

Tencent Effect SDK

APIドキュメント

製品ドキュメント



Tencent Cloud

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.

カタログ：

APIドキュメント

iOS

Android

Flutter

APIドキュメント

iOS

最終更新日： : 2022-08-12 15:14:58

Tencent Effect SDKのコアインターフェースクラス `XMagic.h` は、SDKの初期化、美顔の値の更新、モーションの呼出しなどの機能に使用されます。

Publicメンバー関数

API	説明
initWithRenderSize	初期化インターフェース
initWithGITexture	初期化インターフェース
configPropertyWithType	各美顔エフェクトを設定します
emitBlurStrengthEvent	後処理のぼかし強度を設定します（すべてのぼかしコンポーネントに機能）
setRenderSize	<code>renderSize</code> を設定します
deinit	リソースを解放するインターフェース
process	データを処理するインターフェース
processUIImage	画像を処理します
getConfigPropertyWithName	美顔パラメーターの設定情報を取得します
registerLoggerListener	ログを登録するインターフェース
registerSDKEventListener	SDKのイベント監視インターフェース
clearListeners	クリア用コールバックを登録するインターフェース
getCurrentGLContext	現在のGLのコンテキストを取得するインターフェース
onPause	SDKの一時停止インターフェース
onResume	SDKの再開インターフェース

initWithRenderSize

初期化インターフェース

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

パラメータ

パラメータ	意味
renderSize	レンダリングサイズ
assetsDict	アセット Dict

initWithGLTexture

初期化インターフェース

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

パラメータ

パラメータ	意味
textureID	テクスチャID
width	レンダリングサイズ
height	レンダリングサイズ
flipY	画像を反転しますか
assetsDict	アセット Dict

configPropertyWithType

美顔の各エフェクトを設定します

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

パラメータ

パラメータ	意味
propertyType	エフェクトタイプ
propertyName	エフェクト名
propertyValue	エフェクトの値
extraInfo	リザーブド拡張、オプションナル設定項目dictあり

美顔エフェクトの設定例

- **美顔**：美白エフェクトを設定します

```
NSString *propertyType = @"beauty"; //美顔のエフェクトタイプを設定します。ここでは美顔を例とします
NSString *propertyName = @"beauty.whiten"; //美顔の名前を設定します。ここでは美白を例とします
NSString *propertyValue = @"60"; //美白のエフェクトの値を設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
```

- **フィルター**：ときめきエフェクトを設定します

```
NSString *propertyType = @"lut"; //美顔のエフェクトタイプを設定します。ここではフィルターを例とします
NSString *propertyName = [@"lut.bundle/" stringByAppendingPathComponent:@"xindong_lf.png"]; //美顔の名前を設定します。ここではときめきを例とします
NSString *propertyValue = @"60"; //フィルターのエフェクトの値を設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
```

- **美ボディ**：足長エフェクトを設定します

```
NSString *propertyType = @"body"; //美顔のエフェクトタイプを設定します。ここでは美ボディを例とします
NSString *propertyName = @"body.legStretch"; //美顔の名前を設定します。ここでは足長を例とします
NSString *propertyValue = @"60"; //足長のエフェクトの値を設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
```

- **モーション**：2Dモーションの可愛い落書きエフェクトを設定します

```
NSString *motion2dResPath = [[NSBundle mainBundle] pathForResource:@"2dMotionRes" ofType:@"bundle"]; //ここでは2dMotionResフォルダーの絶対パスを設定します
NSString *propertyType = @"motion"; //美顔のエフェクトタイプを設定します。ここではモーションを例とします
NSString *propertyName = @"video_keaituya"; //美顔の名前を設定します。ここでは2Dアニメーションの可愛い落書きを例とします
NSString *propertyValue = motion2dResPath; //モーションのパスを設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
```

- **メイク**：女性アイドルグループ風メイクエフェクトを設定します

```
NSString *motionMakeupResPath = [[NSBundle mainBundle] pathForResource:@"makeupMotionRes" ofType:@"bundle"]; //ここではmakeupMotionResフォルダーの絶対パスを設定します
NSString *propertyType = @"motion"; //美顔のエフェクトタイプを設定します。ここではメイクを例とします
NSString *propertyName = @"video_nvtuanzhuang"; //美顔の名前を設定します。ここでは女性アイドルグループ風メイクを例とします
NSString *propertyValue = motionMakeupResPath; //モーションのパスを設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:nil];
//設定するメイクの値を以下に示します（前述したモーションは1回だけ呼び出せばよいです。以下に設定するメイクの値は複数回呼び出すことができます）
NSString *propertyTypeMakeup = @"custom"; //美顔のエフェクトタイプを設定します。ここではメイクを例とします
NSString *propertyNameMakeup = @"makeup_strength"; //美顔の名前を設定します。ここでは女性アイドルグループ風メイクを例とします
NSString *propertyValueMakeup = @"60"; //メイクのエフェクトの値を設定します
[self.xmagicApi configPropertyWithType:propertyTypeMakeup withName:propertyNameMakeup withData:propertyValueMakeup withExtraInfo:nil];
```

- **分割**：背景のぼかし（強い効果）を設定します

```
NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotionRes" ofType:@"bundle"]; //ここではsegmentMotionResフォルダーの絶対パスを設定します
NSString *propertyType = @"motion"; //美顔のエフェクトタイプを設定します。ここでは分割を例とします
NSString *propertyName = @"video_segmentation_blur_75"; //美顔の名前を設定します。ここでは背景のぼかし-強を例とします
NSString *propertyValue = motionSegResPath; //モーションのパスを設定します
```

```
NSDictionary *dic = @{@"bgName":@"BgSegmentation.bg.png", @"bgType":@0, @"timeOffset": @0},@"icon":@"segmentation.linjian.png"}; //リザーブドフィールドを設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:dic];
```

• カスタム背景：

```
NSString *motionSegResPath = [[NSBundle mainBundle] pathForResource:@"segmentMotionRes" ofType:@"bundle"]; //ここではsegmentMotionResフォルダーの絶対パスを設定します
NSString *propertyType = @"motion"; //美顔のエフェクトタイプを設定します。ここでは分割を例とします
NSString *propertyName = @"video_empty_segmentation"; //美顔の名前を設定します。ここではカスタム背景を例とします
NSString *propertyValue = motionSegResPath; //モーションのパスを設定します
NSString *imagePath = @"/var/mobile/Containers/Data/Application/06B00BBC-9060-450F-8D3A-F6028D185682/Documents/MediaFile/image.png"; //カスタム背景として使用する画像の絶対パス。カスタム背景としてビデオを選択した場合、ビデオに対して圧縮しトランスコーディングした後の絶対パスを使用します。
int bgType = 0; //カスタム背景のタイプ0は画像、1はビデオを表します
int timeOffset = 0; //時間の長さ背景が画像の場合、0とします。背景がビデオの場合、ビデオの長さとしてします
NSDictionary *dic = @{@"bgName":imagePath, @"bgType":@(bgType), @"timeOffset":@(timeOffset)},@"icon":@"segmentation.linjian.png"}; //リザーブドフィールドを設定します
[self.xmagicApi configPropertyWithType:propertyType withName:propertyName withData:propertyValue withExtraInfo:dic];
```

emitBlurStrengthEvent

後処理のぼかし強度を設定します（すべてのぼかしコンポーネントに機能）

```
- (void)emitBlurStrengthEvent:(int) strength;
```

パラメータ

パラメータ	意味
strength	エフェクトの値

setRenderSize

renderSizeを設定します

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

パラメータ

パラメータ	意味
size	レンダリングサイズ

deinit

リソースを解放するインターフェース |

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

process

データを処理するインターフェース |

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

パラメータ

パラメータ	意味
input	入力データ

processUIImage

画像を処理します

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

パラメータ

パラメータ	意味
inputImage	入力画像の最大サイズは2160×4096までとすることをお勧めします。このサイズを超えると、画像に対して顔認識がうまく機能できないまたは機能できないことがあり、またOOM問題も起こりやすいため、大きな画像を縮小してからアップロードすることをお勧めします
needReset	次の運用シーンでは、needResetにtrueを設定してください。 <ul style="list-style-type: none"> • 画像の切替 • 分割の初回利用 • アニメーションの初回利用 • メイクの初回利用

getConfigPropertyWithName

美颜パラメーターの設定情報を取得します

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

パラメータ

パラメータ	意味
propertyName	設定項目名

registerLoggerListener

ログを登録するインターフェース

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

パラメータ

パラメータ	意味
listener	ログをコールバックするインターフェース
level	ログの出力レベル。デフォルトではERRORとします

registerSDKEventListener

SDKのイベント監視インターフェース |

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

パラメータ

パラメータ	意味
listener	イベント監視関連のコールバック。主にAIイベント、Tipsイベント、Assetイベントに分けられています

clearListeners

クリア用コールバックを登録するインターフェース |

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

getCurrentGLContext

現在のGLのコンテキストを取得するインターフェース

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

onPause

SDKの一時停止インターフェース

```
/// @brief APPを一時停止する場合、SDKの一時停止インターフェースを呼び出します  
- (void)onPause;
```

onResume

SDKの再開インターフェース

```
/// @brief APPを再開する場合、SDKの再開インターフェースを呼び出します  
- (void)onResume;
```

静的関数

API	説明
isBeautyAuthorized	この美顔パラメーターの権限情報を取得します

isBeautyAuthorized

この美顔パラメーターの権限情報を取得します（美顔と美ボディのみをサポート）

```
/// @param featureId 美顔パラメーターを設定します  
/// @return 該当する美顔パラメーターの権限付与情報を返します  
+ (BOOL)isBeautyAuthorized:(NSString * _Nullable)featureId;
```

コールバック

API	説明
YTSDKEventListener	SDKの内部イベントコールバックインターフェース
YTSDKLogListener	ログ監視関連のコールバック

YTSDKEventListener

SDK内部イベントコールバックインターフェース

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

メンバー関数

返却値のタイプ	名前
void	onYTDataEvent
void	onAIEvent
void	onTipsEvent
void	onAssetEvent

関数説明

onYTDataEvent

YTDataUpdate イベントコールバック

```
/// @param event NSString*フォーマットのコールバック
- (void)onYTDataEvent:(id _Nonnull)event;
```

JSON string 構造体を返します。最大5人の顔情報を返します：

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

```

フィールドの意味

フィールド	タイプ	値の範囲	説明
trace_id	int	[1,INF)	顔ID。連続してストリームを取得するとき、IDが同じである場合、同じ顔として認識します
face_256_point	float	[0,screenWidth] 或 [0,screenHeight]	計512数字あり、顔の256重要特徴点があり、画面の左上隅は(0,0)です
face_256_visible	float	[0,1]	顔の256重要特徴点の可視度
out_of_screen	bool	true/false	顔が枠を超えていますか
left_eye_high_vis_ratio	float	[0,1]	左目の特徴点のうち高視認度のものが占める割合
right_eye_high_vis_ratio	float	[0,1]	右目の特徴点のうち高視認度のものが占める割合
left_eyebrow_high_vis_ratio	float	[0,1]	左眉の特徴点のうち高視認度のものが占める割合
right_eyebrow_high_vis_ratio	float	[0,1]	右眉の特徴点のうち高視認度のものが占める割合
mouth_high_vis_ratio	float	[0,1]	口の特徴点のうち高視認度のものが占める割合

onAIEvent

AIイベントコールバック

```
/// @param event dict フォーマットのコールバック
- (void)onAIEvent:(id _Nonnull)event;
```

onTipsEvent

ヒントイベントコールバック

```
/// @param event dict フォーマットのコールバック
- (void)onTipsEvent:(id _Nonnull)event;
```

onAssetEvent

アセットイベントコールバック

```
/// @param event string フォーマットのコールバック
- (void)onAssetEvent:(id _Nonnull)event;
```

YTSDKLogListener

ログ監視関連のコールバック

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

メンバー関数

返却値のタイプ	関数名
void	onLog

関数説明

onLog

ログ監視関連のコールバック

```
/// @param loggerLevel 現在のログレベルを返します
/// @param logInfo 現在のログ情報を返します
- (void)onLog:(YtSDKLogLevel) loggerLevel withInfo:(NSString * _Nonnull) logInfo;
```

Android

最終更新日：：2023-03-20 16:14:08

Tencent Effect SDKのコアインターフェースクラス `XmagicApi.java` は、SDKの初期化、美顔数値の更新、動的エフェクト呼び出しなどの機能に用いられます。

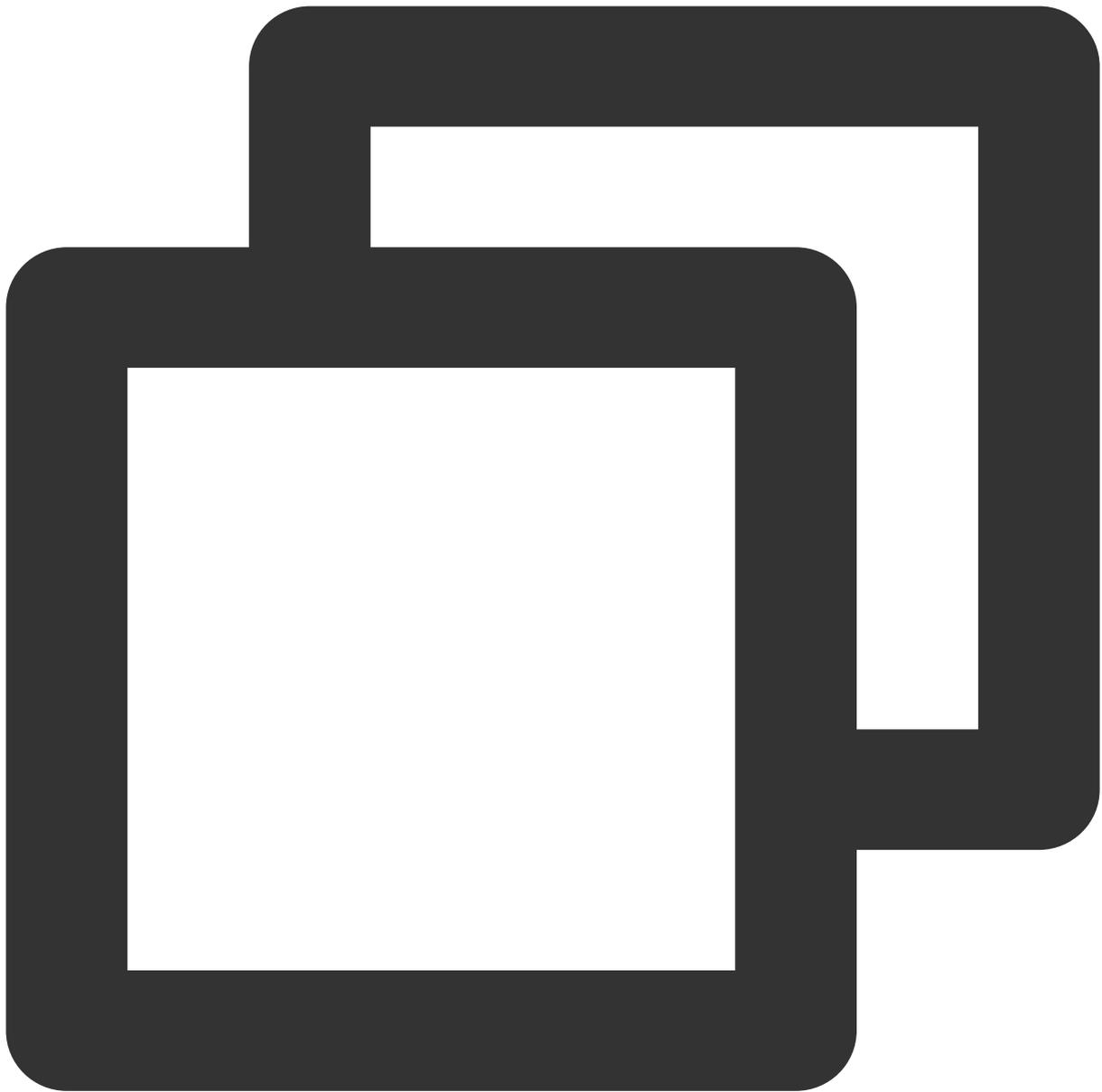
Publicメンバー関数

API	説明
<code>XmagicApi</code>	コンストラクタです。
<code>updateProperty</code>	属性の更新です。任意のスレッドで呼び出すことができます。
<code>updateProperties</code>	属性の更新です。任意のスレッドで呼び出すことができます。
<code>setTipsListener</code>	動的エフェクトプロンプトのコールバック関数を設定します。プロンプトをフロントエンドのページ上に表示するために用いられます。
<code>setYTDataListener</code>	顔の特徴点位置情報などのデータコールバックを設定します（S1-05およびS1-06パッケージのみコールバックあり）。
<code>setAIDataListener</code>	顔、ジェスチャー、体の検出状態コールバックの設定。
<code>onPause</code>	音声の再生を一時停止します。 <code>Activity</code> の <code>onPause</code> ライフサイクルにバインドできます。
<code>onResume</code>	レンダリングを再開します。 <code>Activity</code> の <code>onResume</code> ライフサイクルにバインドできます。
<code>onDestroy</code>	<code>xmagic</code> を破棄するには、GLスレッドで呼び出す必要があります。
<code>process</code>	SDKレンダリングのデータを受信するメソッドです。カメラデータのコールバック関数内で使用できます。
<code>onPauseAudio</code>	オーディオのみ停止する必要がある、GLスレッドをリリースする必要がない場合に、この関数を呼び出すことができます。
<code>sensorChanged</code>	現在のスマートフォンの回転の角度を判断し、それによってAIの顔認識の判断角度の根拠を調整するために用いられます。
<code>isDeviceSupport</code>	動的エフェクトリソースリストをSDKのチェック用に渡します。実行後、 <code>XmagicProperty.isSupport</code> フィールドは、そのアトミック機能が利用可能かどうかを表します。 <code>XmagicProperty.isSupport</code> によってUIレイ

	ヤーのクリック制限を制御することや、直接リソースリストから削除することができます。
<code>getPropertyRequiredAbilities</code>	動的エフェクトリソースリストを渡し、各リソースが使用できるSDKアトミック機能のリストを返します。
<code>getDeviceAbilities</code>	現在のデバイスでサポートされているアトミック機能のテーブルを返します
<code>isSupportBeauty</code>	現在のモデルが美顔（OpenGL3.0）をサポートしているかどうかを判断します。
<code>isBeautyAuthorized</code>	現在のlic権限がどの美顔をサポートしているかを判断します。BEAUTYおよびBODY_BEAUTYタイプの美顔項目のチェックのみサポートします。チェックした結果は、各美顔オブジェクトのXmagicProperty.isAuthフィールドに割り当てられます。
<code>setXmagicStreamType</code>	入力データタイプを設定します。デフォルトは、Android cameraデータストリームです。
<code>setXmagicLogLevel</code>	SDKのログレベルを設定します。開発デバッグの際はLog.DEBUGに、正式リリースの際はLog.WARNにそれぞれ設定することをお勧めします。正式リリースの際にLog.DEBUGに設定していると、大量のログがパフォーマンスに影響する場合があります。 new XmagicApi()の後に呼び出されます。
<code>setAudioMute</code>	動的エフェクト素材を使用するときにミュートをオンにするかどうか（V2.5.0の新機能） パラメータ：trueはミュート、falseはミュートしないことを意味します
<code>enableEnhancedMode</code>	美顔拡張モードをオンにします（V2.5.1の新機能）

XmagicApi

コンストラクタです。



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

パラメータ

パラメータ	意味
Context context	コンテキストです。
String resDir	リソースファイルディレクトリです。

	<p>SDKリソースファイルがassets内にある場合は、SDKの初回使用前にリソースをAppのプライベートディレクトリにcopyしておく必要があります。先に <code>XmagicResParser.setResPath(new File(getFilesDir(), "xmagic").getAbsolutePath())</code> によってリソースパスを設定してから、<code>XmagicResParser.copyRes(getApplicationContext())</code> によってリソースのコピーを完了させます。詳細については、Demoの <code>LaunchActivity.java</code> をご参照ください。</p> <p>SDKリソースファイルがネットワークからダウンロードしたものの場合は、ダウンロード成功後に、<code>XmagicResParser.setResPath(validAssetsDirectory)</code> によってリソースパスを設定します。 <code>XmagicResParser.getResPath()</code> によって、先に設定したパスを取得します。</p>
OnXmagicPropertyErrorListener xmagicPropertyErrorListener	エラーコールバックインターフェースです。

返されるエラーコードの意味対照表：

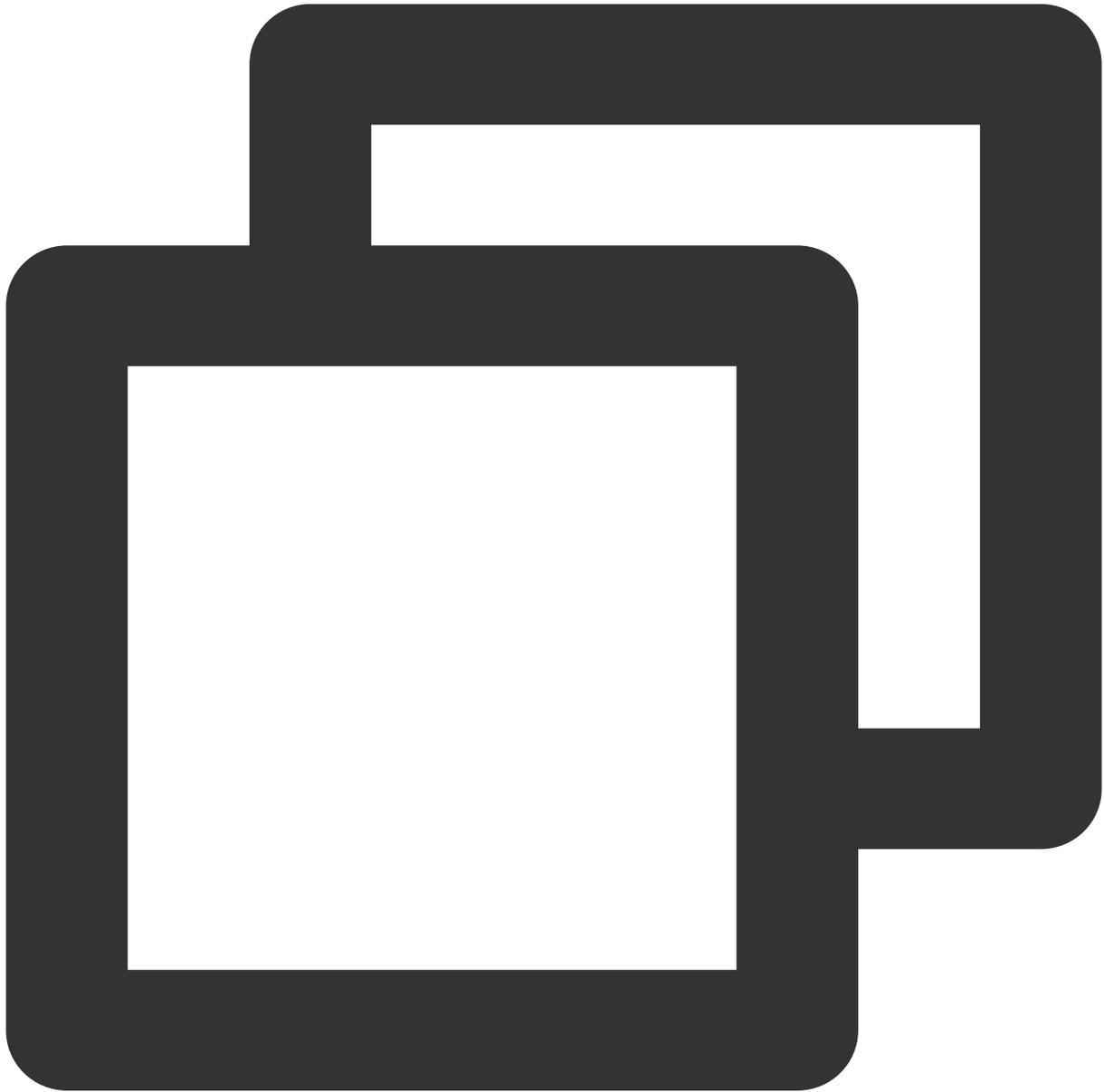
エラーコード	意味
-1	不明なエラー。
-100	3Dエンジンリソースの初期化に失敗しました。
-200	GAN素材をサポートしていません。
-300	デバイスがこの素材コンポーネントをサポートしていません。
-400	テンプレートのJSONの内容が空です。
-500	SDKのバージョンが低すぎます。
-600	分割をサポートしていません。
-700	OpenGLをサポートしていません。
-800	スクリプトをサポートしていません。
5000	分割背景画像の解像度が2160×3840を超えています。
5001	分割背景画像に必要なメモリが不足しています。
5002	分割背景ビデオの解析に失敗しました。
5003	分割背景ビデオが200秒を超えています。

5004

分割背景ビデオのフォーマットをサポートしていません。

updateProperty

ある項目の美顔数値または動的エフェクト、フィルターの変更は、任意のスレッドで呼び出せます。



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

パラメータ

パラメータ

意味

XmagicProperty<?> p	<p>Tencent Effectのデータエンティティクラス。</p> <p>「美肌」を例に取った場合、次のような方法で新しいインスタンスが作成できます。 <code>new XmagicProperty<>(Category.BEAUTY, null, null, BeautyConstant.BEAUTY_SMOOTH, new XmagicPropertyValues(0, 100, 50, 0, 1));</code></p> <p>「2D動的エフェクト兔兔醬」を例にとると、次の方式でインスタンスを新しくすることができます。 <code>new XmagicProperty<>(Category.MOTION, "video_tutujiang", "動的エフェクトのファイルパス", null, null);</code></p> <p>その他の事例については、DemoプロジェクトのXmagicResParser.javaをご参照ください。</p>
---------------------	--

XmagicProperty

美顔

属性フィールド	説明
category	Category.BEAUTY
ID	<p>null</p> <p>特殊な状況： 顔やせの（ナチュラル、女神、イケメン）ID値は、それぞれ <code>BeautyConstant.BEAUTY_FACE_NATURE_ID</code>、 <code>BeautyConstant.BEAUTY_FACE_FEMALE_GOD_ID</code>、 <code>BeautyConstant.BEAUTY_FACE_MALE_GOD_ID</code> です</p> <p>リップのID値は、 <code>XmagicConstant.BeautyConstant.BEAUTY_LIPS_LIPS_MASK</code> です</p> <p>チークのID値 は、 <code>XmagicConstant.BeautyConstant.BEAUTY_MAKEUP_MULTIPLY_MULTIPLY_MASK</code> です</p> <p>立体のID値 は、 <code>XmagicConstant.BeautyConstant.BEAUTY_SOFTLIGHT_SOFTLIGHT_MASK</code> です</p>
resPath	<p>null</p> <p>特殊な状況：リップ、チーク、立体のresPathリソース画像のパスです。詳細については、DemoのXmagicResParserクラスのparseBeauty()メソッドをご参照ください</p>
effkey	<p>入力必須です。Demo参照</p> <p>事例：美白BeautyConstant.BEAUTY_TOOTH_WHITEN</p>
effValue	<p>入力必須です。DemoのXmagicResParserクラスのparseBeauty()メソッドをご参照ください</p>

美ボディ

属性フィールド	説明
category	Category.BODY_BEAUTY
ID	null
resPath	null
effkey	入力必須です。Demo参照 事例：脚長BeautyConstant.BODY_LEG_STRETCH
effValue	入力必須です。DemoのXmagicResParserクラスのparseBeauty()メソッドをご参照ください

フィルター

属性フィールド	説明
category	Category.LUT
ID	画像名、入力必須 事例：dongjing_lf.png ID「なし」の場合は、XmagicProperty.ID_NONE
resPath	フィルター画像のパス。入力必須です。「なし」の場合はnullに設定されます
effkey	null
effValue	入力必須です。「なし」の場合はnullに設定されます

動的エフェクト

属性フィールド	説明
category	Category.MOTION
ID	リソースソースファイル名、入力必須 事例：video_lianliancaomei ID「なし」の場合は、XmagicProperty.ID_NONE
resPath	入力必須です。Demo参照
effkey	null
effValue	null

メイクアップ

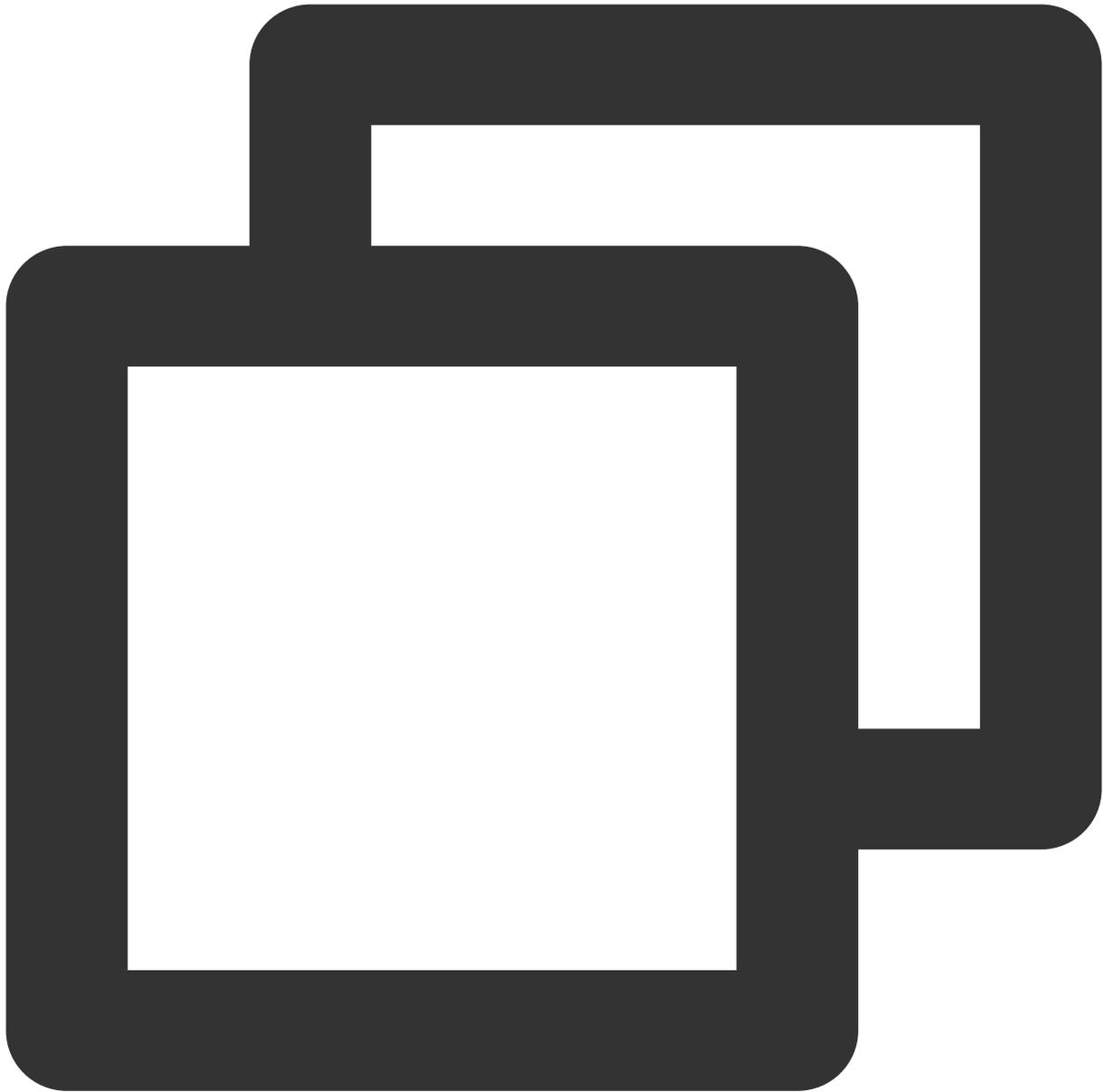
属性フィールド	説明
category	Category.MAKEUP
ID	リソースソースファイル名、入力必須 事例：video_xuejiezhuang ID「なし」の場合は、XmagicProperty.ID_NONE
resPath	入力必須です。Demo参照
effkey	入力必須です。値は、makeup.strength「なし」の場合はnullに設定されます
effValue	入力必須です。「なし」の場合はnullに設定されます

分割

属性フィールド	説明
category	Category.SEGMENTATION
ID	リソースソースファイル名、入力必須 事例：video_segmentation_blur_45 ID「なし」の場合は、XmagicProperty.ID_NONE カスタム分割ID値には、 <code>XmagicConstant.SegmentationId.CUSTOM_SEG_ID</code> を使用する必要があります
resPath	入力必須です。Demo参照
effkey	null（カスタム背景を除く）。カスタム背景の値は選択するリソースパスとします
effValue	null

updateProperties

ある項目の美顔数値または動的エフェクト、フィルターの一括変更は、任意のスレッドで呼び出せます。



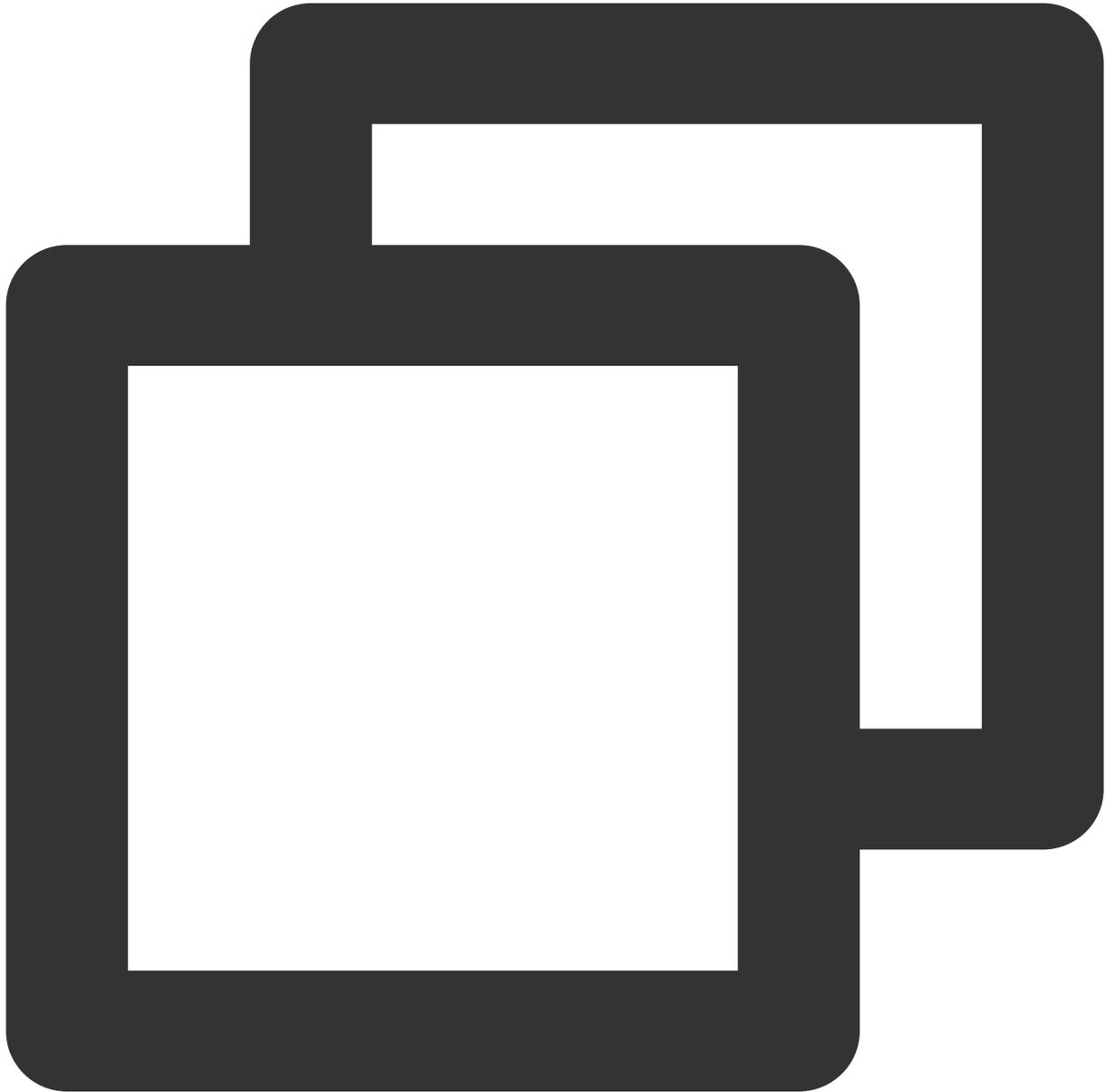
```
void updateProperties (List<XmagicProperty<?>> properties)
```

パラメータ

パラメータ	意味
(List<XmagicProperty<?>> properties)	詳細については、updatePropertyメソッドの説明をご参照ください。

setTipsListener

動的エフェクトプロンプトのコールバック関数を設定します。プロンプトをフロントエンドのページ上に表示するために用いられます。例えば素材の中には、ユーザーに、うなづく、手を伸ばす、指ハートなどを促すものがあります。



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

パラメータ

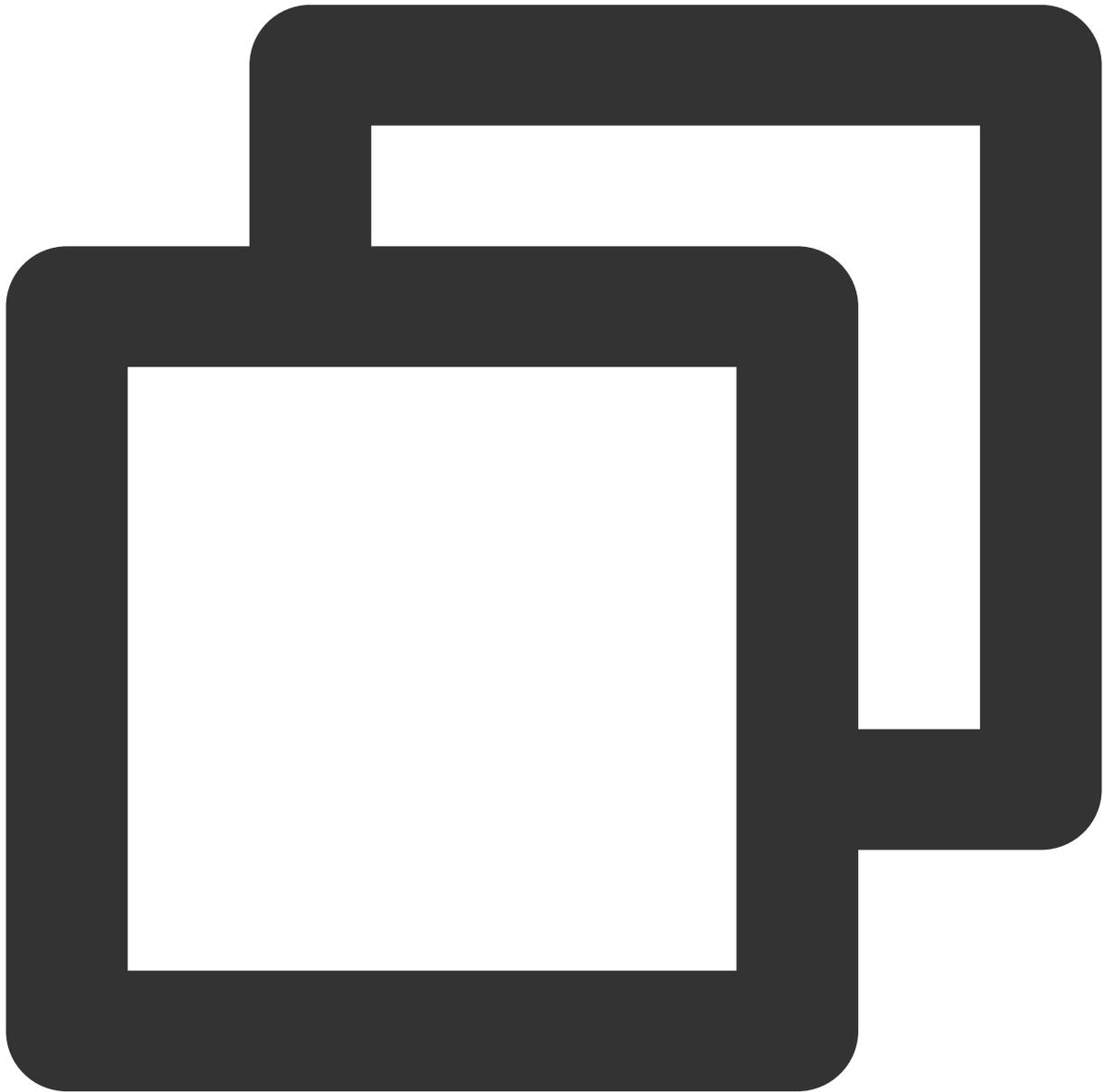
パラメータ	意味
-------	----

XmagicApi.XmagicTipsListener
effectTipsListener

コールバック関数の実装クラス、コールバックはメインスレッドにあるとは限りません。

setYTDataListener

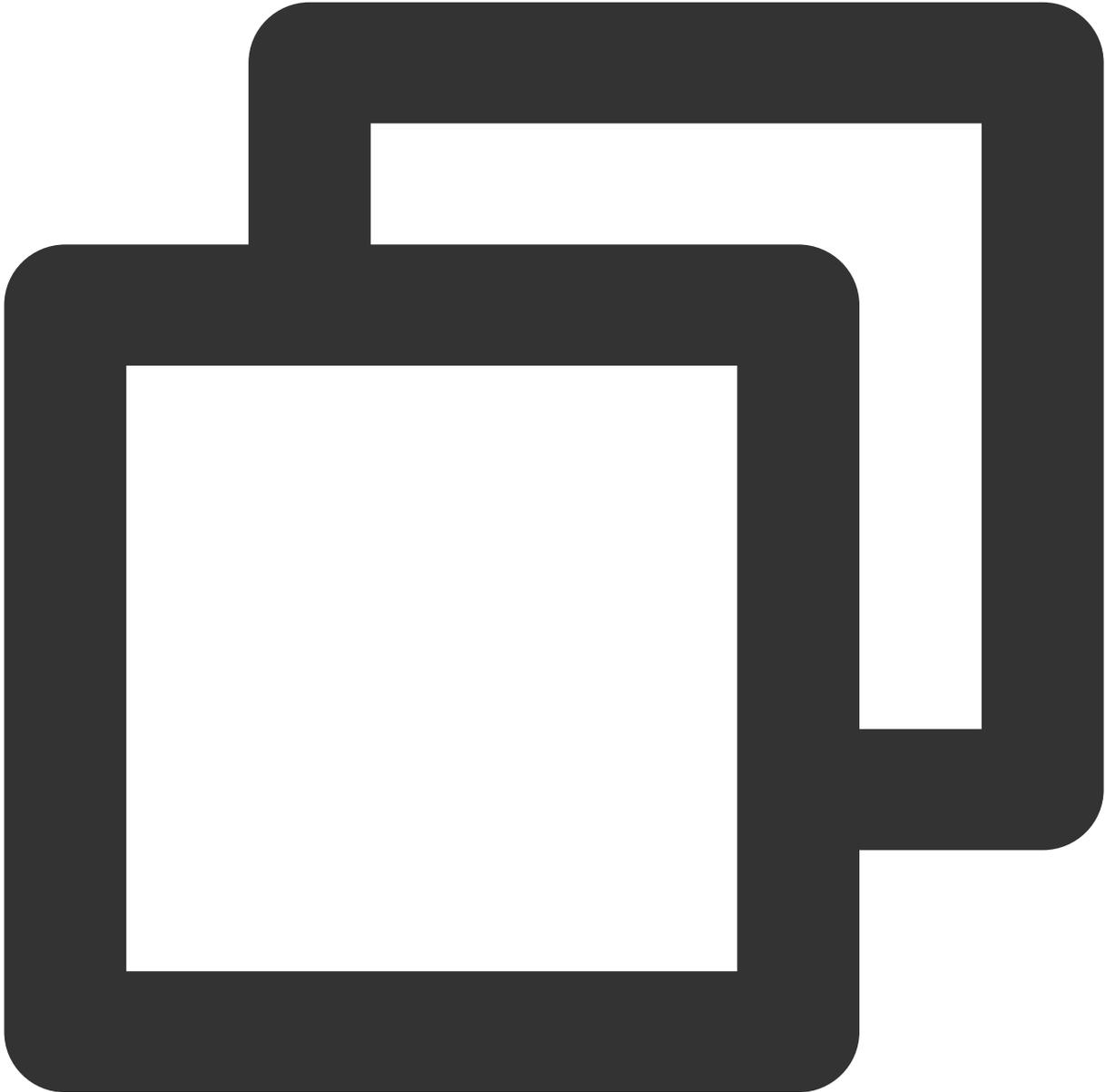
顔の特徴点位置情報などのデータのコールバックを設定します。



```
void setYTDataListener(XmagicApi.XmagicYTDataListener ytDataListener)  
顔の情報などのデータのコールバックを設定します
```

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

onYTDataUpdateはJSON string構造を返します。最大で5つの顔の情報を返します。



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

```

フィールドの意味

フィールド	タイプ	値域	説明
trace_id	int	[1,INF)	顔ID。連続ストリーム取得の過程で、IDが同一であれば同じ顔であると認識できます。
face_256_point	float	[0,screenWidth]または [0,screenHeight]	計512個、顔のキーポイントは256個、画面左上隅は(0,0)です。
face_256_visible	float	[0,1]	顔256キーポイントの視認性。
out_of_screen	bool	true/false	顔がフレームから外れているかどうか。
left_eye_high_vis_ratio	float	[0,1]	左目の高視認性特徴点の割合。
right_eye_high_vis_ratio	float	[0,1]	右目の高視認性特徴点の割合。
left_eyebrow_high_vis_ratio	float	[0,1]	左眉の高視認性特徴点の割合。
right_eyebrow_high_vis_ratio	float	[0,1]	右眉の高視認性特徴点の割合。
mouth_high_vis_ratio	float	[0,1]	口元の高視認性特徴点の割合。

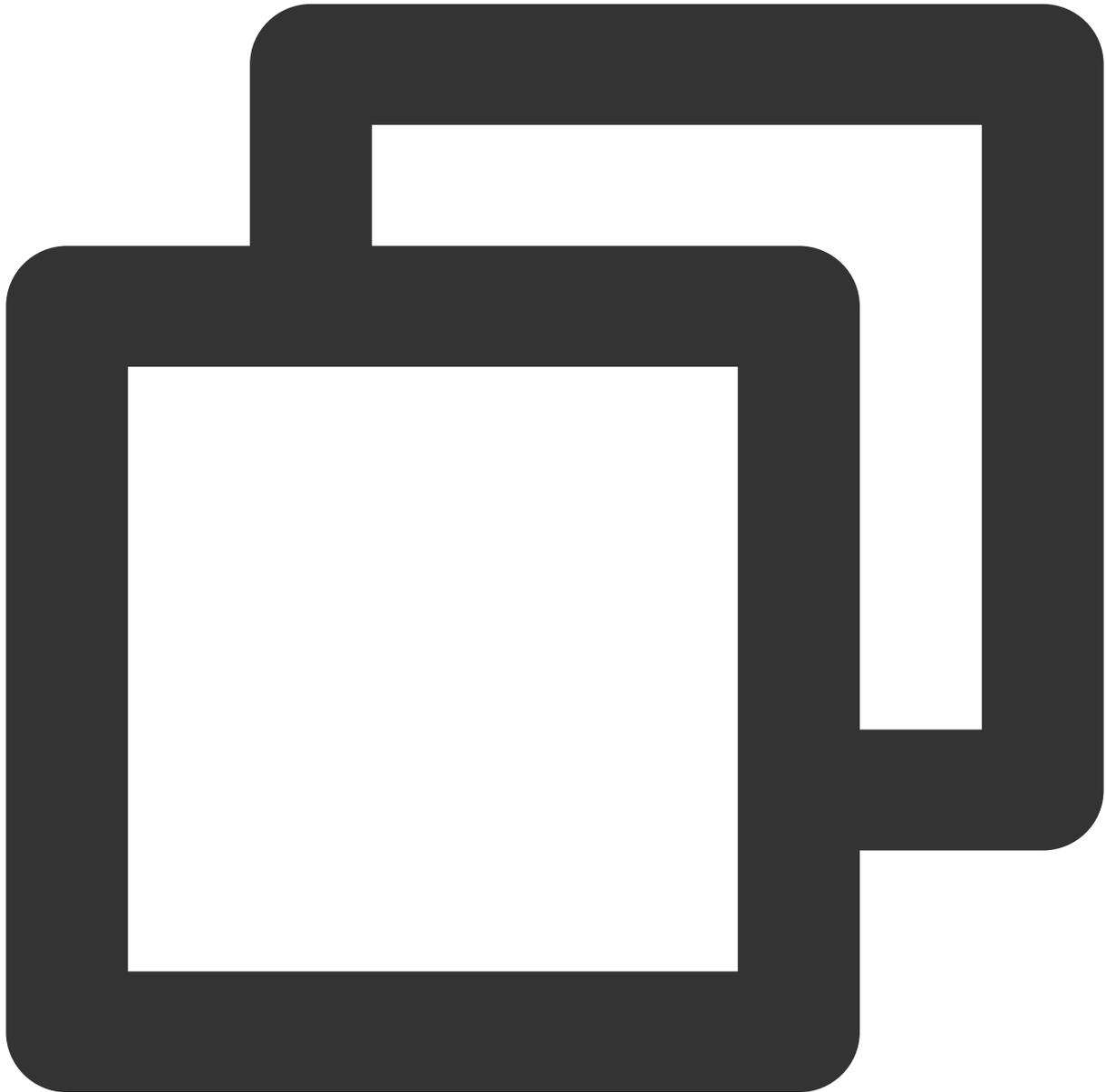
パラメータ

パラメータ	意味
-------	----

XmagicApi.XmagicYTDataListener ytDataListener	コールバック関数の実装クラス。
---	-----------------

setAIDataListener

顔、体、ジェスチャーを検出した際、これらの部位の特徴点位置情報をコールバックします

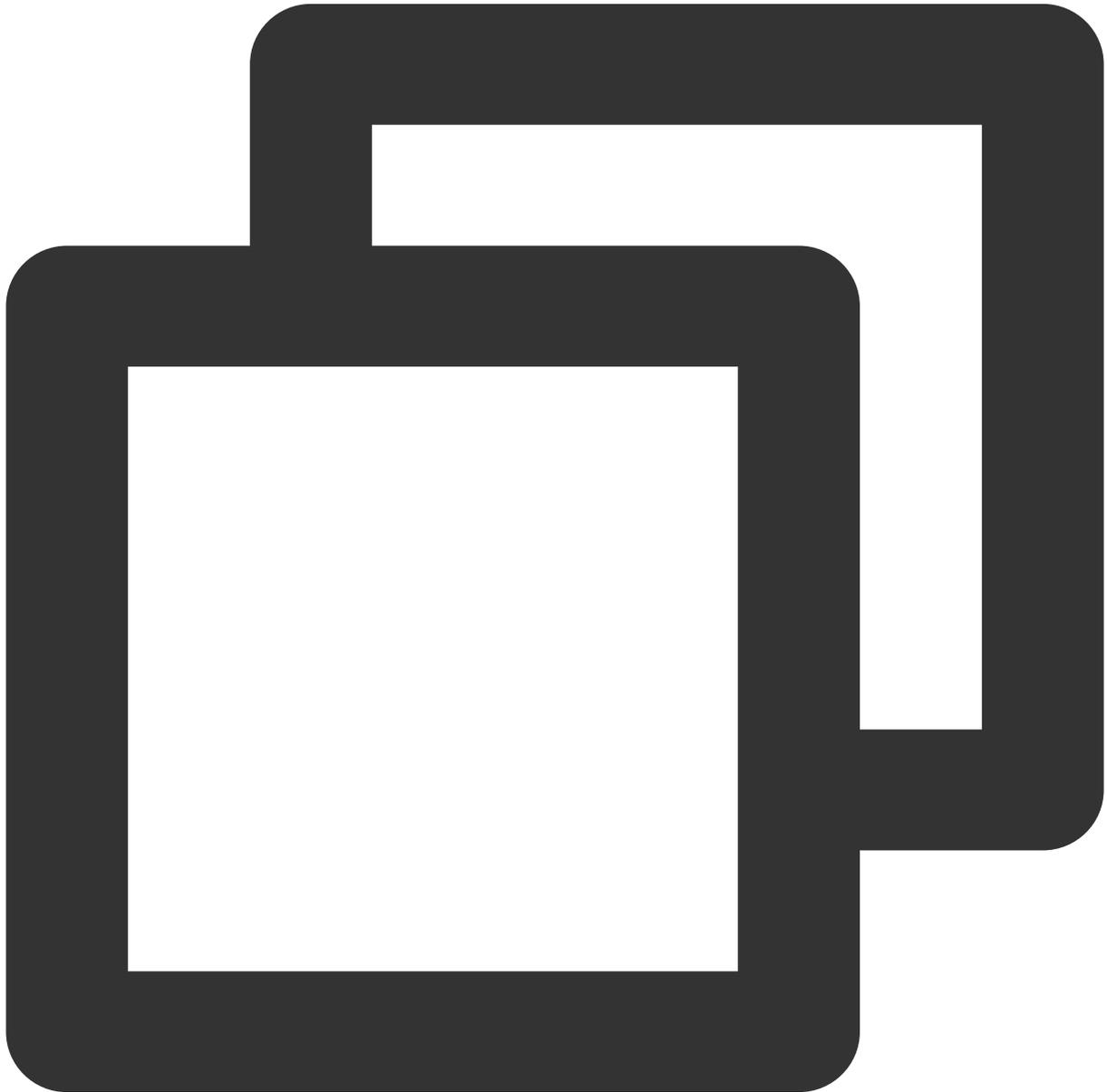


```
public interface OnAIDataListener {  
  
    void onFaceDataUpdated(List<FaceData> faceDataList);  
    void onHandDataUpdated(List<HandData> handDataList);  
    void onBodyDataUpdated(List<BodyData> bodyDataList);  
}
```

```
}
```

onPause

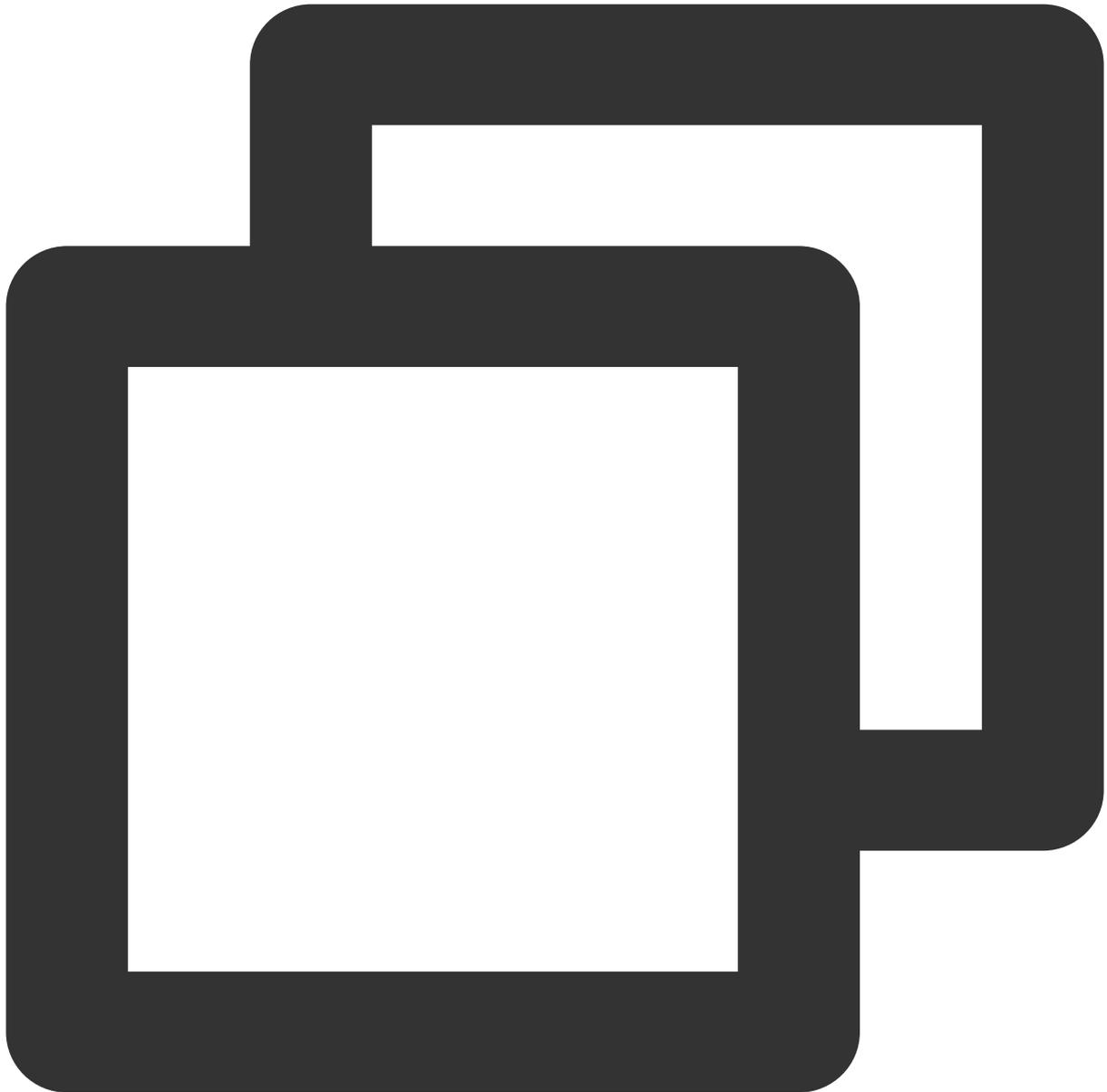
レンダリングを一時停止します。ActivityのonPauseライフサイクルにバインドできます。現時点で内部では `onPauseAudio` のみ呼び出します。



```
void onPause()
```

onResume

レンダリングを再開します。ActivityのonResumeライフサイクルにバインドできます。



```
void onResume ()
```

onDestroy

GLスレッドのリソースをクリアします。GLスレッド内で呼び出す必要があります。サンプルコードは次のとおりです。



```
//サンプルコードはMainActivity.javaをご参照ください
glSurfaceView.queueEvent(() -> {
    if (mXmagicApi != null) {
        mXmagicApi.onPause();
        mXmagicApi.onDestroy();
    }
});

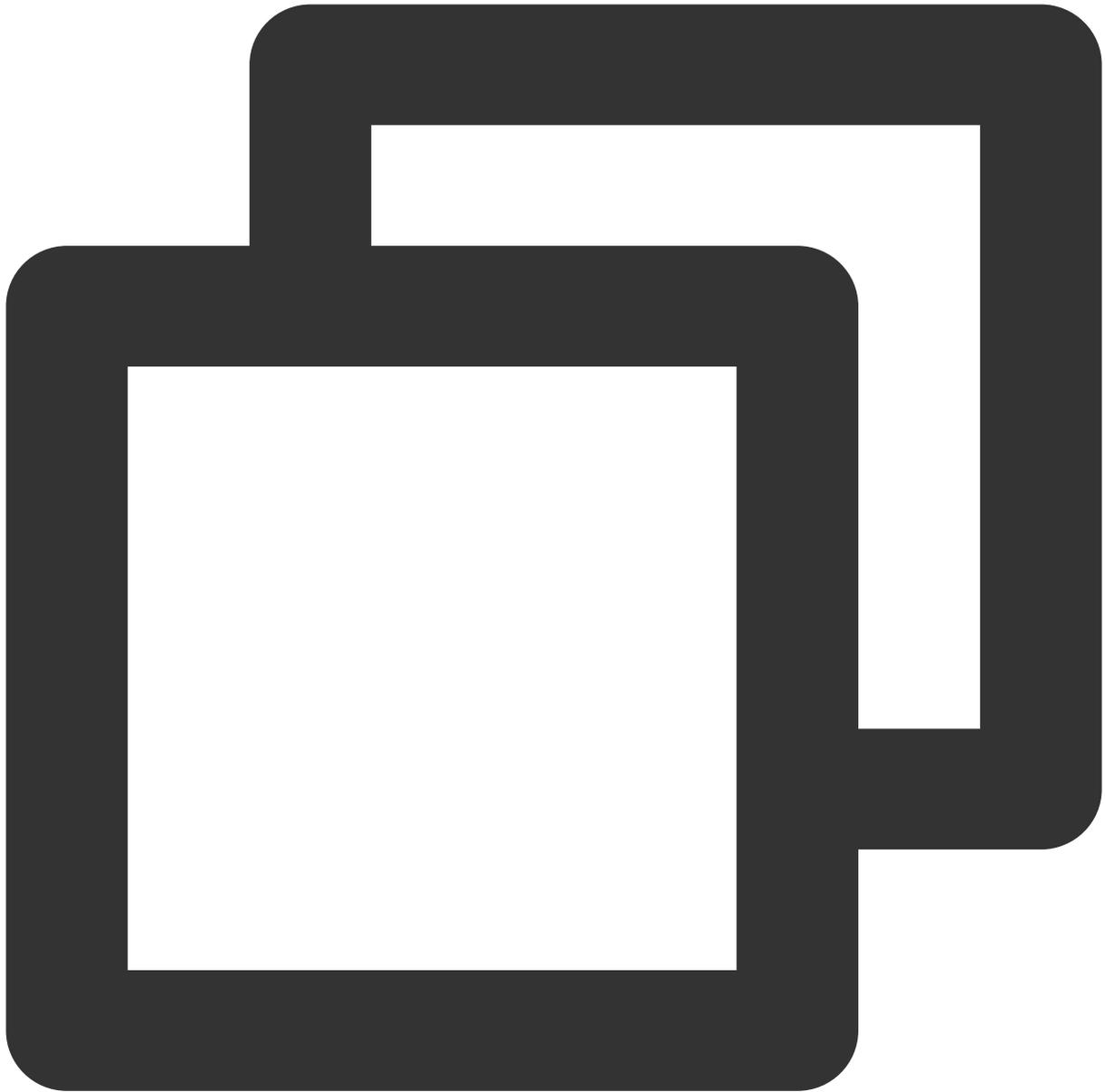
//サンプルコードはImageInputActivity.javをご参照ください
@Override
protected void onDestroy() {
```

```
if (mHandler != null) {
    mHandler.destroy() -> {
        if (mXmagicApi != null) {
            mXmagicApi.onPause();
            mXmagicApi.onDestroy();
        }
    });
    mHandler.waitDone();
}

XmagicPanelDataManager.getInstance().clearData();
super.onDestroy();
}
```

process

SDKレンダリングのデータを受信するメソッドです。カメラデータのコールバック関数内で使用できます。



```
//テクスチャのレンダリング  
int process(int srcTextureId, int srcTextureWidth, int srcTextureHeight)  
//bitmapのレンダリング  
Bitmap process(Bitmap bitmap, boolean needReset){
```

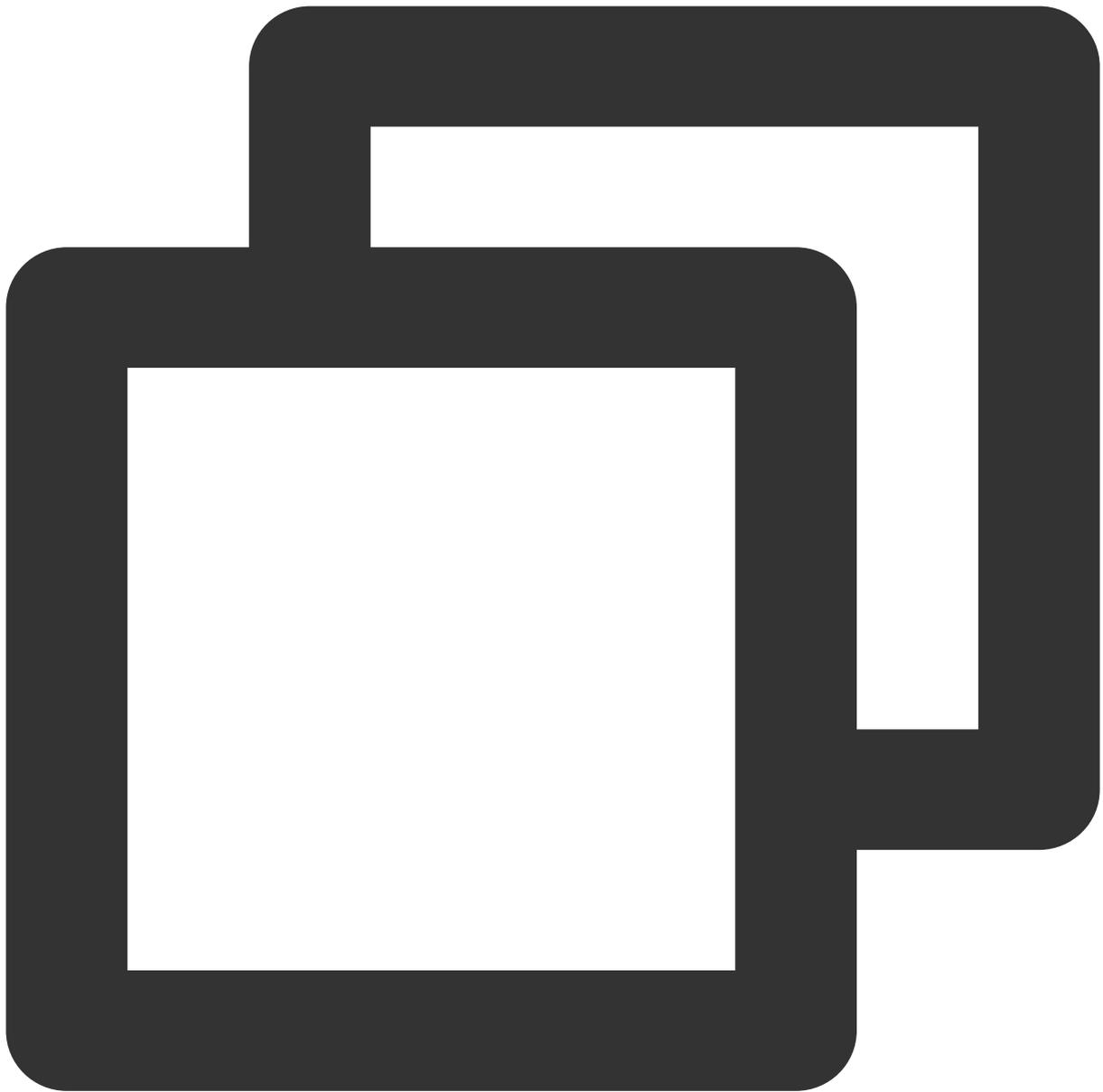
パラメータ

パラメータ	意味
int srcTextureId	レンダリングが必要なテクスチャ。

id int srcTextureWidth	レンダリングが必要なテクスチャの幅。
int srcTextureHeight	レンダリングが必要なテクスチャの高さ。
Bitmap bitmap	最大サイズは2160×4096までとすることをお勧めします。このサイズを超える画像は顔認識効果が低いか、または顔であると認識できない場合があります、またOOMの問題も起こりやすいため、大きな画像は縮小してから渡すことをお勧めします。
boolean needReset	画像を切り替えます。 分割を初めて使用します。 動的エフェクトを初めて使用します。 メイクアップを初めて使用します。 これらのシナリオでは、needResetはtrueに設定されています。

onPauseAudio

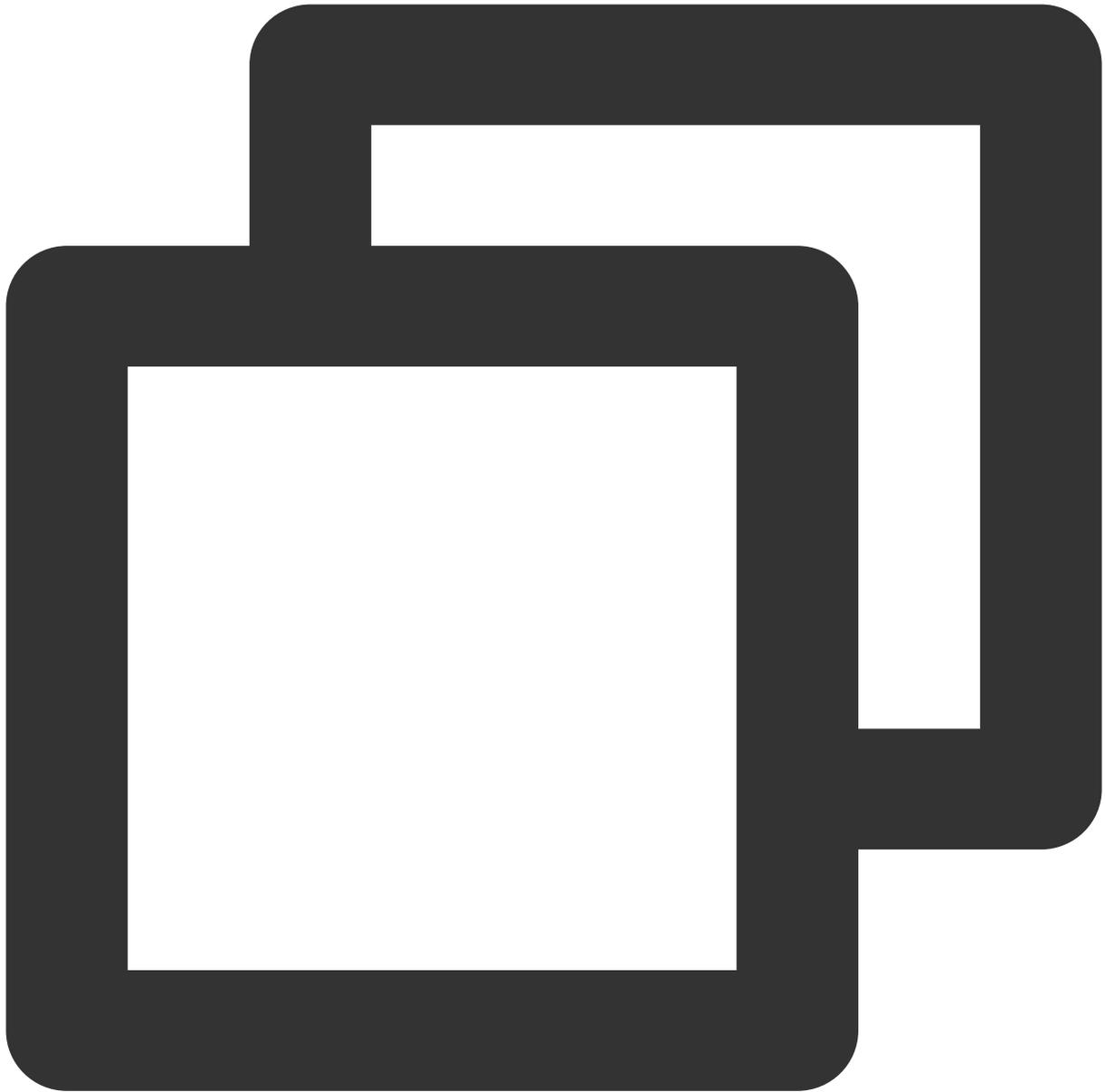
オーディオのみ停止する必要があり、GLスレッドをリリースする必要がない場合に、この関数を呼び出すことができます。



```
void onPauseAudio()
```

sensorChanged

現在のスマートフォンの回転の角度を判断し、それによってAIの顔認識の判断角度の根拠を調整するために用いられます。ジャイロスコープセンサーコールバック関数内で呼び出す必要があります。



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

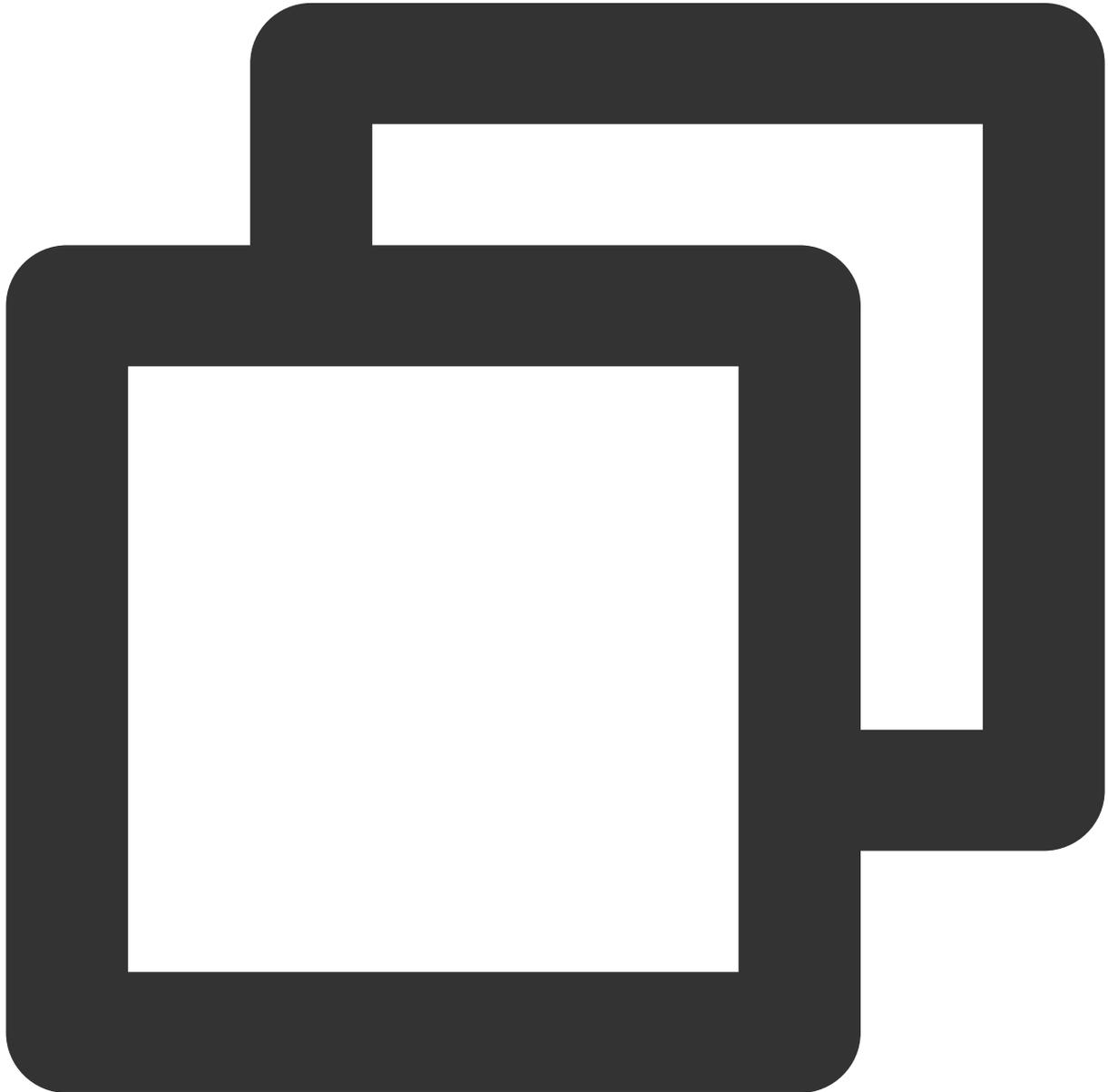
パラメータ

パラメータ	意味
SensorEvent event	ジャイロスコープセンサーのコールバック関数 <code>onSensorChanged</code> が返したイベントエンティティクラス。
Sensor	ジャイロスコープセンサーの事例。

accelerometer

isDeviceSupport

動的エフェクトリソースリストをSDKのチェック用に渡します。実行後、`XmagicProperty.isSupport`フィールドは、その素材が利用可能かどうかを表します。`XmagicProperty.isSupport` によってUIレイヤーのクリック制限を制御することや、直接リソースリストから削除することができます。



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

パラメータ

パラメータ	意味
List<XmagicProperty<?>> assetsList	検査が必要な動的エフェクト素材リスト。

getPropertyRequiredAbilities

動的エフェクトリソースリストを渡し、各リソースが使用できるSDKアトミック機能のリストを返します。メソッドのユースケースは次のとおりです。

いくつかの動的エフェクト素材を購入または制作し、このメソッドを呼び出すと、各素材が使用する必要があるアトミック機能のリストが返されます。例えば、素材1は機能A、B、Cを使用する必要があり、素材2は機能B、C、Dを使用する必要があるので、これらの機能リストをサーバー上で保持します。その後、ユーザーがサーバーから動的エフェクト素材をダウンロードしたい場合、先にgetDeviceAbilitiesメソッドによって、スマートフォンが備えるアトミック機能のリスト（このスマートフォンは機能A、B、Cを備えているが、機能Dは備えていないなど）を取得し、その機能リストをサーバーに渡すことで、サーバーがそのデバイスに機能Dが備わっていないと判断し、ユーザーに素材2を送信しないということが可能になります。

パラメータ

パラメータ	意味
List<XmagicProperty<?>> assets	アトミック機能が必要な動的エフェクトリソースリスト。

戻る

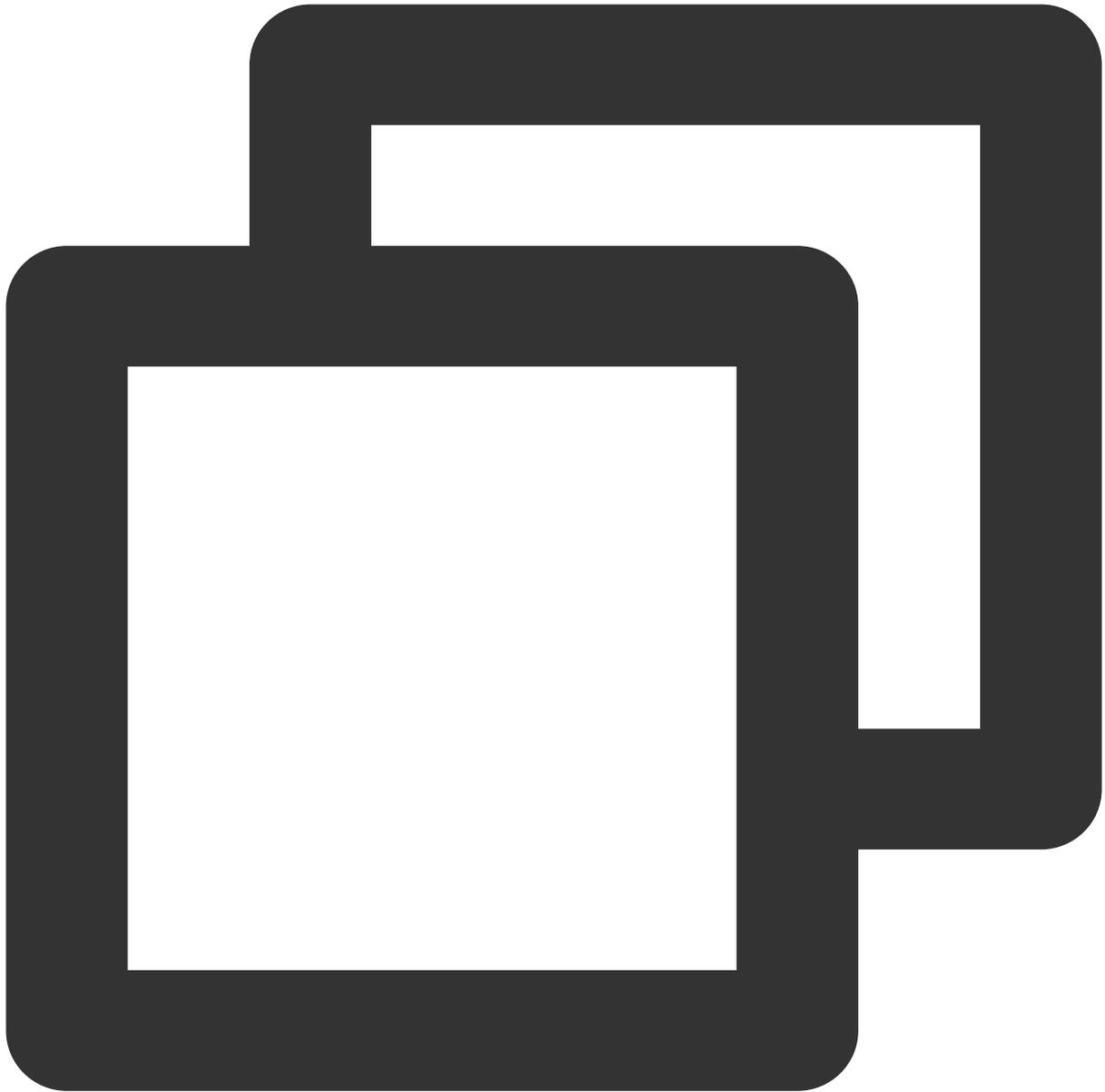
戻り値 Map<XmagicProperty<?>, ArrayList<String>> :

key: 動的エフェクトリソース素材のエンティティクラス。

value: 使用するアトミック機能のリスト。

getDeviceAbilities

現在のデバイスがサポートするアトミック機能のテーブルを返します。getPropertyRequiredAbilitiesメソッドと合わせて使用します。詳細についてはgetPropertyRequiredAbilitiesの説明をご参照ください。



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

戻る

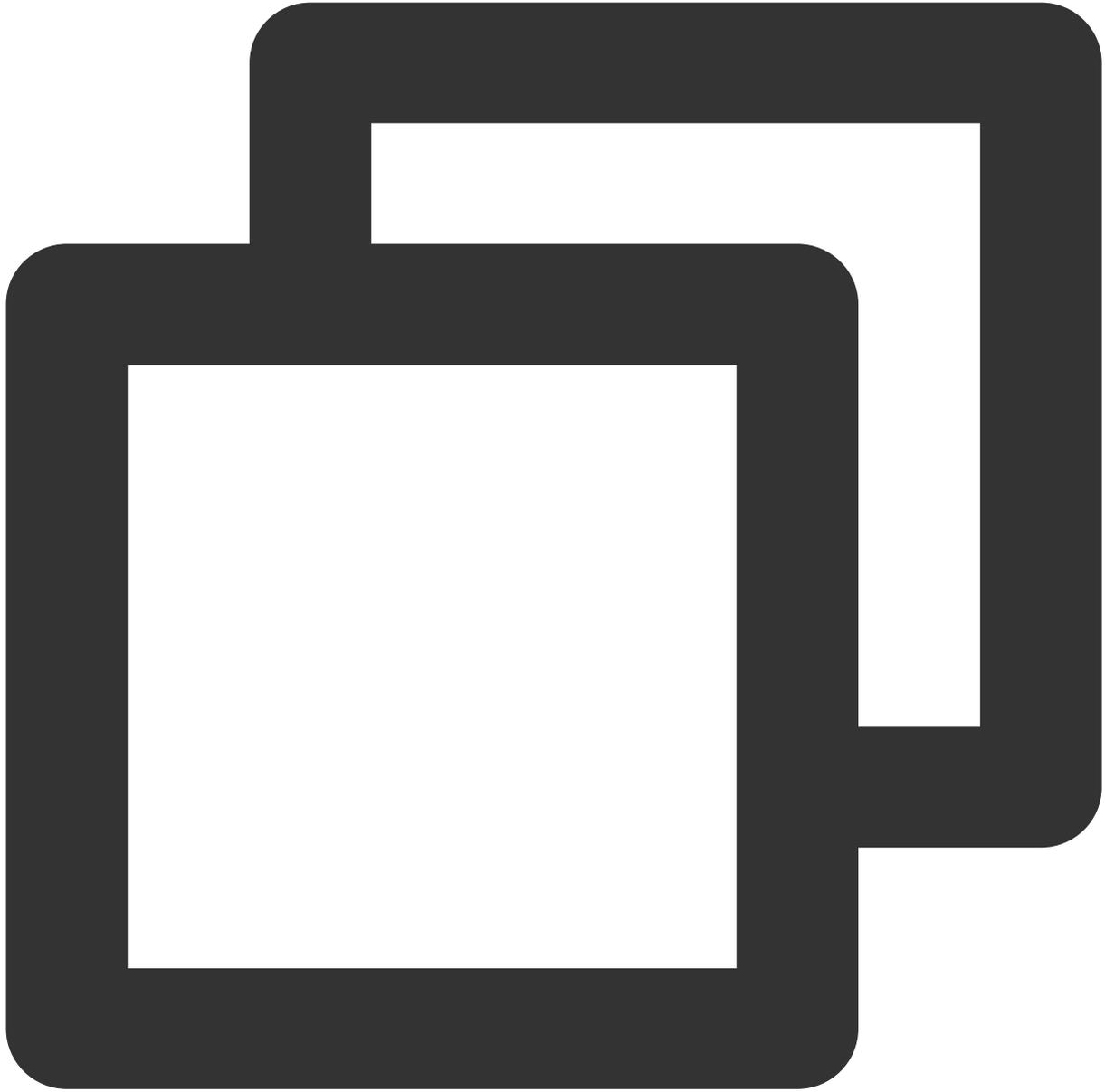
戻り値 `Map<String, Boolean>` :

key : アトミック機能名（素材の機能名に対応）。

value : 現在のデバイスがサポートしているかどうか。

isSupportBeauty

現在のモデルが美顔（OpenGL3.0）をサポートしているかどうかを判断します。



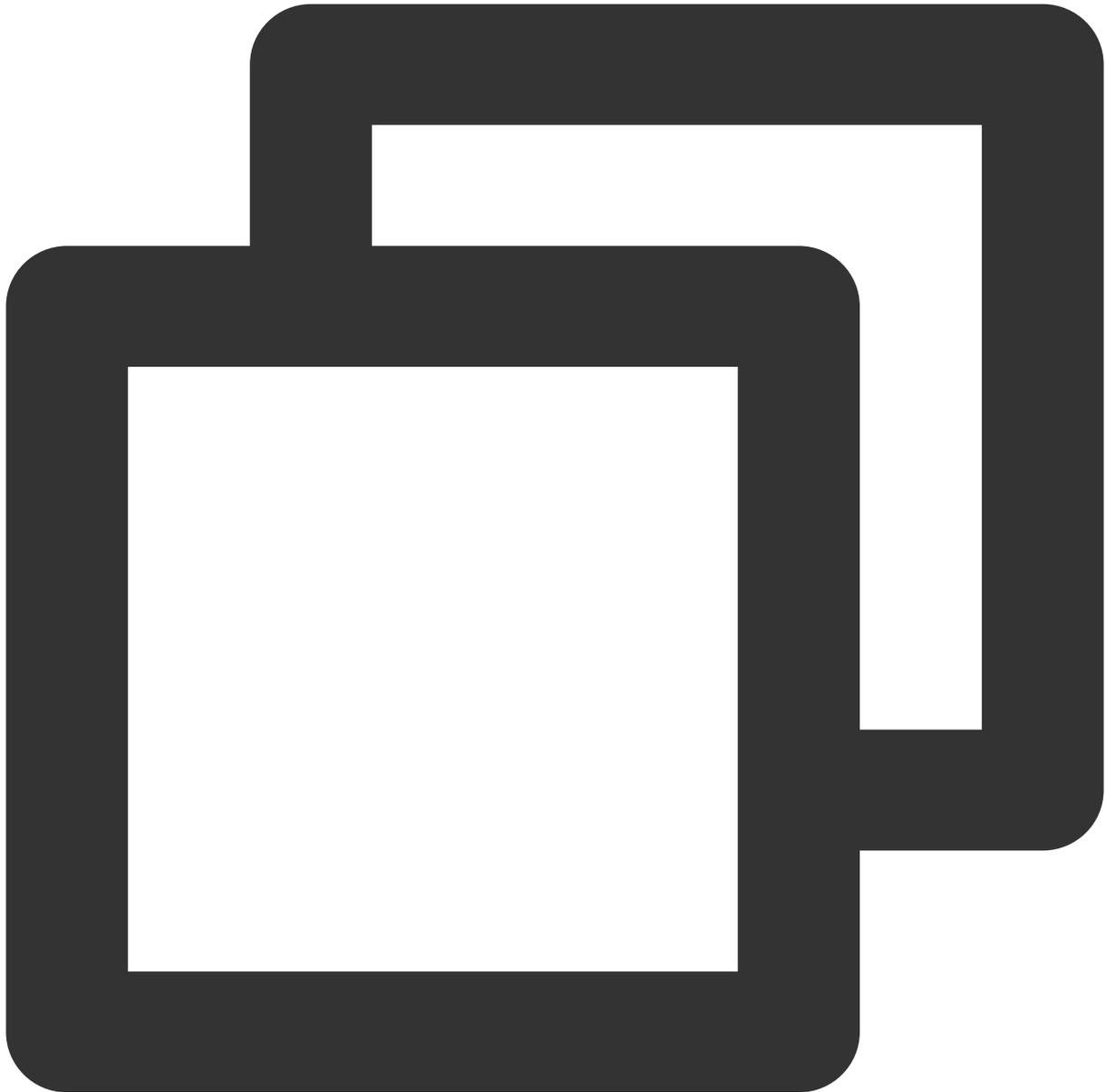
```
boolean isSupportBeauty()
```

戻る

戻り値boolean：美顔をサポートしているかどうか。

isBeautyAuthorized

現在のLicense権限がどの美顔または美ボディをサポートしているかを判断します。BEAUTYおよびBODY_BEAUTYタイプの美顔項目のチェックのみサポートします。チェックの結果は各美顔オブジェクトの `XmagicProperty.isAuth` フィールドに割り当てられます。isAuthフィールドがfalseの場合は、UI上でこれらの項目へのエントリーをブロックすることができます。



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

パラメータ

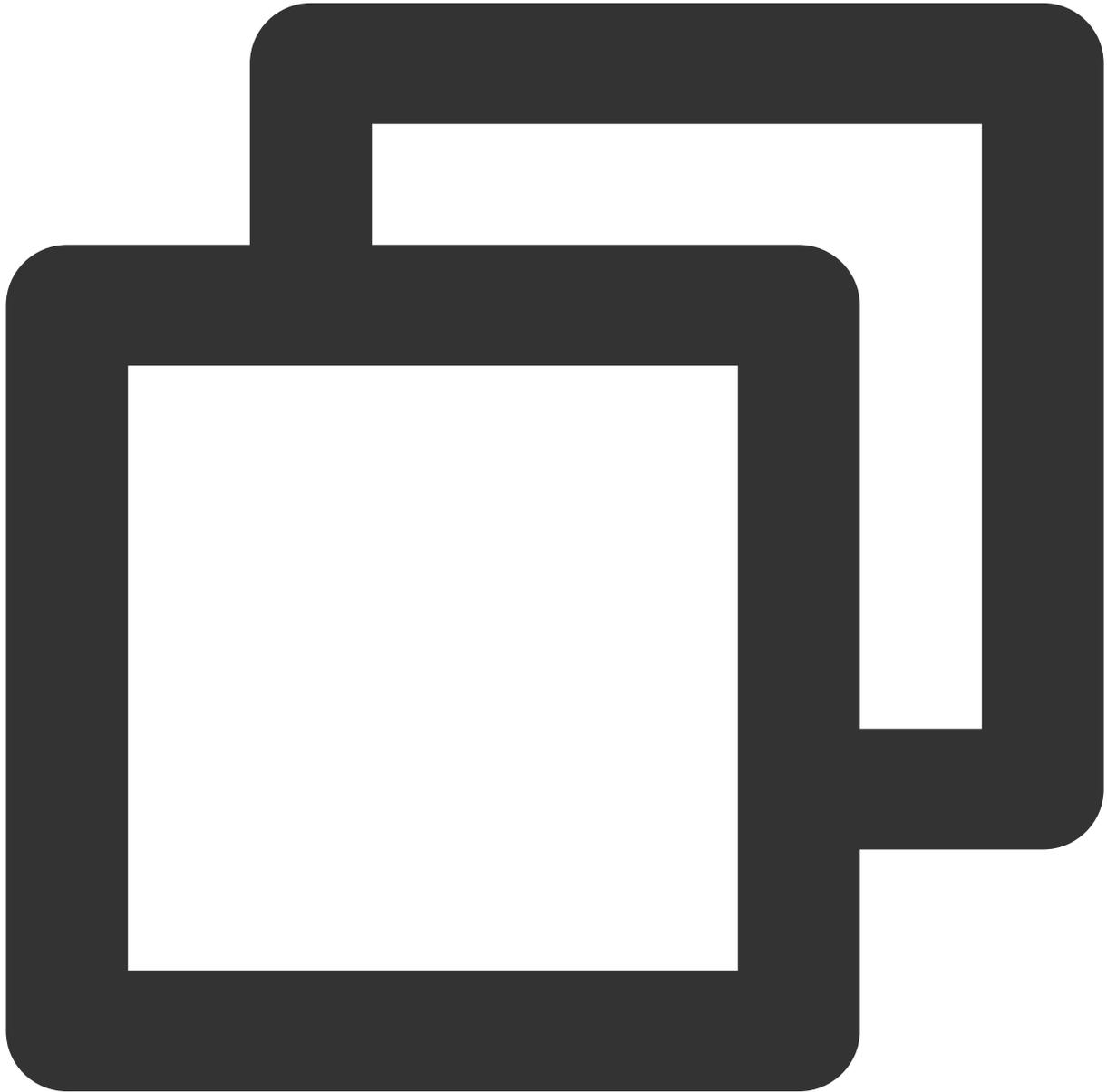
パラメータ	意味

```
List<XmagicProperty<?>> properties
```

検出が必要な美顔アイテム。

setXmagicStreamType

入力データのタイプを設定します。デフォルトではAndroid cameraデータストリームです (XmagicApi.PROCESS_TYPE_CAMERA_STREAM)。



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

パラメータ

パラメータ	意味
int type	データソースタイプ、以下2つのオプションがあります。 XmagicApi.PROCESS_TYPE_CAMERA_STREAM：カメラデータソース。 XmagicApi.PROCESS_TYPE_PICTURE_DATA：画像データソース。

静的関数

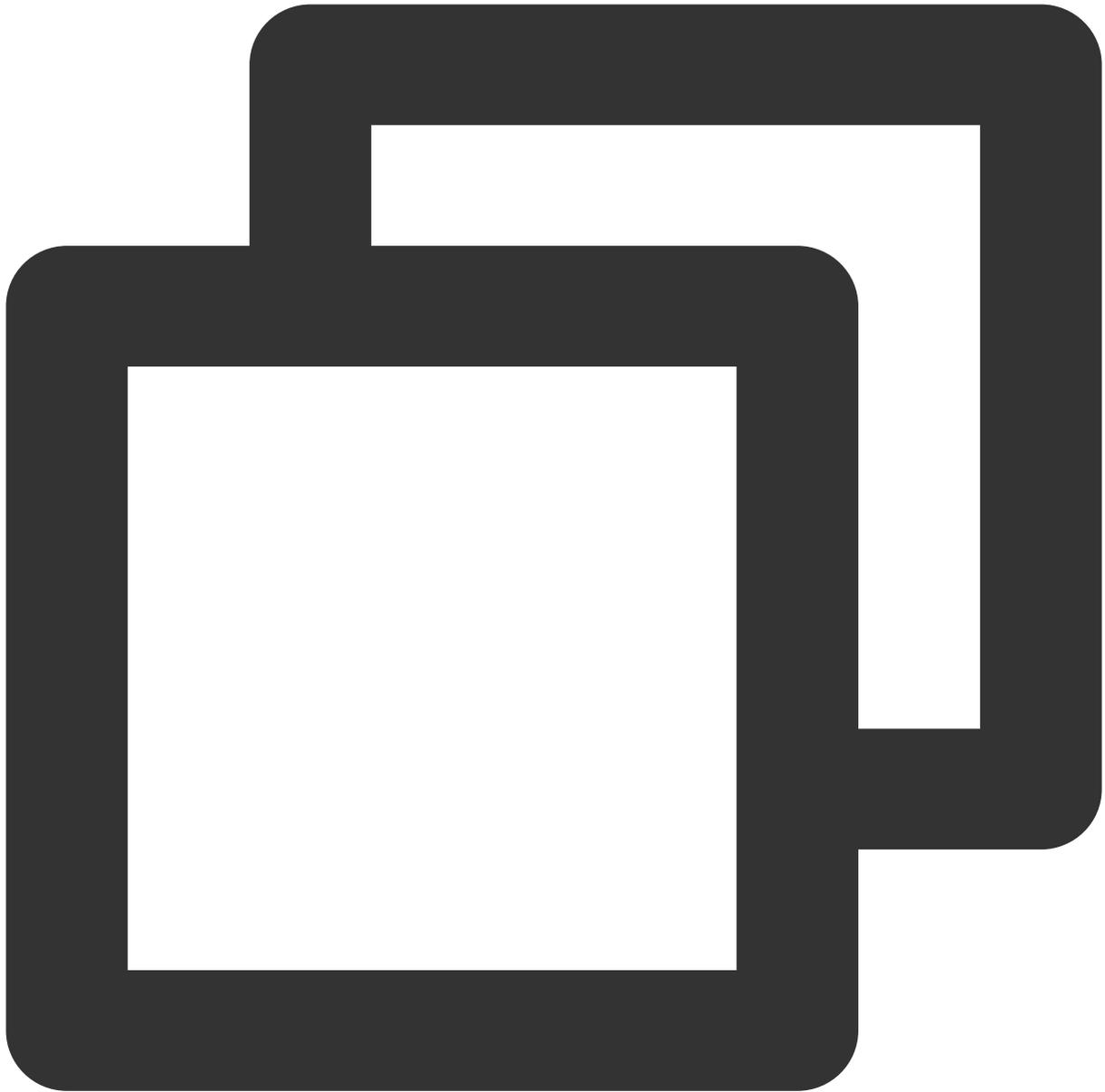
API	説明
setLibPathAndLoad	libPathを設定します。
addAiModeFilesFromAssets	アプリケーションassets下のLight3DPlugin、LightCore、LightHandPlugin、LightBodyPlugin、LightSegmentPluginフォルダの内容を、指定したディレクトリにコピーします
addAiModeFiles	クライアントがダウンロードしたAIモデルファイルを対応するフォルダにコピーします

setLibPathAndLoad

soのパスを設定し、ロードをトリガーします。soがassets内にある場合は、このメソッドを呼び出す必要はありません。soが動的ダウンロードしたものの場合は、認証および `new XmagicApi` の前に呼び出す必要があります。

nullを渡す：デフォルトのパスからsoをロードすることを表します。soがAPKパッケージ内にあることを確認してください。

null以外を渡す：例えば、`data/data/パッケージ名/files/xmagic_libs` などです。このディレクトリからsoをロードすることを表します。



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

パラメータ

パラメータ	意味
String path	soライブラリの格納パス。

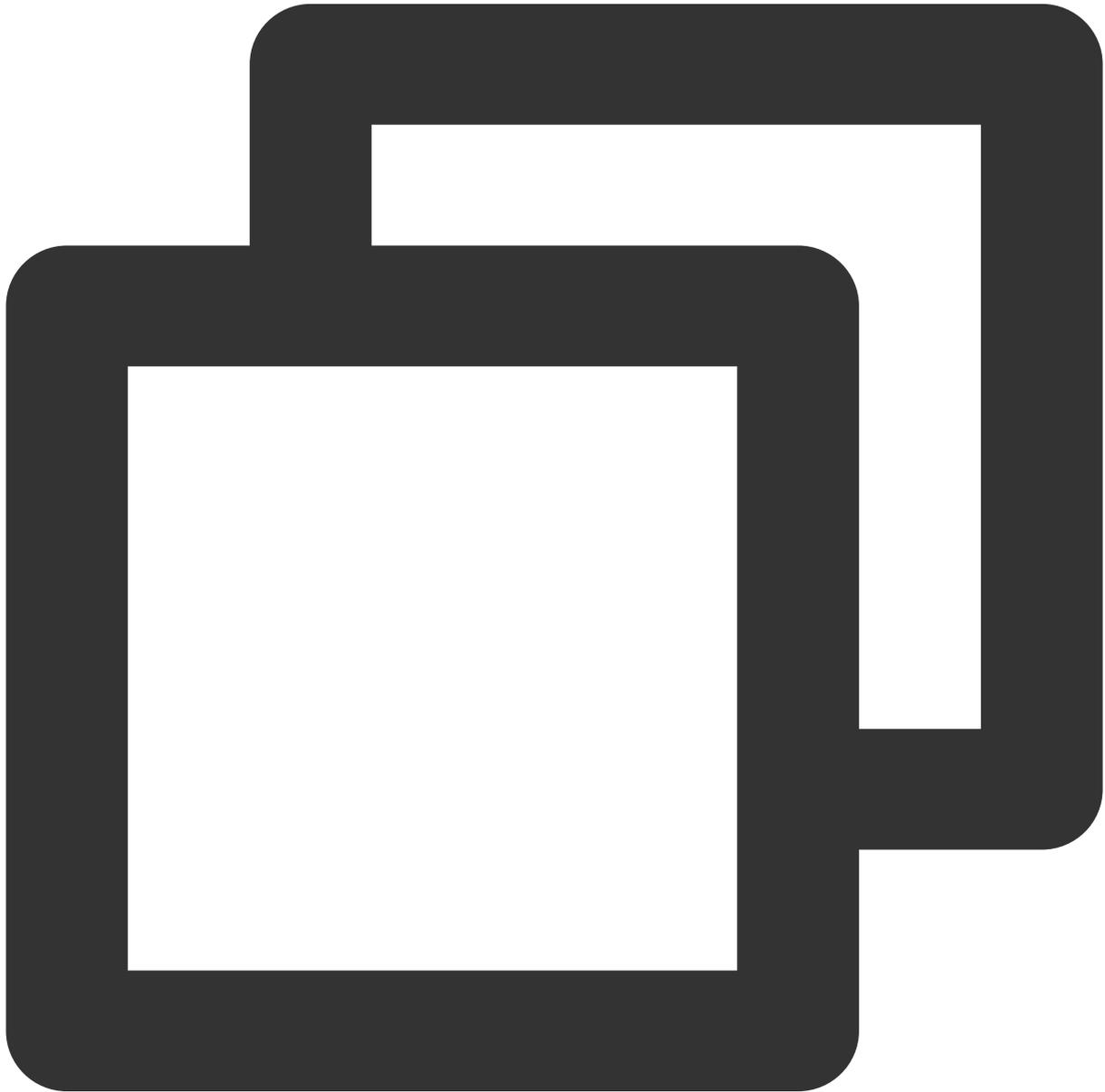
addAiModeFilesFromAssets

アプリケーションassets下のLight3DPlugin、LightCore、LightHandPlugin、LightBodyPlugin、LightSegmentPluginフォルダの内容を、指定したディレクトリにコピーします。

contextアプリケーションのコンテキスト。

resDir 美顔リソースを格納するためのルートディレクトリです。このディレクトリは、xmagicApiオブジェクトを作成するために渡されたパスと同一です。

戻り値：0: コピーが成功したことを示します -1:contextがnullであることを示します -2:IOエラーを示します。



```
static int addAiModeFilesFromAssets(Context context, String resDir)
```

パラメータ

パラメータ	意味
Context context	アプリケーションのコンテキスト。
String resDir	美顔リソースを格納するためのルートディレクトリです。 このディレクトリは、 <code>xmagicApi</code> オブジェクトを作成するために渡されたパスと同一です。

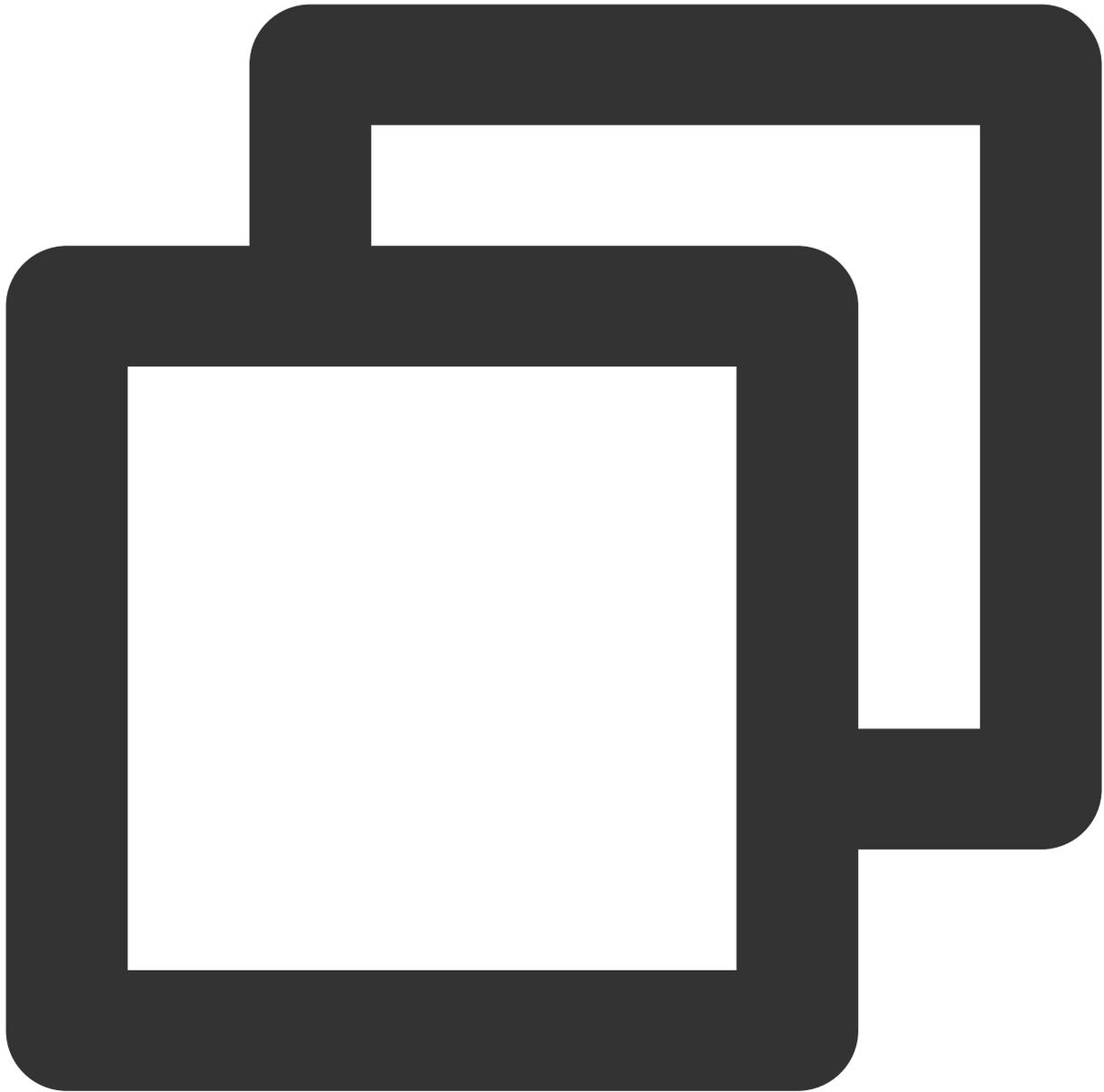
addAiModeFiles

お客様がダウンロードしたAIモデルファイルに対応するフォルダにコピーします。

`inputResDir` ダウンロードに成功したモデルファイルのフォルダです。

`resDir` 美顔リソースを格納するためのルートディレクトリです。このディレクトリは、`xmagicApi`オブジェクトを作成するために渡されたパスと同一です。

戻り値：0は成功したことを示します -1：inputResDir is not exists -2:IOエラーを示します。



```
static int addAiModeFiles(String inputResDir, String resDir)
```

パラメータ

パラメータ	意味
String inputResDir	ダウンロードに成功したモデルファイルのフォルダです。
String resDir	美顔リソースを格納するためのルートディレクトリです。 このディレクトリは、xmagicApiオブジェクトを作成する

ために渡されたパスと同一です。

Flutter

最終更新日：：2022-12-15 11:30:53

Tencent Effect SDK Flutter版のコアインターフェースクラス `TencentEffectApi` は、美顔数値の更新、動的エフェクト呼び出しなどの機能に用いられます。

Publicメンバー関数

API	説明
<code>initXmagic</code>	美顔データを初期化します。美顔の使用前に必ずこのメソッドを呼び出してください
<code>setLicense</code>	美顔権限の承認を行います
<code>setXmagicLogLevel</code>	SDKのログレベルを設定します。開発デバッグの際はLog.DEBUGに、正式リリースの際はLog.WARNにそれぞれ設定することをお勧めします。正式リリースの際にLog.DEBUGに設定していると、大量のログがパフォーマンスに影響する場合があります
<code>onResume</code>	レンダリングを再開します。ページが表示されている場合に呼び出します
<code>onPause</code>	レンダリングを一時停止します。ページが非表示の場合に呼び出します
<code>updateProperty</code>	美顔属性を更新します。任意のスレッドで呼び出せます
<code>setOnCreateXmagicApiErrorListener</code>	美顔オブジェクト作成時のコールバックインターフェースを設定します（エラー発生時にこのインターフェースを呼び出します）
<code>setTipsListener</code>	動的エフェクトプロンプトのコールバック関数を設定します。プロンプトをフロントエンドのページ上に表示するために用いられます
<code>setYTDataListener</code>	顔の特徴点位置情報などのデータのコールバックを設定します（S1-05およびS1-06パッケージのみコールバックあり）
<code>setAIDataListener</code>	顔、ジェスチャー、体の検出状態のコールバックを設定します
<code>isBeautyAuthorized</code>	現在のlic権限がどの美顔をサポートしているかを判断します。BEAUTYおよびBODY_BEAUTYタイプの美顔項目のチェックのみサポートします。チェックの結果は各美顔オブジェクトのXmagicProperty.isAuthフィールドに割り当てられます

API	説明
<code>isSupportBeauty</code>	現在のモデルが美顔機能をサポートしているかどうかを判断します (OpenGL3.0)
<code>getDeviceAbilities</code>	現在のデバイスでサポートされているアトミック機能のテーブルを返します
<code>isDeviceSupport</code>	動的エフェクトリソースリストをSDKのチェック用に渡します。実行後、 <code>XmagicProperty.isSupport</code> フィールドは、そのアトミック機能が利用可能かどうかを表します。 <code>XmagicProperty.isSupport</code> によってUIレイヤーのクリック制限を制御することや、リソースリストから直接削除することができます
<code>getPropertyRequiredAbilities</code>	動的エフェクトリソースリストを渡し、各リソースが使用できるSDKアトミック機能のリストを返します

メンバー関数の説明

initXmagic

初期化メソッドです。

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

パラメータ

パラメータ	意味
String xmagicResDir	リソースファイルを配置したディレクトリ
InitXmagicCallBack callBack	コールバックインターフェースの初期化

setLicense

認証データを設定し、美顔権限の承認を行います

```
///美顔認証処理の実行
void setLicense(String licenseKey, String licenseUrl, LicenseCheckListener checkListener);
```

```
/// 権限承認検証結果のコールバックメソッド
```

```
typedef LicenseCheckListener = void Function(int errorCode, String msg);
```

パラメータ

パラメータ	意味
String licenseKey	認証LicenseKey
String licenseUrl	認証LicenseUrl
LicenseCheckListener checkListener	権限承認結果のコールバックインターフェース

setXmagicLogLevel

SDKのログレベルを設定します

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

パラメータ

パラメータ	意味
int logLevel	LogLevelを使用して定義可能なタイプを設定します

onResume

美顔処理を再開します

```
void onResume();
```

onPause

美顔処理を一時停止します

```
void onPause();
```

updateProperty

ある項目の美顔数値または動的エフェクト、フィルターを設定します。任意のスレッドで呼び出せます。

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

パラメータ

パラメータ	意味
XmagicProperty xmagicProperty	美顔属性パッケージ化オブジェクト

setOnCreateXmagicApiErrorListener

美顔オブジェクト作成時のエラーコールバックインターフェースを設定します

```
void setOnCreateXmagicApiErrorListener(OnCreateXmagicApiErrorListener? errorListener);
///美顔インスタンス作成時のエラーコールバックメソッド
typedef OnCreateXmagicApiErrorListener = void Function(String errorMsg, int code);
```

パラメータ

パラメータ	意味
OnCreateXmagicApiErrorListener? errorListener	美顔オブジェクト作成時のエラーメッセージコールバックインターフェース

返されるエラーコードの意味対照表：

エラーコード	意味
-1	不明なエラー
-100	3Dエンジンリソースの初期化に失敗しました
-200	GAN素材をサポートしていません
-300	デバイスがこの素材コンポーネントをサポートしていません
-400	テンプレートのJSONの内容が空です
-500	SDKのバージョンが低すぎます

エラーコード	意味
-600	分割をサポートしていません
-700	OpenGLをサポートしていません
-800	スクリプトをサポートしていません
5000	分割背景画像の解像度が2160×3840を超えています
5001	分割背景画像に必要なメモリが不足しています
5002	分割背景ビデオの解析に失敗しました。
5003	分割背景ビデオが200秒を超えています
5004	分割背景ビデオのフォーマットをサポートしていません

setTipsListener

動的エフェクトプロンプトのコールバック関数を設定します。プロンプトをフロントエンドのページ上に表示するために用いられます。例えば素材の中には、ユーザーに、うなづく、手を伸ばす、指ハートなどを促すものがあります。

```
void setTipsListener(XmagicTipsListener? xmagicTipsListener);
abstract class XmagicTipsListener {
    /// tipsを表示します。 Show the tip.
    /// @param tips tips文字列。 Tip's content
    /// @param tipsIcon tipsのicon。 Tip's icon
    /// @param type tipsタイプ。 0は文字列とiconの両方を表示することを表し、1はpag素材がiconのみを表示することを表します。 tips category, 0 means that both strings and icons are displayed, 1 means that only the icon is displayed for the pag material
    /// @param duration tipsは時間 (ミリ秒) を表示します。 Tips display duration, milliseconds
    void tipsNeedShow(String tips, String tipsIcon, int type, int duration);
    /// *
    /// tipsを非表示にします。 Hide the tip.
    /// @param tips tips文字列。 Tip's content
    /// @param tipsIcon tipsのicon。 Tip's icon
    /// @param type tipsタイプ。 0は文字列とiconの両方を表示することを表し、1はpag素材がiconのみを表示することを表します。 tips category, 0 means that both strings and icons are displayed, 1 means that only the icon is displayed for the pag material
    void tipsNeedHide(String tips, String tipsIcon, int type);
}
```

パラメータ

パラメータ	意味
XmagicTipsListener xmagicTipsListener	コールバック関数実装クラス

setYTDataListener

顔の特徴点位置情報などのデータのコールバックを設定します。

```
///顔の特徴点位置情報などのデータコールバックを設定します (s1-05およびs1-06パッケージのみコールバックあり)
void setYTDataListener(XmagicYTDataListener? xmagicYTDataListener);
顔の情報などのデータのコールバックを設定します
abstract class XmagicYTDataListener {
//Youtu AIデータコールバック。
void onYTDataUpdate(String data);
}
```

onYTDataUpdateはJSON string構造を返します。最大で5つの顔の情報を返します。

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

フィールドの意味

フィールド	タイプ	値の範囲	説明
trace_id	int	[1,INF)	顔ID。連続ストリーム取得の過程で、IDが同一であれば同じ顔であると認識できます
face_256_point	float	[0,screenWidth]または [0,screenHeight]	計512個。顔の256個の重要特徴点であり、画面左上隅が(0,0)です
face_256_visible	float	[0,1]	顔の256個の重要特徴点の視認度
out_of_screen	bool	true/false	顔が枠外に出ていないか
left_eye_high_vis_ratio	float	[0,1]	左目の特徴点のうち高視認度のものが占める割合
right_eye_high_vis_ratio	float	[0,1]	右目の特徴点のうち高視認度のものが占める割合
left_eyebrow_high_vis_ratio	float	[0,1]	左眉の特徴点のうち高視認度のものが占める割合
right_eyebrow_high_vis_ratio	float	[0,1]	右眉の特徴点のうち高視認度のものが占める割合
mouth_high_vis_ratio	float	[0,1]	口の特徴点のうち高視認度のものが占める割合

パラメータ

パラメータ	意味
XmagicYTDataListener xmagicYTDataListener	コールバック関数実装クラス

setAIDataListener

顔、体、ジェスチャーを検出した際、これらの部位の特徴点位置情報をコールバックします

```
void setAIDataListener(XmagicAIDataListener? aiDataListener);

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

```
void onBodyDataUpdated(String bodyDataList);
}
```

isBeautyAuthorized

現在のLicense権限がどの美顔または美ボディをサポートしているかを判断します。BEAUTYおよびBODY_BEAUTYタイプの美顔項目のチェックのみサポートします。チェックの結果は各美顔オブジェクトの `XmagicProperty.isAuth` フィールドに割り当てられます。isAuthフィールドがfalseの場合は、UI上でこれらの項目へのエントリーをブロックすることができます。

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

パラメータ

パラメータ	意味
List<XmagicProperty> properties	チェックする必要がある美顔項目

isSupportBeauty

現在のモデルが美顔（OpenGL3.0）をサポートしているかどうかを判断します。

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

戻り値

戻り値bool：美顔をサポートしているかどうか。

getDeviceAbilities

現在のデバイスがサポートするアトミック機能のテーブルを返します。getPropertyRequiredAbilitiesメソッドと合わせて使用します。

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

戻り値

戻り値 `Map<String, bool>` :

- key：アトミック機能名（素材の機能名に対応）。
- value：現在のデバイスがサポートしているかどうか。

getPropertyRequiredAbilities

動的エフェクトリソースリストを渡し、各リソースが使用できるSDKアトミック機能のリストを返します。

メソッドのユースケースは次のとおりです。

いくつかの動的エフェクト素材を購入または制作し、このメソッドを呼び出すと、各素材が使用する必要があるアトミック機能のリストが返されます。例えば、素材1は機能A、B、Cを使用する必要があり、素材2は機能B、C、Dを使用する必要があるので、これらの機能リストをサーバー上で保持します。その後、ユーザーがサーバーから動的エフェクト素材をダウンロードしたい場合、先にgetDeviceAbilitiesメソッドによって、スマートフォンが備えるアトミック機能のリスト（このスマートフォンは機能A、B、Cを備えているが、機能Dは備えていないなど）を取得し、その機能リストをサーバーに渡すことで、サーバーがそのデバイスに機能Dが備わっていないと判断し、ユーザーに素材2を送信しないということが可能になります。

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

パラメータ

パラメータ	意味
List<XmagicProperty> assetsList	アトミック機能をチェックする必要がある動的エフェクトリソースリスト

戻り値

戻り値Map<XmagicProperty, List<String>?> :

- key：動的エフェクトリソース素材のエンティティクラス。
- value：使用するアトミック機能のリスト。

isDeviceSupport

動的エフェクトリソースリストをSDKのチェック用に渡します。実行後、XmagicProperty.isSupportフィールドは、その素材が利用可能かどうかを表します。XmagicProperty.isSupportによってUIレイヤーのクリック制限を制御することや、直接リソースリストから削除することができます。

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

パラメータ

パラメータ	意味
List<XmagicProperty> assetsList	チェックする必要がある動的エフェクト素材リスト