebrowDensity; // Eyebrow density. 0: light eyebrow  
  
        public int eyebrowCurve;   // Eyebrow curvature. 0: not curved,  
    }  
  
    public static class Hair {  
        public int length; // Hair length information. 0: bald, 1: short  
  
        public int bang;     // Bangs information. 0: no bangs, 1: with b  
  
        public int color;    // Hair color information. 0: black, 1: gol  
    }  
  
    public static class Hat {  
        public int style;     // Hat wearing status information. 0: no hat  
  
        public int color;     // Hat color. 0: no hat, 1: red series, 2: y  
    }
```

Facial Expressions

iOS

Last updated : 2023-05-18 16:31:37

Overview

This capability processes the OpenGL textures of the camera and outputs blendshape data that meets the standards of Apple ARKit. For details, see [ARFaceAnchor](#). You can pass the data to Unity to drive your model or use the data to implement other features.

Integration (iOS)

Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the Tencent Effect SDK.

API Calls

1. Enable the feature.



```
[self.beautyKit setFeatureEnableDisable:ANIMOJI_52_EXPRESSION enable:YES];
```

2. Configure the callback of facial keypoints.

Version 2.6.0 and earlier versions use the following method.



```
//XMagic.h
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;

@implementation listener
- (void)onYTDataEvent:(id)event
{
    NSLog(@"YTData %@", event);
}
```

Version 3.0.0 uses the following method.

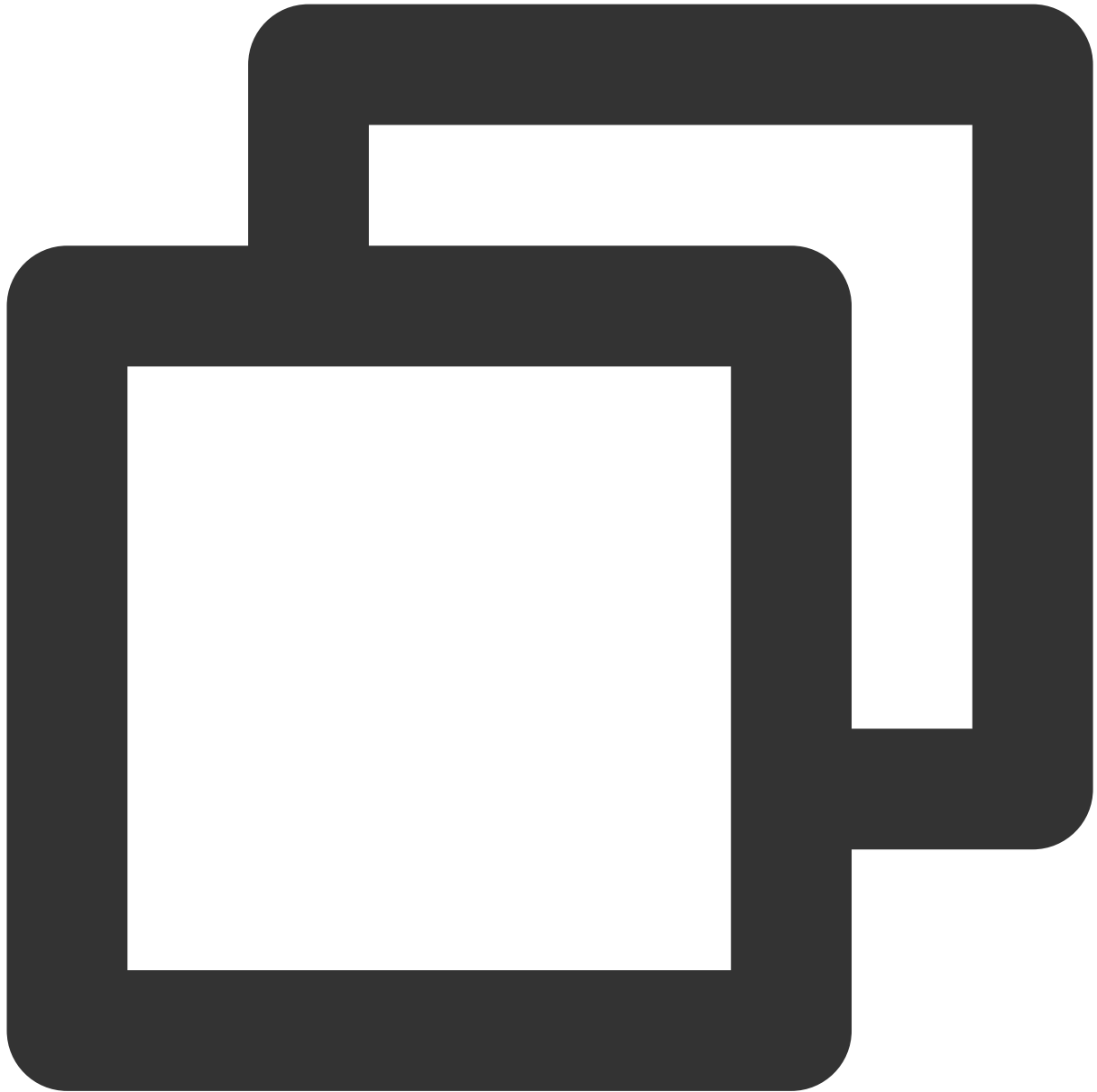


```
//XMagic.h
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;

- (void)onAIEvent:(id)event
{
    NSDictionary *eventDict = (NSDictionary *)event;
    if (eventDict[@"ai_info"] != nil) {
        NSLog(@"ai_info %@", eventDict[@"ai_info"]);
    }
}
```

`onYTDataUpdate` returns a JSON string structure that contains the information of up to 5 faces:

`onAIEvent` returns a JSON string structure that contains the information of up to 5 faces:



```
{
  "face_info": [{
    "trace_id": 5,
    "face_256_point": [
      180.0,
      112.2,
      ...
    ],
  },
```

```

    "face_256_visible": [
      0.85,
      ...
    ],
    "out_of_screen": true,
    "left_eye_high_vis_ratio": 1.0,
    "right_eye_high_vis_ratio": 1.0,
    "left_eyebrow_high_vis_ratio": 1.0,
    "right_eyebrow_high_vis_ratio": 1.0,
    "mouth_high_vis_ratio": 1.0,
    "expression_weights": [
      0.12,
      -0.32
      ...
    ]
  },
  ...
]
}

```

Field descriptions

trace_id: The face ID. If the faces obtained from a continuous video stream have the same face ID, they belong to the same person.

expression_weights: The real-time blendshape data. The array contains 52 elements. The value range of each element is 0-1.0.

```

{ "eyeBlinkLeft", "eyeLookDownLeft", "eyeLookInLeft", "eyeLookOutLeft", "eyeLookUpLeft", "eyeSquintLeft", "eyeWideLeft", "eyeBlinkRight", "eyeLookDownRight", "eyeLookInRight", "eyeLookOutRight", "eyeLookUpRight", "eyeSquintRight", "eyeWideRight", "jawForward", "jawLeft", "jawRight", "jawOpen", "mouthClose", "mouthFunnel", "mouthPucker", "mouthRight", "mouthLeft", "mouthSmileLeft", "mouthSmileRight", "mouthFrownRight", "mouthFrownLeft", "mouthDimpleLeft", "mouthDimpleRight", "mouthStretchLeft", "mouthStretchRight", "mouthRollLower", "mouthRollUpper", "mouthShrugLower", "mouthShrugUpper", "mouthPressLeft", "mouthPressRight", "mouthLowerDownLeft", "mouthLowerDownRight", "mouthUpperUpLeft", "mouthUpperUpRight", "browDownLeft", "browDownRight", "browInnerUp", "browOuterUpLeft", "browOuterUpRight", "cheekPuff", "cheekSquintLeft"

```

```
";"cheekSquintRight";"noseSneerLeft";"noseSneerRight";"tongueOut"}
```

The other fields are [facial keypoint information](#). Whether they are returned depends on the type of license you use. If you only need facial expression data, you can ignore those fields.

Android

Last updated : 2023-05-18 16:32:30

Overview

This capability processes the OpenGL textures of the camera and outputs blendshape data that meets the standards of Apple ARKit. For details, see [ARFaceAnchor](#). You can pass the data to Unity to drive your model or use the data to implement other features.

Integration (Android)

Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the Tencent Effect SDK.

API calls

1. Enable the feature.



```
//XmagicApi.java
//featureName = XmagicConstant.FeatureName.ANIMOJI_52_EXPRESSION
public void setFeatureEnableDisable(String featureName, boolean enable);
```

2. Configure the callback of facial keypoints.

Version 2.6.0 and earlier versions use the following method.



```
void setYTDataListener(XmagicApi.XmagicYTDataListener ytDataListener)
```

```
public interface XmagicYTDataListener {  
    void onYTDataUpdate(String data)  
}
```

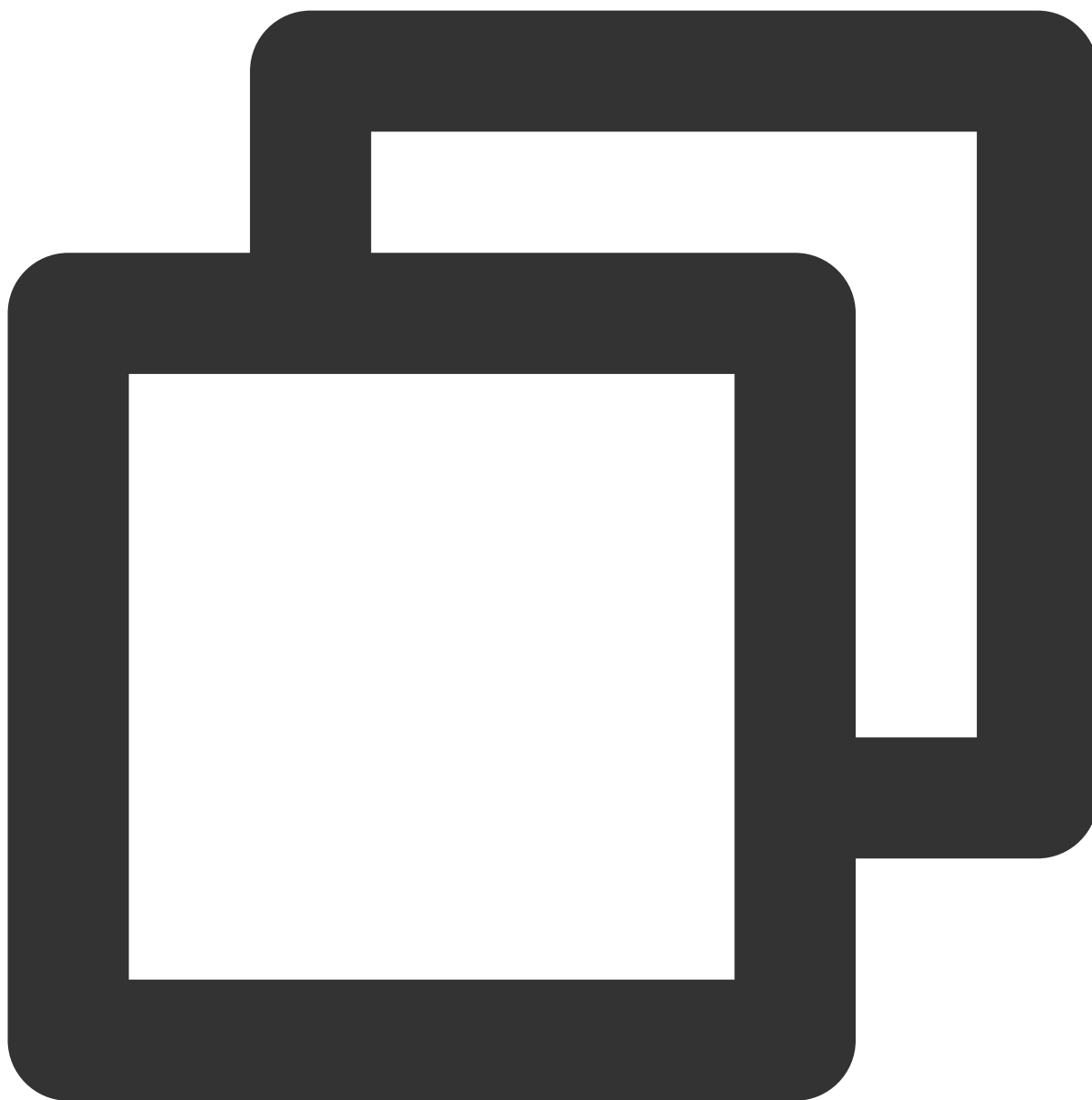
Version 3.0.0 uses the following method.



```
void setAIDataListener(XmagicApi.OnAIDataListener aiDataListener)

public interface OnAIDataListener {
    void onFaceDataUpdated(List<TEFaceData> faceDataList);
    void onHandDataUpdated(List<TEHandData> handDataList);
    void onBodyDataUpdated(List<TEBodyData> bodyDataList);
    void onAIDataUpdated(String data); //This method is a new method added in versi
}
```

`onYTDataUpdate` and `onAIDataUpdated` returns a JSON string structure that contains the information of up to 5 faces:



```
{
  "face_info": [{
    "trace_id": 5,
    "face_256_point": [
      180.0,
      112.2,
      ...
    ],
  },
```

```

    "face_256_visible": [
      0.85,
      ...
    ],
    "out_of_screen": true,
    "left_eye_high_vis_ratio": 1.0,
    "right_eye_high_vis_ratio": 1.0,
    "left_eyebrow_high_vis_ratio": 1.0,
    "right_eyebrow_high_vis_ratio": 1.0,
    "mouth_high_vis_ratio": 1.0,
    "expression_weights": [
      0.12,
      -0.32
      ...
    ]
  },
  ...
]
}

```

Field descriptions

trace_id: The face ID. If the faces obtained from a continuous video stream have the same face ID, they belong to the same person.

expression_weights: The real-time blendshape data. The array contains 52 elements. The value range of each element is 0-1.0.

```

{ "eyeBlinkLeft", "eyeLookDownLeft", "eyeLookInLeft", "eyeLookOutLeft", "eyeLookUpLeft", "eyeSquintLeft", "eyeWideLeft", "eyeBlinkRight", "eyeLookDownRight", "eyeLookInRight", "eyeLookOutRight", "eyeLookUpRight", "eyeSquintRight", "eyeWideRight", "jawForward", "jawLeft", "jawRight", "jawOpen", "mouthClose", "mouthFunnel", "mouthPucker", "mouthRight", "mouthLeft", "mouthSmileLeft", "mouthSmileRight", "mouthFrownRight", "mouthFrownLeft", "mouthDimpleLeft", "mouthDimpleRight", "mouthStretchLeft", "mouthStretchRight", "mouthRollLower", "mouthRollUpper", "mouthShrugLower", "mouthShrugUpper", "mouthPressLeft", "mouthPressRight", "mouthLowerDownLeft", "mouthLowerDownRight", "mouthUpperUpLeft", "mouthUpperUpRight", "browDownLeft", "browDownRight", "browInnerUp", "browOuterUpLeft", "browOuterUpRight", "cheekPuff", "cheekSquintLeft", "cheekSquintRight", "noseSneerLeft", "noseSneerRight", "tongueOut" }

```

The other fields are [facial keypoint information](#). Whether they are returned depends on the type of license you use. If you only need facial expression data, you can ignore those fields.

Body Keypoints

iOS

Last updated : 2023-05-18 16:33:08

Overview

This capability processes the OpenGL textures of the camera and outputs 3D body data. You can pass the data to Unity to drive your model or use the data to implement other features.

Integration (iOS)

Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the Tencent Effect SDK.

API calls

1. Enable the feature (XMagic.h).



```
- (void)setFeatureEnableDisable:(NSString *_Nonnull)featureName enable:(BOOL)enable
```

Set `featureName` to `XmagicConstant.FeatureName.BODY_3D_POINT` .

2. Configure the data callback (XMagic.h).

Version 2.6.0 and earlier versions use the following method.



```
//XMagic.h
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;

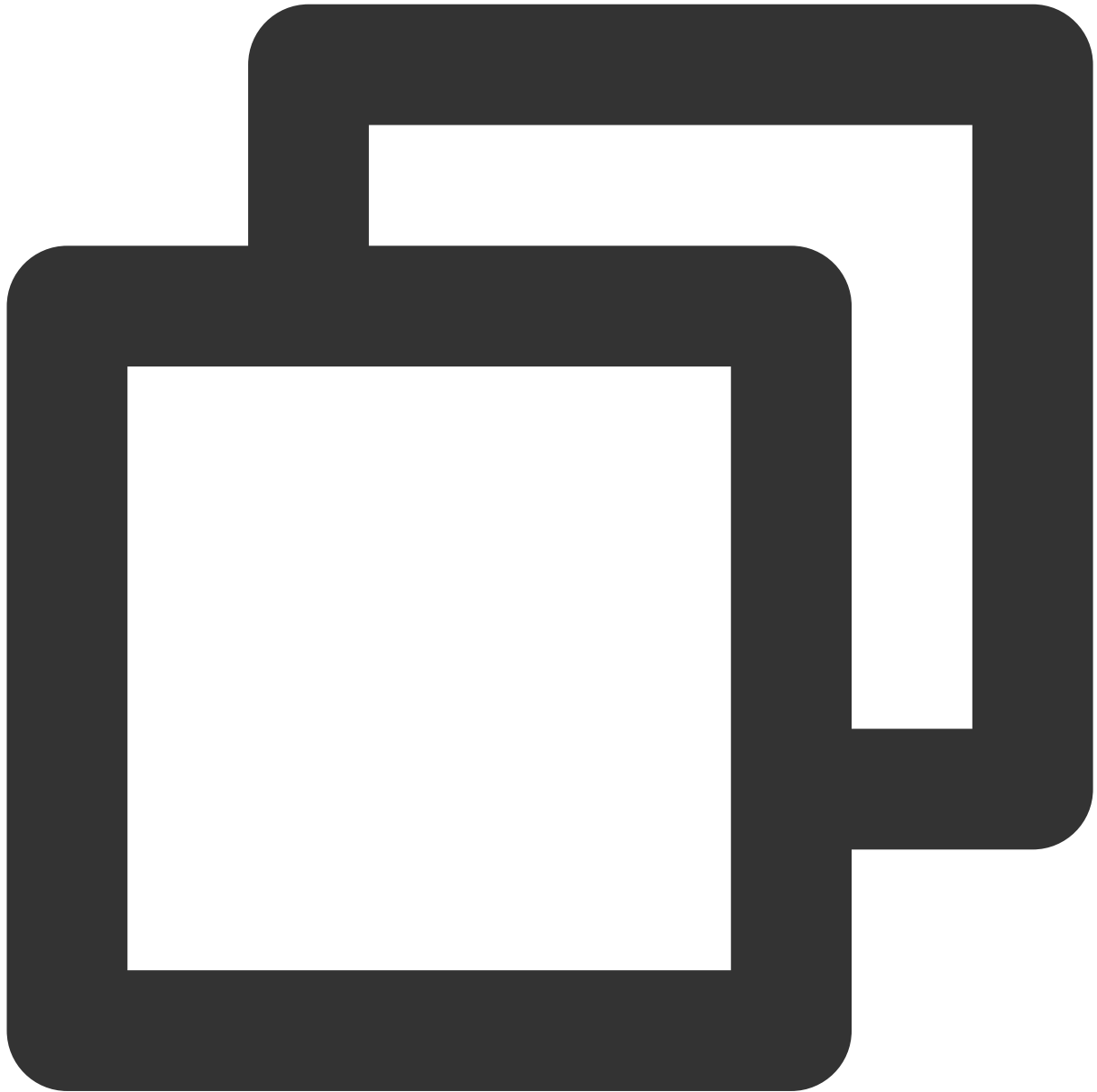
@implementation listener
- (void)onYTDataEvent:(id)event
{
    NSLog(@"YTData %@", event);
}
@end
```

`onYTDataEvent` returns a JSON string.

`face_info` is facial data, which is not used by this capability.

For the meanings of fields in `body_3d_info`, see below.

Version 3.0.0 uses the following method.



```
//XMagic.h
- (void)registerSDKEventListener:(id<YTSDKEventListener> _Nullable)listener;

- (void)onAIEvent:(id)event
{
```

```
NSMutableDictionary *eventDict = (NSMutableDictionary *)event;
if (eventDict[@"ai_info"] != nil) {
    NSLog(@"ai_info %@", eventDict[@"ai_info"]);
}
}
```

`eventDict[@"ai_info"]` returns a JSON string.

`face_info` is facial data, which is not used by this capability.

For the meanings of fields in `body_3d_info`, see below.

Body Keypoints and Descriptions

For details about body keypoints, see [Body Keypoints and Descriptions](#).

Android

Last updated : 2023-05-18 16:33:49

Overview

This capability processes the OpenGL textures of the camera and outputs 3D body data. You can pass the data to Unity to drive your model or use the data to implement other features.

Integration (Android)

Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the Tencent Effect SDK.

API calls

1. Enable the feature (XmagicApi.java).



```
public void setFeatureEnableDisable(String featureName, boolean enable);
```

Set `featureName` to `XmagicConstant.FeatureName.BODY_3D_POINT` .

2. Configure the data callback (XmagicApi.java).

Version 2.6.0 and earlier versions use the following method.



```
void setYTDataListener(XmagicApi.XmagicYTDataListener ytDataListener)
```

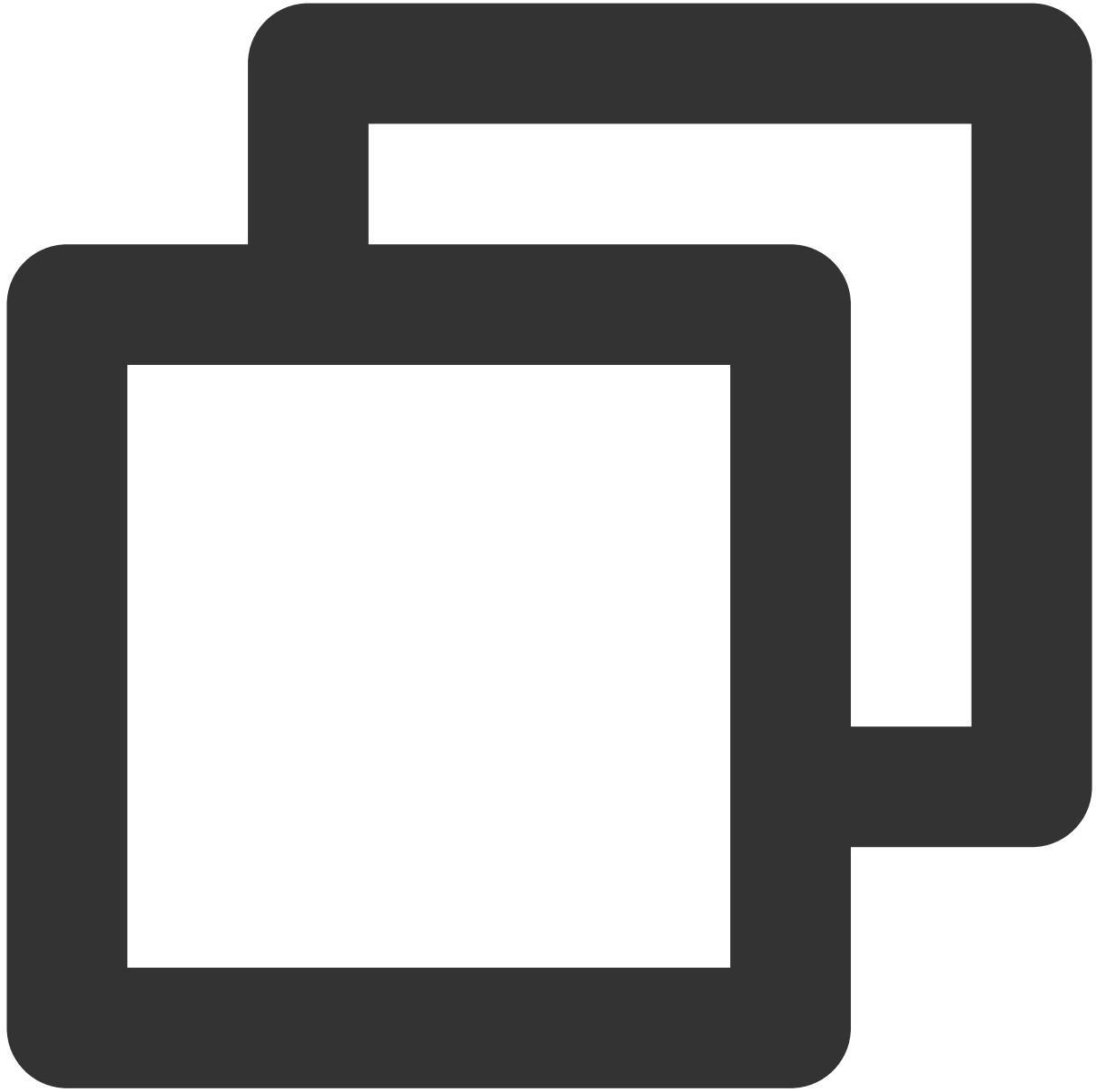
```
public interface XmagicYTDataListener {  
    void onYTDataUpdate(String data)  
}
```

`onYTDataUpdate` returns a JSON string. You can find an example below.

`face_info` is facial data, which is not used by this capability.

For the meanings of fields in `body_3d_info`, see below.

Version 3.0.0 uses the following method.



```
void setAIDataListener(XmagicApi.OnAIDataListener aiDataListener)

public interface OnAIDataListener {
    void onFaceDataUpdated(List<TEFaceData> faceDataList);
    void onHandDataUpdated(List<TEHandData> handDataList);
    void onBodyDataUpdated(List<TEBodyData> bodyDataList);
    void onAIDataUpdated(String data); //This method is a new method added in versi
}
```

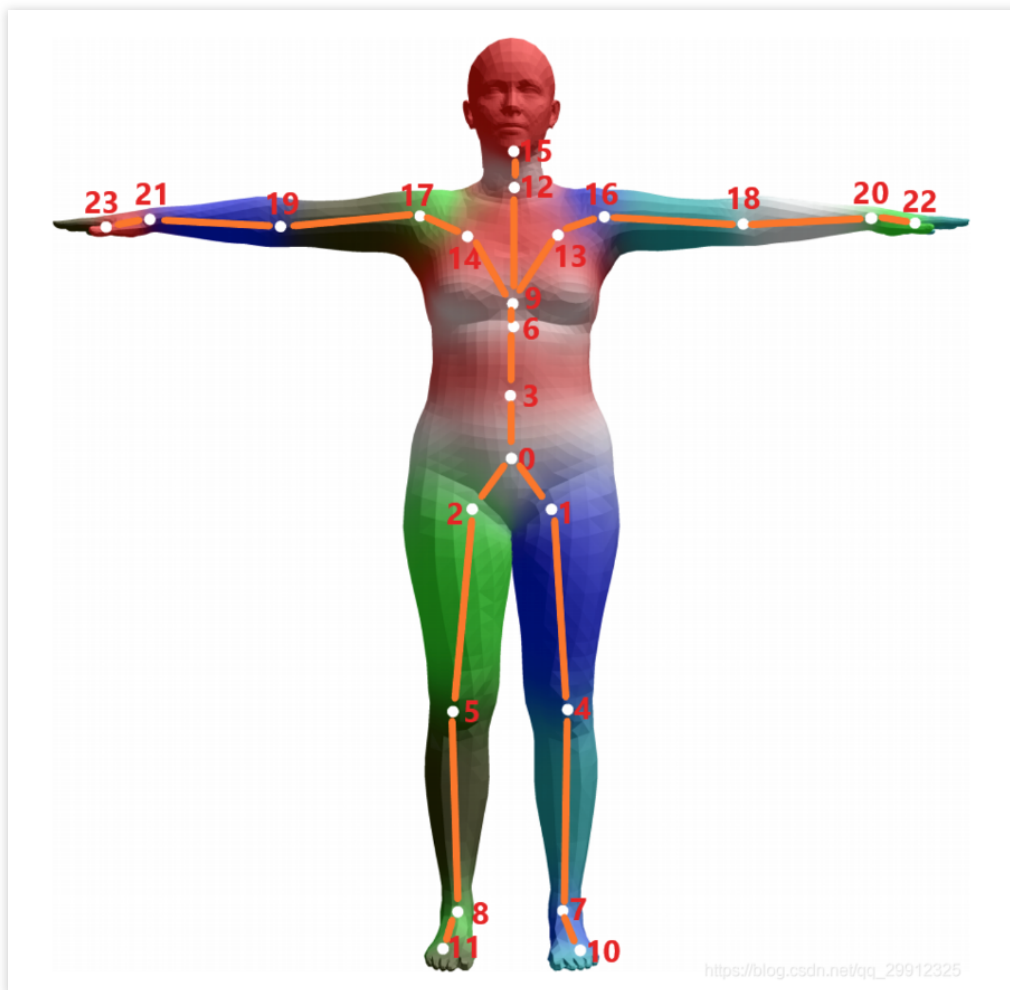
`onAIDataUpdated` returns a JSON string. You can find an example below.

`face_info` is facial data, which is not used by this capability.

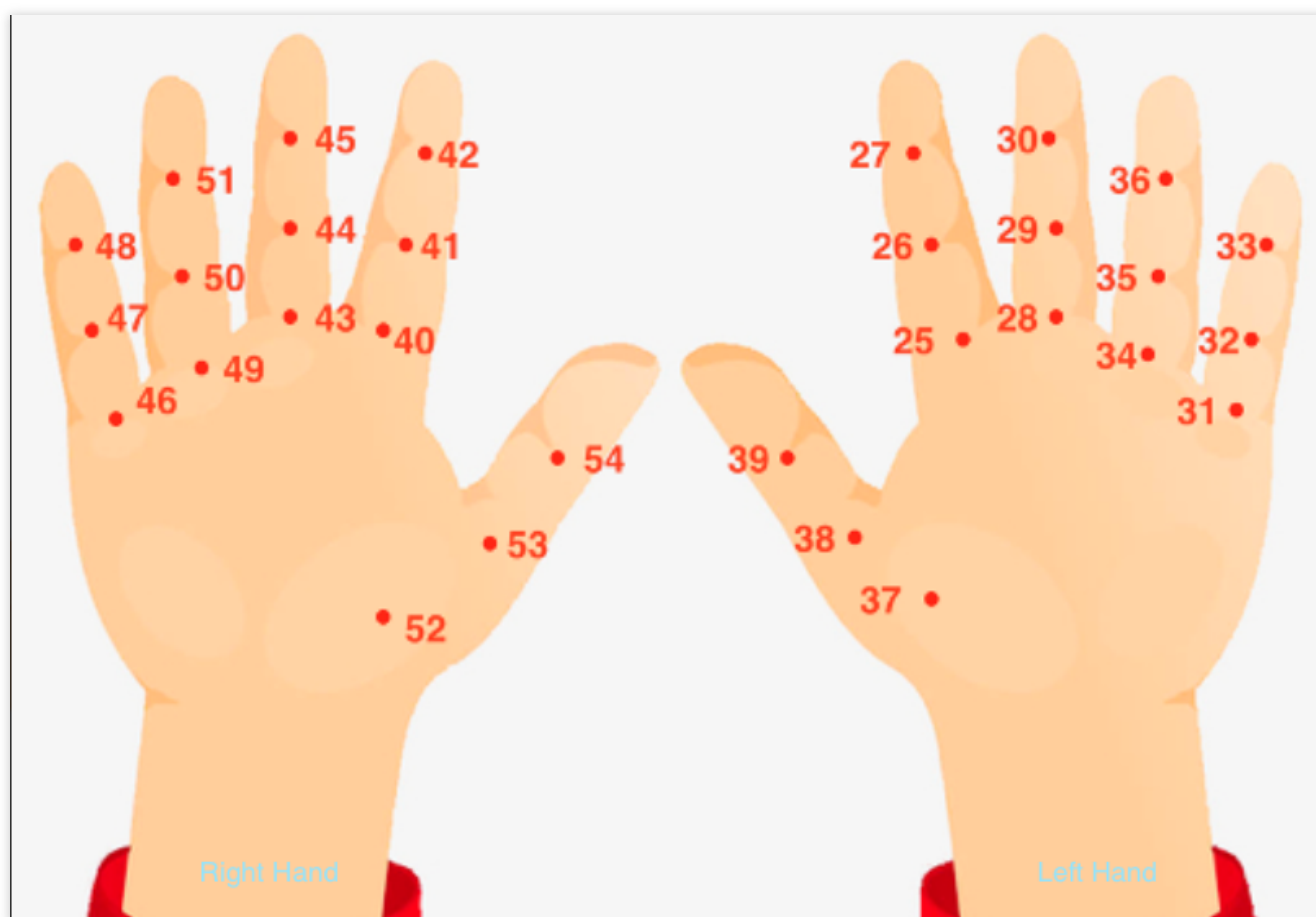
For the meanings of fields in `body_3d_info`, see below.

Body Keypoints and Descriptions

SMPL keypoints



SMPL-X hand keypoints



Here is an example of the JSON string output by the SDK:

```
{
  "face_info": [{ ... }],
  "body_3d_info": {
    "imageHeight": 652,
    "imageWidth": 320,
    "items": [{
      "index": 1,
      "pose": [-0.014862474985420704, 0.72017294, ...],
      "position_x": [150.62857055664062, 190.150, ...],
      "position_y": [605.454833984375, 655.21258, ...],
      "position_z": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      "rotation": [{
        "data": [0.9990578889846802, 0.0258054, ...],
        ...
      }],
      ...
    }]
  }
}
```

The following section details the fields in `body_3d_info` :

`imageWidth` and `imageHeight` : The width and height of images sent to the SDK.

`items` : An array. Currently, it contains only one element.

`index` : A reserved field. You can ignore it.

`pose` :

1. Location [0,2]. The 3D location (xyz) of the body root bone with the camera at the center.
2. Location [3,12]. The body shape. It includes 10 floating numbers, which are based on 10 meshes of SMPL.
3. Location [13]. The focal length, which is 5000.
4. Location [14,29]. The OpenGL projection matrix in a 3D space, which is generated based on the focal length. A 4 x 4 projection matrix is calculated as follows in the algorithm:



```
matrix={
    2 * focal_length / img_wid, 0, 0, 0,
    0, 2 * focal_length / img_hei, 0,0,
    0,0, (zf + zn) / (zn - zf), -1,
    0, 0, (2.0f * zf * zn) / (zn - zf), 0};
}
```

5. Location [30,33]. Whether the left toe, left heel, right toe, and right heel are on the ground.

position_x,position_y,position_z :

1. Location [0,23]. 2D body keypoints (figure 1 above). The value of `position_z` for all 2D keypoints is `0` .

2. Location [24,47]. 3D body keypoints (figure 1 above).

rotation

1. Location [0,23]. The body bone rotation quaternions (wxyz).

2. Location [25,54]. The hand bone rotation quaternions (wxyz). There are 15 quaternions for each hand.

Bone Names

No.	Bone Name	Bone Name 2
0	"pelvis",	"Hips"
1	"left_hip",	"LeftUpLeg"
2	"right_hip",	"RightUpLeg"
3	"spine1",	"Spine"
4	"left_knee",	"LeftLeg"
5	"right_knee",	"RightLeg"
6	"spine2",	"Spine1"
7	"left_ankle",	"LeftFoot"
8	"right_ankle",	"RightFoot"
9	"spine3",	"Spine2"
10	"left_foot",	""
11	"right_foot",	""
12	"neck",	"Neck"
13	"left_collar",	"LeftShoulder"
14	"right_collar",	"RightShoulder"
15	"head",	"Head"
16	"left_shoulder",	"LeftArm"
17	"right_shoulder",	"RightArm"
18	"left_elbow",	"LeftForeArm"
19	"right_elbow",	"RightForeArm"
20	"left_wrist",	"LeftHand"
21	"right_wrist",	"RightHand"
22	"left_hand"	""
23	"right_hand"	""
25	"left_index1"	IndexFinger1_L
26	"left_index2"	IndexFinger2_L
27	"left_index3"	IndexFinger3_L
28	"left_middle1"	MiddleFinger1_L
29	"left_middle2"	MiddleFinger2_L
30	"left_middle3"	MiddleFinger3_L
31	"left_pinky1"	PinkyFinger1_L
32	"left_pinky2"	PinkyFinger2_L
33	"left_pinky3"	PinkyFinger3_L

34	"left_ring1"	RingFinger1_L
35	"left_ring2"	RingFinger2_L
36	"left_ring3"	RingFinger3_L
37	"left_thumb1"	ThumbFinger1_L
38	"left_thumb2"	ThumbFinger2_L
39	"left_thumb3"	ThumbFinger3_L
40	"right_index1"	IndexFinger1_R
41	"right_index2"	IndexFinger2_R
42	"right_index3"	IndexFinger3_R
43	"right_middle1"	MiddleFinger1_R
44	"right_middle2"	MiddleFinger2_R
45	"right_middle3"	MiddleFinger3_R
46	"right_pinky1"	PinkyFinger1_R
47	"right_pinky2"	PinkyFinger2_R
48	"right_pinky3"	PinkyFinger3_R
49	"right_ring1"	RingFinger1_R
50	"right_ring2"	RingFinger2_R
51	"right_ring3"	RingFinger3_R
52	"right_thumb1"	ThumbFinger1_R
53	"right_thumb2"	ThumbFinger2_R
54	"right_thumb3"	ThumbFinger3_R

Converting Audio to Expressions

Converting Audio to Expression

Last updated : 2023-02-27 12:20:01

Overview

This capability processes audio data and outputs blendshape data that meets the standards of Apple ARKit. For details, see [ARFaceAnchor](#). You can pass the data to Unity to drive your model or use the data to implement other features.

Integration

Method 1: Integrate the Tencent Effect SDK

The capability of converting audio to expressions is built into the Tencent Effect SDK, so to use the capability, you can integrate the Tencent Effect SDK.

1. Download the complete edition of the [Tencent Effect SDK](#).
2. Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the SDK.

Method 2: Integrate the Audio-to-Expression SDK

If you do not need other Tencent Effect capabilities, you can integrate the Audio-to-Expression SDK, the AAR file of which is about 6 MB. You can contact us to get the SDK.

Directions

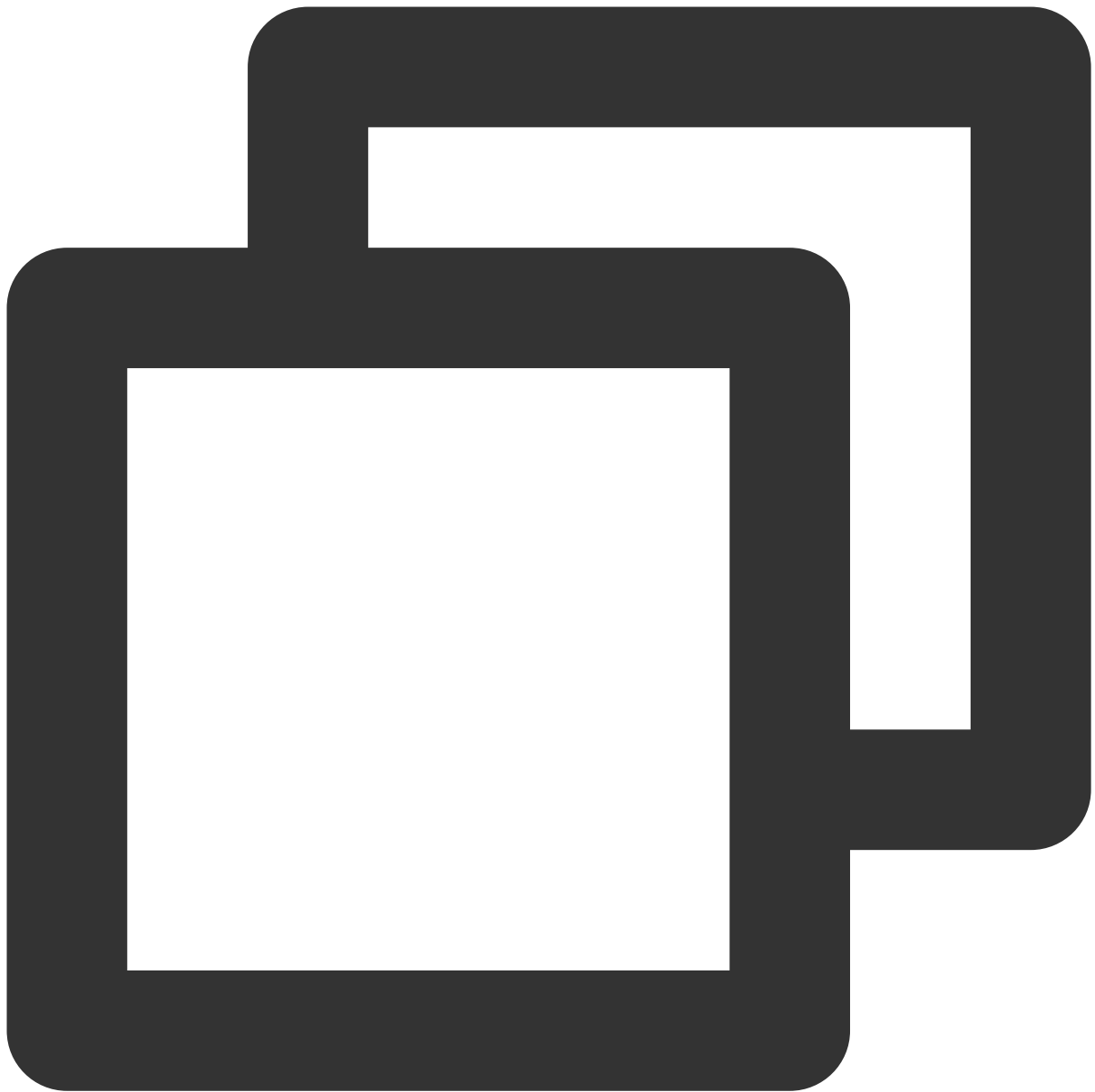
1. **Set the license.** For detailed directions, see [Integrating Tencent Effect SDK - Step 1. Authenticate](#).
2. **Configure the model files:** Copy the model files from `assets` to a private directory of your application, such as `context.getFilesDir() + "/my_models_dir/audio2exp"`. Then, call the `init(String modelPath)` API of `Audio2ExpApi`, passing in `context.getFilesDir() + "/my_models_dir"`. You can find the model files at the following location of the SDK package:



APIs

API	Description
<pre>public int Audio2ExpApi.init(String modelPath);</pre>	Initializes the SDK. You need to pass in the path of the model files. 0 indicate
<pre>public float[] Audio2ExpApi.parseAudio(float[] inputData);</pre>	The input is audio, which must be one-channel and have a sample rate of 16 K Apple {"eyeBlinkLeft", "eyeLookDownLeft", "eyeLookInLeft", "e
<pre>public int Audio2ExpApi.release();</pre>	Releases resources. Call this API when you no longer need the capability.

Integration Code Sample



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    findViewById(R.id.button).setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View view) {
            TELicenseCheck.getInstance().setTELICENSE(MainActivity.this)
        }
        @Override
        public void onLicenseCheckFinish(int errorCode, String message) {
            Log.d(TAG, "onLicenseCheckFinish: errorCode=" + errorCode + ", message=" + message);
        }
    });
}
```



```
        if (errorCode == TELicenseCheck.ERROR_OK) {
            //license check success
            Audio2ExpApi audio2ExpApi = new Audio2ExpApi();
            int err = audio2ExpApi.init(MainActivity.this);
            Log.d(TAG, "onLicenseCheckFinish: errorCode=" + errorCode);
            //TODO start record and parse audio
        } else {
            // license check failed
        }
    }
}

});
}

});
}
```

Note

For the full code sample, see [Demos](#).

For audio recording, see `com.tencent.demo.avatar.audio.AudioCapturer`.

For more information on how to use the APIs, see

`com.tencent.demo.avatar.activity.Audio2ExpActivity` and its related classes.

iOS

Last updated : 2023-02-27 12:15:21

Overview

This capability processes audio data and outputs blendshape data that meets the standards of Apple ARKit. For details, see [ARFaceAnchor](#). You can pass the data to Unity to drive your model or use the data to implement other features.

Integration

Method 1: Integrate the Tencent Effect SDK

1. The capability of converting audio to expressions is built into the Tencent Effect SDK, so to use the capability, you can integrate the Tencent Effect SDK.
2. Download the complete edition of the [Tencent Effect SDK](#).
3. Follow the directions in [Integrating Tencent Effect SDK](#) to integrate the SDK.
4. Import `Audio2Exp.framework` in the [SDK](#) to your project. Select your target, under the **General** tab, find **Frameworks,Libraries, and Embedded Content**, and set `Audio2Exp.framework` to **Embed & Sign**.

Method 2: Integrate the Audio-to-Expression SDK

If you only need the capability of converting audio to expressions, you can integrate the Audio-to-Expression SDK (`Audio2Exp.framework` is about 7 MB). Import the two dynamic frameworks `Audio2Exp.framework` and `YTCommonXMagic.framework` to your project. Select your target, under the **General** tab, find **Frameworks,Libraries, and Embedded Content**, and set `Audio2Exp.framework` and `YTCommonXMagic.framework` to **Embed & Sign**.

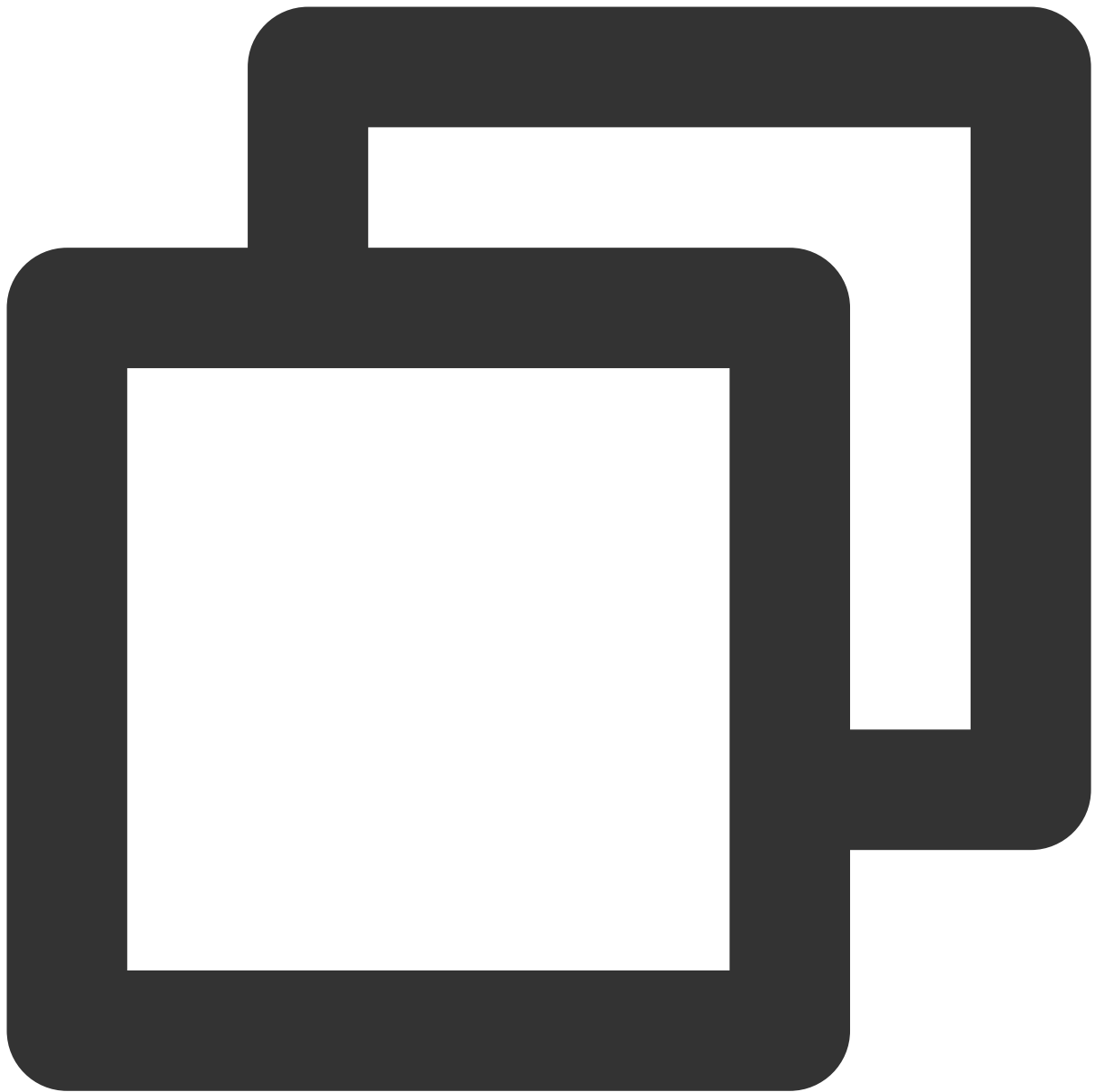
Directions

1. Set the license. For detailed directions, see [Integrating Tencent Effect SDK - Step 1. Authenticate](#).
2. Configure the model file. Copy the model file `audio2exp.bundle` to your project directory. When calling `initWithModelPath:` of `Audio2ExpApi`, pass in the path of model file.

APIs

API	Description
<code>+ (int)initWithModelPath:(NSString*)modelPath;</code>	Initializes the SDK. You need to pass in the path of the model file. <code>0</code> indicates the initialia
<code>+ (NSArray*)parseAudio:(NSArray*)inputData;</code>	The input is audio, which must be one-channel and have a sample rate of 16 Kbps and a Apple {"eyeBlinkLeft","eyeLookDownLeft","eyeLookInLeft","eyeLookOutLeft","eyeLookUp
<code>+ (int)releaseSdk</code>	Releases resources. Call this API when you no longer need the capability.

Integration Code Sample



```
// Initialize the Audio-to-Expression SDK
NSString *path = [[NSBundle mainBundle] pathForResource:@"audio2exp" ofType:@"bundl
int ret = [Audio2ExpApi initWithModelPath:path];
// Convert audio to blendshape data
NSArray *emotionArray = [Audio2ExpApi parseAudio:floatArr];
// Release the SDK
[Audio2ExpApi releaseSdk];

// Use with the Tencent Effect SDK
// Initialize the SDK
self.beautyKit = [[XMagic alloc] initWithRenderSize:previewSize assetsDict:assetsDi
```

```
// Load the avatar materials
[self.beautyKit loadAvatar:bundlePath exportedAvatar:nil completion:nil];
// Pass the blendshape data to the SDK, and the effects will be applied.
[self.beautyKit updateAvatarByExpression:emotionArray];
```

Note:

For the full code sample, see [Demos](#).

For audio recording, see `TXCAudioRecorder` .

For more information on using the APIs, see `VoiceViewController` and the related classes.