

# **Tencent Effect SDK**

## **API Documentation**

### **Product Documentation**



## Copyright Notice

©2013-2022 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

 Tencent Cloud

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

## API Documentation

iOS

Android

Flutter

# API Documentation

## iOS

Last updated : 2022-07-22 12:04:37

This document describes core APIs ( `XMagic.h` ) of the Tencent Effect SDK, which you can use to initialize the SDK, change beauty filter parameters, and call animated effects.

### Public Member APIs

API	Description
<a href="#">initWithRenderSize</a>	Initialization API
<a href="#">initWithGITexture</a>	Initialization API
<a href="#">configPropertyWithType</a>	Configures effects.
<a href="#">setRenderSize</a>	Sets the render size.
<a href="#">deinit</a>	Releases resources.
<a href="#">process</a>	Processes data.
<a href="#">processUIImage</a>	Processes an image.
<a href="#">getConfigPropertyWithName</a>	Gets effect information.
<a href="#">registerLoggerListener</a>	Registers a log listener.
<a href="#">registerSDKEventListener</a>	Register a listener for SDK events.
<a href="#">clearListeners</a>	Removes listeners.
<a href="#">getCurrentGLContext</a>	Gets the current OpenGL context.
<a href="#">onPause</a>	Pauses the SDK.
<a href="#">onResume</a>	Resumes the SDK.

#### **initWithRenderSize**

This API is used for initialization.

```
- (instancetype _Nonnull) initWithRenderSize: (CGSize) renderSize
assetsDict: (NSDictionary* _Nullable) assetsDict;
```

### Parameters

Parameter	Description
renderSize	The render size.
assetsDict	The resource dictionary.

### initWithGLTexture

This API is used for initialization.

```
- (instancetype _Nonnull) initWithGLTexture: (unsigned) textureID
width: (int) width
height: (int) height
flipY: (bool) flipY
assetsDict: (NSDictionary* _Nullable) assetsDict;
```

### Parameters

Parameter	Description
textureID	The texture ID.
width	The render size.
height	The render size.
flipY	Whether to flip the image.
assetsDict	The resource dictionary.

### configPropertyWithType

This API is used to configure effects.

```
- (int) configPropertyWithType: (NSString* _Nonnull) propertyType withName: (NSString*
_Nonnull) propertyName withData: (NSString* _Nonnull) propertyValue withExtraInfo: (i
d _Nullable) extraInfo;
```

### Parameters

Parameter	Description
propertyType	The effect type.
propertyName	The effect name.
propertyValue	The effect value.
extraInfo	A reserved parameter, which can be used for dictionary configuration.

## Examples

- **Beautification:** Configuring the skin brightening effect

```
NSString *propertyType = @"beauty"; //Set the effect type
NSString *propertyName = @"beauty.whiten"; //Specify the effect name
NSString *propertyValue = @"60"; //Set the effect value
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withD
ata:propertyValue withExtraInfo:nil];
```

- **Filter:** Configuring the Allure filter

```
NSString *propertyType = @"lut"; //Set the effect type
NSString *propertyName = [@"lut.bundle/" stringByAppendingPathComponent:@"xindo
ng_lf.png"]; //Specify the effect name
NSString *propertyValue = @"60"; //Set the effect value
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withD
ata:propertyValue withExtraInfo:nil];
```

- **Body retouch:** Configuring the long leg effect

```
NSString *propertyType = @"body"; //Set the effect type
NSString *propertyName = @"body.legStretch"; //Specify the effect name
NSString *propertyValue = @"60"; //Set the effect value
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withD
ata:propertyValue withExtraInfo:nil];
```

- **Animated effect:** Configuring the animated 2D cute effect

```
NSString *motion2dResPath = [[NSBundle mainBundle] pathForResource:@"2dMotionRe
s" ofType:@"bundle"]; //The absolute path of the `2dMotionRes` folder
```

```

NSString *propertyType = @"motion"; //Set the effect type
NSString *propertyName = @"video_keaituya"; //Specify the effect name
NSString *propertyValue = motion2dResPath; //Set the path of the animated effect
t
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];

```

- **Makeup:** Configuring the girl group makeup effect

```

NSString *motionMakeupResPath = [[NSBundle mainBundle] pathForResource:@"makeupMotionRes" ofType:@"bundle"]; //The absolute path of the `makeupMotionRes` folder
NSString *propertyType = @"motion"; //Set the effect type
NSString *propertyName = @"video_nvtuanzhuang"; //Specify the effect name
NSString *propertyValue = motionMakeupResPath; //Set the path of the animated effect
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
//Below are settings for the makeup effect (you only need to configure the above parameters once and can change the following settings multiple times)
NSString *propertyTypeMakeup = @"custom"; //Set the effect type
NSString *propertyNameMakeup = @"makeup.strength"; //Specify the effect name
NSString *propertyValueMakeup = @"60"; //Set the effect value
[self.xmagicApi configPropertyWithType:propertyTypeMakeup withName:propertyNameMakeup withData:propertyValueMakeup withExtraInfo:nil];

```

- **Keying:** Configuring the background blurring effect (strong)

```

NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotionRes" ofType:@"bundle"]; //The absolute path of the `segmentMotionRes` folder
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, @"timeOffset": @0}, @"icon":@"segmentation.linjian.png"}; //Configure the reserved parameter
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:dic];

```

- **Custom background:**

```

NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotionRes" ofType:@"bundle"]; //The absolute path of the `segmentMotionRes` folder
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-8D3A-F6028D185682/Documents/MediaFile/image.png"; //The absolute path of the background image or video (after compression)
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 is 0; if a video is used, its value is the video length.
NSDictionary *dic = @{@"bgName": imagePath, @"bgType": @(bgType), @"timeOffset": @(timeOffset)}, @{@"icon": @"segmentation.linjian.png"}; //Configure the reserved parameter
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:dic];

```

## setRenderSize

This API is used to set the render size.

```
- (void) setRenderSize: (CGSize) size;
```

### Parameters

Parameter	Description
size	The render size.

## deinit

This API is used to release resources.

```
- (void) deinit;
```

## Process

This API is used to process data.

```
- (YTProcessOutput* _Nonnull) process: (YTProcessInput * _Nonnull) input;
```

### Parameters



Parameter	Description
input	The input data.

## processUIImage

This API is used to process an image.

```
- (UIImage* _Nullable)processUIImage:(UIImage* _Nonnull)inputImage needReset:(boolean)needReset;
```

### Parameters

Parameter	Description
inputImage	The input image. If your image is larger than 2160 x 4096, we recommend you reduce its size before passing it in; otherwise, face recognition may fail or may be inaccurate. It may also cause an OOM error.
needReset	This parameter must be set to <code>true</code> in the following cases: <ul style="list-style-type: none"> <li>The image processed is changed</li> <li>The first time a keying effect is used</li> <li>The first time an animated effect is used</li> <li>The first time a makeup effect is used</li> </ul>

## getConfigPropertyWithName

This API is used to get effect information.

```
- (YTBeautyPropertyInfo* _Nullable)getConfigPropertyWithName:(NSString* _Nonnull)propertyName;
```

### Parameters

Parameter	Description
propertyName	The effect name.

## registerLoggerListener

This API is used to register a log listener.

```
- (void)registerLoggerListener:(id<YTSDKLogListener> _Nullable)listener withDefaultLevel:(YTSDKLogLevel)level;
```

## Parameters

Parameter	Description
listener	The log callback.
level	The log output level, which is ERROR by default.

## registerSDKEventListener

This API is used to register a listener for SDK events.

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

## Parameters

Parameter	Description
listener	The listener for SDK events, including AI events, tips, and resource events.

## clearListeners

This API is used to remove listeners.

```
- (void)clearListeners;
```

## getCurrentGLContext

This API is used to get the current OpenGL context.

```
- (nullable EAGLContext*)getCurrentGLContext;
```

## onPause

This API is used to pause the SDK.

```
/// @brief When your app is switched to the background, you need to call this API  
to pause the SDK  
- (void)onPause;
```

## onResume

This API is used to resume the SDK.

```

/// @brief When your app is switched back to the foreground, you need to call this API to resume the SDK
- (void) onResume;

```

## Static APIs

API	Description
<a href="#">isBeautyAuthorized</a>	Gets the authorization information of an effect parameter.

### isBeautyAuthorized

This API is used to get the authorization information of an effect parameter.

```

/// @param featureId: The effect parameter.
/// @return: The authorization information of the effect parameter.
+ (BOOL) isBeautyAuthorized: (NSString * _Nullable) featureId;

```

## Callback APIs

API	Description
<a href="#">YTSDKEventListener</a>	The SDK event callback.
<a href="#">YTSDKLogListener</a>	The log callback.

### YTSDKEventListener

The callback for the internal events of the SDK.

```
@protocol YTSDKEventListener <NSObject>
```

### Member callback APIs

Return Type	Callback
void	<a href="#">onYTDataEvent</a>
void	<a href="#">onAIEvent</a>

Return Type	Callback
void	<a href="#">onTipsEvent</a>
void	<a href="#">onAssetEvent</a>

## Callback description

### onYTDataEvent

The YTDataUpdate event callback.

```
/// @param event: Callback in NSString* format
- (void)onYTDataEvent:(id _Nonnull)event;
```

The information of up to five faces is returned as JSON strings.

```
{
  "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
  },
  ...
]
```

## Field Description

Field	Type	Value Range	Remarks
-------	------	-------------	---------

Field	Type	Value Range	Remarks
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]	The visibility of the 256 facial keypoints.
out_of_screen	bool	true/false	Whether only part of the face is captured.
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.

**onAIEvent**

The callback for AI events.

```
/// @param event: Callback in dict format
- (void)onAIEvent:(id _Nonnull)event;
```

**onTipsEvent**

The callback for tips.

```
/// @param event: Callback in dict format
- (void)onTipsEvent:(id _Nonnull)event;
```

**onAssetEvent**

The callback for resource events.

```
/// @param event: Callback in string format
- (void)onAssetEvent:(id _Nonnull)event;
```

## YTSDKLogListener

The log callback.

```
@protocol YTSDKLogListener <NSObject>
```

### Member callback APIs

Return Type	API
void	onLog

### Callback description

#### onLog

The log callback.

```
/// @param loggerLevel: The current log level.
/// @param logInfo: The log information.
- (void)onLog:(YtSDKLoggerLevel) loggerLevel withInfo:(NSString * _Nonnull) logInfo;
```

# Android

Last updated : 2022-08-12 15:17:20

This document describes core API class `XmagicApi.java` of the Tencent Effect SDK used to initialize the SDK, update beauty filter values, and call animated effects.

## Public Member Functions

API	Description
<a href="#">XmagicApi</a>	Constructor.
<a href="#">updateProperty</a>	Updates an attribute and can be called in any thread.
<a href="#">updateProperties</a>	Updates attributes and can be called in any thread.
<a href="#">setTipsListener</a>	Sets the callback function for the animated effect prompt to display the prompt on the frontend page.
<a href="#">setYTDataListener</a>	Sets the callback for data such as face point information (only available in S1-05 and S1-06 packages).
<a href="#">setAIDataListener</a>	Sets the callback of face, gesture, and body detection status.
<a href="#">onPause</a>	Pauses sound playback and can be bound to the <code>onPause</code> lifecycle of <code>Activity</code> .
<a href="#">onResume</a>	Resumes rendering and can be bound to the <code>onResume</code> lifecycle of <code>Activity</code> .
<a href="#">onDestroy</a>	Terminates xMagic, which needs to be called in the GL thread.
<a href="#">process</a>	Method to receive data by SDK rendering, which can be used within the camera data callback function.
<a href="#">onPauseAudio</a>	Call this function when you only need to stop the audio but don't need to release the GL thread.
<a href="#">sensorChanged</a>	Used to determine the current rotation angle of the phone, so as to adjust the basis for AI to recognize the angle of the face.

API	Description
<a href="#">isDeviceSupport</a>	Passes in the list of animated effect resources into the SDK for check, after which the <code>XmagicProperty.isSupport</code> field identifies whether the atomic capability is available. According to the value of <code>XmagicProperty.isSupport</code> , click restrictions can be controlled at the UI layer or removed from the resource list directly.
<a href="#">getPropertyRequiredAbilities</a>	Passes in a list of animated effect resources and returns a list of the SDK atomic capabilities used by each resource.
<a href="#">getDeviceAbilities</a>	Returns the list of atomic capabilities supported by the current device.
<a href="#">isSupportBeauty</a>	Determines whether the current model supports beauty filters (OpenGL3.0).
<a href="#">isBeautyAuthorized</a>	Determines which beauty filters are supported by the current <code>lic</code> . Only beauty filters of the <code>BEAUTY</code> and <code>BODY_BEAUTY</code> types can be checked. The check result is assigned to the <code>XmagicProperty.isAuth</code> field of each beauty filter object.
<a href="#">setXmagicStreamType</a>	Sets the input data type, which is Android camera data stream by default.
<a href="#">setXmagicLogLevel</a>	Sets the log level of the SDK. We recommend you set it to <code>Log.DEBUG</code> during developing and debugging and to <code>Log.WARN</code> after official release. If it is set to <code>Log.DEBUG</code> after release, many logs will be generated and affect the performance. <b>It should be called after <code>new XmagicApi()</code>.</b>

## Static Functions

API	Description
<a href="#">setLibPathAndLoad</a>	Sets <code>libPath</code> .

## Public Member Function Description

### XmagicApi

Constructor.

```
XmagicApi(Context context, String resDir)
XmagicApi(Context context, String resDir, OnXmagicPropertyErrorListener xmagicProp
```



```
ertyErrorListener)
```

## Parameters

Parameter	Description
Context context	Context.
String resDir	<p>Resource file directory.</p> <ul style="list-style-type: none"> <li>If the SDK resource files are built into `assets`, before using the SDK for the first time, you need to copy the resources to the app's private directory: set the resource path through <code>XmagicResParser.setResPath(new File(getFilesDir(), "xmagic").getAbsolutePath())</code> first and then copy the resources through <code>XmagicResParser.copyRes(getApplicationContext())</code>. For more information, see the <code>LaunchActivity.java</code> of the demo.</li> <li>If the SDK resource files are downloaded from the internet, after the download succeeds, set the resource path through <code>XmagicResParser.setResPath(validAssetsDirectory)</code>.</li> <li>Get the previously set path through <code>XmagicResParser.getResPath()</code>.</li> </ul>
OnXmagicPropertyErrorListener xmagicPropertyErrorListener	Error callback API.

## Error codes and descriptions:

Error Code	Description
-1	An unknown error occurred.
-100	Failed to initialize 3D SDK resources.
-200	The GAN material is not supported.
-300	The device does not support this material component.
-400	The template JSON content is empty.
-500	The SDK version is too low.
-600	Keying is not supported.
-700	OpenGL is not supported.

Error Code	Description
-800	The script is not supported.
5000	The resolution of the background image to be keyed exceeds 2160 x 3840.
5001	The memory required by background image keying is insufficient.
5002	Failed to parse the background video to be keyed.
5003	The background video to be keyed exceeds 200 seconds.
5004	The format of the background video to be keyed is not supported.

## updateProperty

Modifies a beauty filter value, animated effect, or filter and can be called in any thread.

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

### Parameters

Parameter	Description
XmagicProperty<?> p	<p>Tencent Effect SDK data entity class.</p> <ul style="list-style-type: none"> <li>Taking "skin smoothing" as an example, you can create an instance as follows: <pre>new XmagicProperty&lt;&gt;(Category.BEAUTY, null, null, BeautyConstant.BEAUTY_SMOOTH, new XmagicPropertyValues(0, 100, 50, 0, 1));</pre> </li> <li>Taking "2D animated effect Bunny" as an example, you can create an instance as follows: <pre>new XmagicProperty&lt;&gt;(Category.MOTION, "video_tutujiang", "animated effect file path", null, null);</pre> </li> </ul> <p>For more examples, see the <code>XmagicResParser.java</code> of the demo project.</p>

## updateProperties

Batch modifies beauty values, animated effects, or filters and can be called in any thread.

```
void updateProperties(List<XmagicProperty<?>> properties)
```

## Parameters

Parameter	Description
(List<XmagicProperty<?>> properties)	See the <code>updateProperty</code> method description.

## setTipsListener

Sets the callback function for the animated effect prompt to display the prompt on the frontend page; for example, some materials will prompt the user to nod, stretch out their palms, or make a finger heart.

```
void setTipsListener(XmagicApi.XmagicTipsListener effectTipsListener)
```

## Parameters

Parameter	Description
XmagicApi.XmagicTipsListener effectTipsListener	Callback function implementation class. The callback is not necessarily in the main thread.

## setYTDataListener

Sets the callback for data such as face point information.

```
void setYTDataListener(XmagicApi.XmagicYTDataListener ytDataListener)
Sets the callback for data such as face information.
public interface XmagicYTDataListener {
void onYTDataUpdate(String data)
}
```

`onYTDataUpdate` 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
},
...
]
}

```

### Field description

Field	Type	Range	Description
trace_id	int	[1,INF)	Face ID. The same ID points to the same face in the process of continuous stream fetching.
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 out of the screen.
left_eye_high_vis_ratio	float	[0,1]	Percentage of the highly visible points of the left eye.
right_eye_high_vis_ratio	float	[0,1]	Percentage of the highly visible points of the right eye.
left_eyebrow_high_vis_ratio	float	[0,1]	Percentage of the highly visible points of the left eyebrow.
right_eyebrow_high_vis_ratio	float	[0,1]	Percentage of the highly visible points of the right eyebrow.
mouth_high_vis_ratio	float	[0,1]	Percentage of the highly visible points of the mouth.

### Parameters

Parameter	Description
XmagicApi.XmagicYTDataListener ytDataListener	Callback function implementation class.

## setAIDataListener

Calls back the point information of the detected face, body parts, and gestures.

```
public interface OnAIDataListener {  
    void onFaceDataUpdated(List<FaceData> faceDataList);  
    void onHandDataUpdated(List<HandData> handDataList);  
    void onBodyDataUpdated(List<BodyData> bodyDataList);  
}
```

## onPause

Pauses rendering, can be bound to the `onPause` lifecycle of `Activity`, and internally calls `onPauseAudio` only.

```
void onPause()
```

## onResume

Resumes rendering and can be bound to the `onResume` lifecycle of `Activity`.

```
void onResume()
```

## onDestroy

Clears GL thread resources and needs to be called within the GL thread. Sample code:

```
// See the sample code in `MainActivity.java`  
glSurfaceView.queueEvent(() -> {  
    if (mXmagicApi != null) {  
        mXmagicApi.onPause();  
        mXmagicApi.onDestroy();  
    }  
});
```

```
// See the sample code in `ImageInputActivity.java`
@Override
protected void onDestroy() {
    if (mHandler != null) {
        mHandler.destroy() -> {
            if (mXmagicApi != null) {
                mXmagicApi.onPause();
                mXmagicApi.onDestroy();
            }
        });
        mHandler.waitDone();
    }
    XmagicPanelDataManager.getInstance().clearData();
    super.onDestroy();
}
}
```

## process

Method to receive data by SDK rendering, which can be used within the camera data callback function.

```
// Render the texture
int process(int srcTextureId, int srcTextureWidth, int srcTextureHeight)
// Render the bitmap
Bitmap process(Bitmap bitmap, boolean needReset){
```

## Parameters

Parameter	Description
int srcTextureId	Texture that needs to be rendered.
id int srcTextureWidth	Width of the texture that needs to be rendered.
int srcTextureHeight	Height of the texture that needs to be rendered.
Bitmap bitmap	The recommended maximum size is 2160 x 4096. Larger images have poor face recognition results or cannot get faces recognized and are likely to cause OOM problems. Shrink such images first before passing them in.

Parameter	Description
boolean needReset	Set <code>needReset</code> to <code>true</code> for the following scenarios. <ul style="list-style-type: none"> <li>• Switch image.</li> <li>• Use keying for the first time.</li> <li>• Use animated effect for the first time.</li> <li>• Use makeup for the first time.</li> </ul>

## onPauseAudio

Call this function when you only need to stop the audio but don't need to release the GL thread.

```
void onPauseAudio()
```

## sensorChanged

Used to determine the current rotation angle of the phone, so as to adjust the basis for AI to recognize the angle of the face. This needs to be called in the callback function of the G-sensor.

```
void sensorChanged(SensorEvent event, Sensor accelerometer)
```

### Parameters

Parameter	Description
SensorEvent event	Event entity class returned by the G-sensor callback function <code>onSensorChanged</code> .
Sensor accelerometer	Sample G-sensor.

## isDeviceSupport

Passes in the list of animated effect resources into the SDK for check, after which the

`XmagicProperty.isSupport` field identifies whether the material is available. According to the value of `XmagicProperty.isSupport`, click restrictions can be controlled at the UI layer or removed from the resource list directly.

```
void isDeviceSupport(List<XmagicProperty<?>> assetsList)
```

## Parameters

Parameter	Description
List<XmagicProperty<?>> assetsList	List of animated effect materials to be checked.

## getPropertyRequiredAbilities

Passes in a list of animated effect resources and returns a list of the SDK atomic capabilities used by each resource.

Use case of this method:

If you have purchased or made several animated effect materials, calling this method will return the list of atomic capabilities that each material needs to use. For example, material 1 needs to use capabilities A, B, and C, while material 2 needs to use capabilities B, C, and D. Then, you can keep this list on the server. After that, when a user wants to download the materials from the server, the user first gets the list of atomic capabilities supported by the user's phone through the `getDeviceAbilities` method (for example, the phone has capabilities A, B, and C but not D), and the list is then sent to the server. The server determines that the phone does not have capability D, so it will not deliver material 2 to the user.

## Parameters

Parameter	Description
List<XmagicProperty<?>> assets	List of animated effect resources for which to check the atomic capabilities.

## Response

Returned value `Map<xmagicproperty<?>, ArrayList<string>>` :

- key: animated effect resource material entity class.
- value: list of used atomic capabilities.

## getDeviceAbilities

Returns the list of atomic capabilities supported by the current device and needs to be used together with the

`getPropertyRequiredAbilities` method. For more information, see the

`getPropertyRequiredAbilities` description.

```
Map<String, Boolean> getDeviceAbilities()
```

## Response



Returned value `Map<string,boolean>` :

- key: atomic capability name (corresponding to the material capability name).
- value: whether it is supported by the current device.

## isSupportBeauty

Determines whether the current model supports beauty filters (OpenGL3.0).

```
boolean isSupportBeauty()
```

### Response

Returned value `boolean` : whether beauty filter is supported.

## isBeautyAuthorized

Determines which face/body beauty filters are supported by the current license. Only beauty filters of the `BEAUTY` and `BODY_BEAUTY` types can be checked. The check result is assigned to the `XmagicProperty.isAuth` field of each beauty filter object. Access to these items can be blocked on the UI if the `isAuth` field is `false`.

```
void isBeautyAuthorized(List<XmagicProperty<?>> properties)
```

### Parameters

Parameter	Description
List<XmagicProperty<?>> properties	Beauty filters that need to be checked.

## setXmagicStreamType

Sets the input data type, which is Android camera data stream (`XmagicApi.PROCESS_TYPE_CAMERA_STREAM`) by default.

```
void setXmagicStreamType(int type)
```

### Parameters

Parameter	Description
-----------	-------------

Parameter	Description
int type	Data source type. Valid values: <ul style="list-style-type: none"> <li><code>XmagicApi.PROCESS_TYPE_CAMERA_STREAM</code> : camera data source</li> <li><code>XmagicApi.PROCESS_TYPE_PICTURE_DATA</code> : image data source</li> </ul>

## Static Function Description

### setLibPathAndLoad

Sets the `so` file path and triggers loading. If the `so` file is built into `assets`, you don't need to call this method. If it is downloaded dynamically, this method needs to be called before authentication and `new XmagicApi`.

- If a null value is passed in, the `so` library will be loaded from the default path. Make sure that it is built into the APK package.
- If a non-null value is passed in, such as `data/data/package name/files/xmagic_libs`, the `so` library will be loaded from this directory.

```
static boolean setLibPathAndLoad(String path)
```

### Parameters

Parameter	Description
String path	Path to store the <code>so</code> library.

```
</string,boolean></xmagicproperty<?>
```

# Flutter

Last updated : 2022-11-30 18:02:11

`TencentEffectApi` is the core API class of the Tencent Effect Flutter SDK. It offers capabilities including setting the effect strength and applying animated effects.

## Public Member APIs

API	Description
<a href="#">initXmagic</a>	Initializes data. You need to call this API before using the Tencent Effect SDK.
<a href="#">setLicense</a>	Configures the license.
<a href="#">setXmagicLogLevel</a>	Sets the log level of the SDK. We recommend you set it to <code>Log.DEBUG</code> for debugging and <code>Log.WARN</code> for official release. If you set it to <code>Log.DEBUG</code> in a production environment, the output of a large amount of log data may affect your application's performance.
<a href="#">onResume</a>	Resumes rendering. Call this API when the page is visible.
<a href="#">onPause</a>	Pauses rendering. Call this API when the page is invisible.
<a href="#">updateProperty</a>	Updates an effect property. This API can be called in any thread.
<a href="#">setOnCreateXmagicApiErrorListener</a>	Configures the callback for creating an effect object. The callback will be triggered if an error occurs.
<a href="#">setTipsListener</a>	Configures the callback for animated effect tips. The tips can be displayed on the UI.
<a href="#">setYTDataListener</a>	Configures the callback of facial keypoints and other data. The callback is only available in S1 - 05 and S1 - 06.
<a href="#">setAIDataListener</a>	Configures the callback of face, gesture, and body detection results.
<a href="#">isBeautyAuthorized</a>	Checks whether the current license supports a particular type of effects. This API can only check the authorization of <code>BEAUTY</code> and <code>BODY_BEAUTY</code> effects. The result returned determines the value of <code>XmagicProperty.isAuth</code> .
<a href="#">isSupportBeauty</a>	Checks whether the current device supports effects (OpenGL 3.0).

API	Description
<a href="#">getDeviceAbilities</a>	Gets a list of Tencent Effect capabilities supported by the current device.
<a href="#">isDeviceSupport</a>	Checks whether a list of animated effect resources are supported. The result is indicated by <code>XmagicProperty.isSupport</code> . For unsupported resources, you can either disable tapping on the UI or delete them from the resource list.
<a href="#">getPropertyRequiredAbilities</a>	Gets the Tencent Effect capabilities used by different animated effect resources.

## API Description

### initXmagic

This API is used to initialize the Tencent Effect SDK.

```
void initXmagic(String xmagicResDir, InitXmagicCallBack callBack);
typedef InitXmagicCallBack = void Function(bool reslut);
```

### Parameters

Parameter	Description
String xmagicResDir	The resource directory.
InitXmagicCallBack callBack	The initialization callback.

### setLicense

This API is used to set the license.

```
///Set the Tencent Effect license
void setLicense(String licenseKey, String licenseUrl, LicenseCheckListener checkL
istener);
//The callback of the authorization result
typedef LicenseCheckListener = void Function(int errorCode, String msg);
```

### Parameters

Parameter	Description
String licenseKey	The license key.
String licenseUrl	The license URL.
LicenseCheckListener checkListener	The callback of the authorization result.

## setXmagicLogLevel

This API is used to set the log level of the SDK.

```
void setXmagicLogLevel(int logLevel);
```

### Parameters

Parameter	Description
int logLevel	You can set the log level using a type defined for <code>LogLevel</code> .

## onResume

This API is used to resume effect rendering.

```
void onResume();
```

## onPause

This API is used to pause effect rendering.

```
void onPause();
```

## updateProperty

This API is used to set an effect value, an animated effect, or a filter. You can call it in any thread.

```
void updateProperty(XmagicProperty xmagicProperty);
```

### Parameters

Parameter	Description
XmagicProperty xmagicProperty	The object of the effect property.

## setOnCreateXmagicApiErrorListener

This API is used to configure the callback for errors for the creation of an effect object.

```
void setOnCreateXmagicApiErrorListener(OnCreateXmagicApiErrorListener? errorListener);
/// The callback for errors for the creation of an effect object
typedef OnCreateXmagicApiErrorListener = void Function(String errorMsg, int code);
```

### Parameters

Parameter	Description
OnCreateXmagicApiErrorListener? errorListener	The callback for errors for the creation of an effect object.

Error codes:

Error Code	Description
-1	Unknown error.
-100	Failed to initialize the 3D engine.
-200	GAN materials are not supported.
-300	The device does not support this material component.
-400	The JSON template is empty.
-500	The SDK version is too old.
-600	Keying is not supported.
-700	OpenGL is not supported.
-800	The script is not supported.
5000	The resolution of the video to be keyed exceeds 2160 x 3840.
5001	Insufficient memory for keying.
5002	Failed to parse the video to be keyed.
5003	The video to be keyed is longer than 200 seconds.
5004	Unsupported video format for keying.

## setTipsListener

This API is used to configure the callback for animated effect tips. The tips can be displayed on the UI, asking users to nod, show their palms, or make finger hearts.

```
void setTipsListener(XmagicTipsListener? xmagicTipsListener);
abstract class XmagicTipsListener {
    /// Show the tip
    /// @param tips: The content of the tip (string).
    /// @param tipsIcon: The icon for the tip.
    /// @param type: The display type. If it is set to `0`, both the tip string and i
    con will be displayed. If it is set to `1`, only the icon will be displayed for P
    AG materials.
    /// @param duration: How long (milliseconds) to show the tip.
    void tipsNeedShow(String tips, String tipsIcon, int type, int duration);
    /// *
    /// Hide the tip
    /// @param tips: The content of the tip (string).
    /// @param tipsIcon: The icon for the tip.
    /// @param type: The display type. If it is set to `0`, both the tip string and i
    con will be displayed. If it is set to `1`, only the icon will be displayed for P
    AG materials.
    void tipsNeedHide(String tips, String tipsIcon, int type);
}
```

### Parameters

Parameter	Description
XmagicTipsListener xmagicTipsListener	The callback implementation class.

## setYTDataListener

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

```
/// Configure the callback of facial keypoints and other data (only available in
S1 - 05 and S1 - 06)
void setYTDataListener(XmagicYTDataListener? xmagicYTDataListener);
Configure the callback of facial keypoints and other data
abstract class XmagicYTDataListener {
    // YouTu AI data
    void onYTDataUpdate(String data);
}
```

`onYTDataUpdate` 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	Range	Description
trace_id	int	[1,INF)	The face ID. If the faces obtained from a continuous 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]	The visibility of the 256 facial keypoints.
out_of_screen	bool	true/false	Whether only part of the face is captured.
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.



Field	Type	Range	Description
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

Parameter	Description
XmagicYTDataListener xmagicYTDataListener	The callback implementation class.

## setAIDataListener

This API is used to configure the callback of face, gesture, and body detection results.

```
void setAIDataListener(XmagicAIDataListener? aiDataListener);

abstract class XmagicAIDataListener {
void onFaceDataUpdated(String faceDataList);
void onHandDataUpdated(String handDataList);
void onBodyDataUpdated(String bodyDataList);
}
```

## isBeautyAuthorized

This API is used to check whether the current license supports a particular type of effects. It can only check the authorization of `BEAUTY` and `BODY_BEAUTY` effects. The result returned determines the value of `XmagicProperty.isAuth`. If `isAuth` is `false`, you can disable the corresponding effects on the UI.

```
Future<List<XmagicProperty>> isBeautyAuthorized(
List<XmagicProperty> properties);
```

## Parameters

Parameter	Description
List<XmagicProperty> properties	The type of effects to check.

## isSupportBeauty

This API is used to check whether the current device supports effects (OpenGL 3.0).

```
Future<bool> isSupportBeauty();
```

### Response

A Boolean value indicating whether effects are supported.

## getDeviceAbilities

This API is used to get a list of Tencent Effect capabilities supported by the current device. You can use it together with `getPropertyRequiredAbilities`.

```
Future<Map<String, bool>> getDeviceAbilities();
```

### Response

```
Map<string,boolean> :
```

- key: The name of a capability (the material name).
- value: Whether the current device supports the capability.

## getPropertyRequiredAbilities

This API is used to get the Tencent Effect capabilities used by different animated effect resources.

Use case:

This API is useful if you have purchased animated effects or made your own animated effect materials. It returns the capabilities each material uses. For example, material 1 uses capabilities A, B, and C, and material 2 relies on capabilities B, C, and D. You can store such information in the server. When a user downloads the two materials from the server, call `getDeviceAbilities` first to get the capabilities supported by their device. The result is then passed to the server. For example, if a user's device supports capabilities A, B, and C, but not D, the server will not provide material 2 to the user.

```
Future<Map<XmagicProperty, List<String>?>> getPropertyRequiredAbilities (
    List<XmagicProperty> assetsList);
```

### Parameters

Parameter	Description
List<XmagicProperty> assetsList	A list of the animated effects to check.

## Response

Map<XmagicProperty, List<String>?>:

- key: The entity class of the animated effect.
- value: A list of capabilities used by the effect.

## isDeviceSupport

This API is used to check whether a list of animated effect resources are supported. The result is indicated by `XmagicProperty.isSupport` . For unsupported resources, you can either disable tapping on the UI or delete them from the resource list.

```
Future<List<XmagicProperty>> isDeviceSupport (List<XmagicProperty> assetsList);
```

## Parameters

Parameter	Description
List<XmagicProperty> assetsList	A list of animated effect resources to check.