Tencent Effect SDK

기능 사례

제품 문서

©2013-2022 Tencent Cloud. All rights reserved.

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

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.

목록:

©2013-2022 Tencent Cloud. All rights reserved.

기능 사례
SDK 패키지 축소
iOS

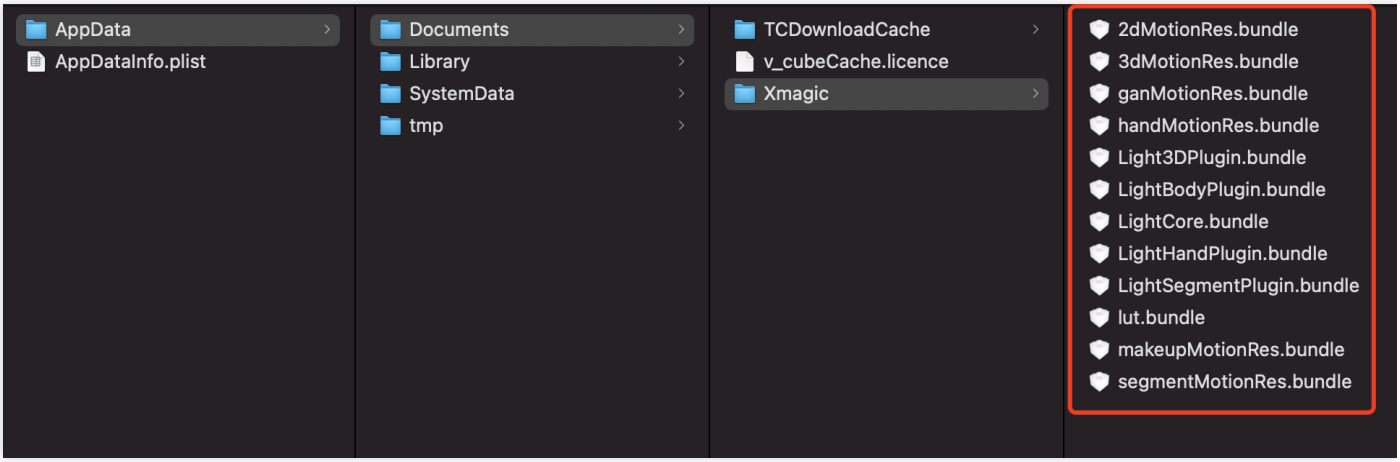
최종 업데이트 날짜: : 2022-08-12 15:14:58

리소스 동적 다운로드

패키지 크기를 줄이기 위해 SDK에 필요한 모델 리소스 및 애니메이션 리소스 MotionRes(일부 기본 버전 SDK 애니메이션 리소스 없음)를 인터넷으로 변경하여 다운로드할 수 있습니다. 다운로드가 완료 후 위 파일의 경로를 SDK로 설정합니다.

1. 뷰티 필터 리소스 ZIP 패키지를 클라우드에 업로드하여 다운로드 URL을 생성합니다. 예시: `https://서버 주소/LightCore.bundle.zip` .

2. 프로젝트에서 생성된 URL을 사용하여 파일을 다운로드하고 샌드박스에 압축을 해제합니다(예시: 샌드박스 경로 `Document/Xmagic`). 이 시점에서 Document/Xmagic 폴더에는 SDK에 필요한 리소스가 들어 있습니다.



3. SDK가 초기화되면 root_path 필드에 이전 단계의 샌드박스 경로를 입력합니다.

```
NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
@"root_path":_filePath ,//_filePath는 뷰티 필터 리소스가 로컬 Ducument/Xmagic에 다운로드된 후의 상위 디렉터리입니다.
@"tnn_"
@"beauty_config":beautyConfigJson
};
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Init beauty kit @"root_path":Ducument/Xmagic,
self.beautyKit = [[XMagic alloc] initWithRenderSize:_inputSize assetsDict:asset
sDict];
```

4. 뷰티 필터 패널에서 각 뷰티 필터 효과의 icon을 설정하고, 다운로드한 리소스 파일에서 해당 이미지를 가져옵니다.

```
NSMutableArray *arrayModels = [NSMutableArray array];
for (NSDictionary* dict in motionArray) {
BeautyCellModel* model = [BeautyCellModel beautyWithDict:dict];
// Load default mainbundle path of motionres
if ([model.title isEqualToString:NSLocalizedString(@"item_none_label",nil)]) {
model.icon = [NSString stringWithFormat:@"%@/%@.png", [[NSBundle mainBundle] bu
ndlePath], model.key];
[arrayModels addObject:model];
} else {
if(_useNetResource && _filePath != nil){ //네트워크 리소스 사용 시
NSString *DirPath = [_filePath stringByAppendingPathComponent:@"2dMotionRes.bun
dle/"]; //뷰티 필터 리소스의 절대 경로
model.icon = [NSString stringWithFormat:@"%@/%@/template.png", DirPath, model.k
ey];
}else{
model.icon = [NSString stringWithFormat:@"%@/%@/template.png", [[NSBundle mainB
undle] pathForResource:@"2dMotionRes" ofType:@"bundle"], model.key];
}
if ([fileManager fileExistsAtPath:model.icon]) {
[arrayModels addObject:model];
}
}
}
```

5. 뷰티 필터 효과의 매개변수 전달 설정(매개변수의 구체적인 설정은 API 문서 참고):

```
/// @brief 다양한 뷰티 필터 효과 설정
/// @param propertyType 효과 유형 문자열: beauty, lut, motion
/// @param propertyName 효과 이름
/// @param propertyValue 효과 값
/// @param extraInfo 예약 확장, 추가 별도 구성 Dict
/// @return 성공 반환 0, 실패 반환 기타
/// @note 설명
/**
| 효과 유형 | 효과 이름 | 효과 값 | 설명 | 비고 |
| :---- | :---- |:---- | :---- | :---- |
| beauty | 뷰티 필터 id 이름 | 뷰티 필터 효과 강도 값 |뷰티 필터 유형 설정 인터페이스 |
```

©2013-2022 Tencent Cloud. All rights reserved.

```
없음 |
| lut | 필터 경로+필터 이름 | 필터 강도 값 | 필터 유형 설정 인터페이스 | 없음 |
| motion | 애니메이션 경로 이름 | 애니메이션 경로 | 애니메이션 유형 설정 인터페이스| **참고**: 리소스에 zip이 있는 경우 입력한 애니메이션 경로가 쓰기 가능한 경로인지 확인하고, 그렇지 않으면 app 패키지를 수동으로 unzip해야 사용 가능 |
```

```
**/
```

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

예시

뷰티 필터 효과 설정

'뷰티 필터'와 '몸매 보정'의 특수 효과를 처리할 필요가 없으며 다운로드한 리소스 파일은 SDK 내에서 자동으로 사용됩니다. SDK 매개변수 전달 예시인 뷰티 필터의 미백 효과를 예로 들어 보겠습니다:

```
[self.beautyKitRef configPropertyWithType:@"beauty" withName:@"beauty.whiten" withData:@"30" withExtraInfo:nil];
```

이 때 SDK로 전달되는 각 매개변수의 값은 다음과 같습니다:

필드	값
propertyType	beauty
propertyName	beauty.whiten
propertyValue	30
extraInfo	nil

필터 효과 설정

©2013-2022 Tencent Cloud. All rights reserved.

key를 처리해야 하며 내장된 로컬 뷰티 필터 리소스 사용 또는 네트워크에서 로컬로 다운로드한 뷰티 필터 리소스 사용:

```
NSString *key = [_model.lutIDs[index] path];
if (key != nil) {
key = [@"lut.bundle/" stringByAppendingPathComponent:key];//필터 효과 이미지의 상대 경로
}
if(_useNetResource && _filePath != nil){ //다운로드한 뷰티 필터 리소스를 사용하는 경우
key = [_filePath stringByAppendingPathComponent:key];//효과 이미지의 절대 경로 생성
}
[self.beautyKitRef configPropertyWithType:@"lut" withName:key withData:[NSString stringWithFormat:@"%f",value] withExtraInfo:nil];
```

필터에 화이트 효과 설정

로컬 리소스 및 네트워크 리소스를 사용한 매개변수 전달 예시:

필드	로컬 리소스를 사용할 때 전달되는 매개변수	네트워크 리소스를 사용할 때 전달되는 매개변수
propertyType	lut	lut
propertyName	lut.bundle/n_baixi.png	`/var/mobile/Containers/Data/Application/25C7D01A-73F6-4F1B-AEB6-5EE03A221D18/Documents/Xmagic/lut.bundle/n_baixi.pr`
propertyValue	60.000000	60.000000
extraInfo	null	null

애니메이션 효과, 메이크업, 분할 효과 설정

propertyValue 필드를 처리해야 하며 내장된 로컬 뷰티 필터 리소스 사용 또는 네트워크에서 로컬로 다운로드한 뷰티 필터 리소스 사용:

```
NSString *key = [_model.motionIDs[index] key];
NSString *path = [_model.motionIDs[index] path];
NSString *motionRootPath = path==nil?[[NSBundle mainBundle] pathForResource:@"MotionRes" ofType:@"bundle"]:path;
if(_useNetResource && _filePath != nil){//다운로드한 뷰티 필터 리소스를 사용하는 경우
motionRootPath = [_filePath stringByAppendingPathComponent:@"2dMotionRes.bundle"];//2dMotionRes의 절대 경로 생성
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
[self.beautyKitRef configPropertyWithType:@"motion" withName:key withData:motionR
ootPath withExtraInfo:nil];
```

2D 애니메이션 설정-귀여운 낙서 효과

로컬 리소스 및 네트워크 리소스를 사용한 매개변수 전달 예시:

필드	로컬 리소스를 사용할 때 전달되는 매개변수	네트워크 리소스를
propertyType	motion	motion
propertyName	video_keaituya	video_keaituya
propertyValue	`/private/var/containers/Bundle/Application/FD2D7912-E58E-4584-B7E4-8715B8D2338F/BeautyDemo.app/2dMotionRes.bundle`	`/var/mobile/` `73F6-4F1B-AEB` `5EE03A221D18/`
extraInfo	nil	nil

©2013-2022 Tencent Cloud. All rights reserved.

Android

최종 업데이트 날짜: : 2022-08-12 15:16:36

패키지의 크기를 줄이려면 온라인에서 SDK에 필요한 assets 리소스, so 라이브러리 그리고 애니메이션 효과 리소스 MotionRes(일부 기본 SDK에서는 사용할 수 없음)를 다운로드할 수 있습니다. 다운로드 성공 후 SDK에서 위 파일의 경로를 설정합니다.

Demo의 다운로드 로직을 재사용하는 것이 좋습니다. 기존 다운로드 서비스를 사용할 수도 있습니다.

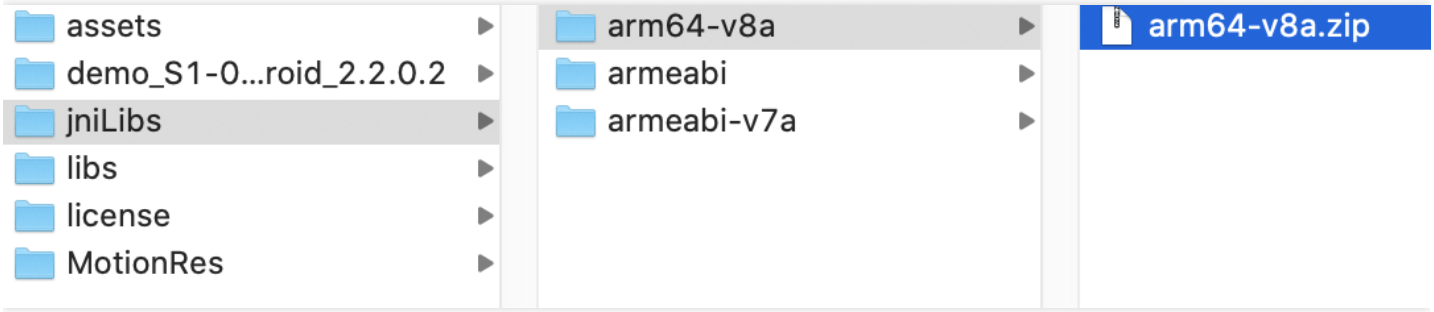
Demo 다운로드 로직을 재사용하는 경우 체크포인트 재시작 기능은 Demo에서 기본적으로 활성화되어 있어 다운로드가 중단된 경우 나중에 다시 시작할 수 있습니다. 이 기능을 사용하려면 다운로드 서버가 체크포인트 재시작 기능을 지원하는지 확인하십시오.

확인 방법

```
Web 서버가 Range 요청을 지원하는지 확인하십시오. Range 요청이 지원되는 경우 서버는 체크포인트 재시작을 지원합니다. 명령줄에서 다음 curl 명령을 실행합니다.
curl -i --range 0-9 https://귀하의 서버 주소/다운로드할 파일 이름
예시:
curl -i --range 0-9 https://mediacloud-76607.gzc.vod.tencent-cloud.com/TencentEffect/Android/2.4.1.119/xmagic_S1-04_android_2.4.1.119.zip
반환된 콘텐츠에 Content-Range 필드가 포함된 경우 서버는 체크포인트 재시작을 지원합니다.
```

so 라이브러리 동적 다운로드

so 라이브러리 패키지는 `jniLibs/arm64-v8a` 및 `jniLibs/armeabi-v7a` 에 있습니다.



- Demo에서 다운로드 서비스를 재사용하려면
- 자체 다운로드 서비스를 이용하려면

©2013-2022 Tencent Cloud. All rights reserved.

1. 두 ZIP 패키지의 MD5 값을 계산합니다. Mac에서 그렇게 하려면 터미널에서 `md5 파일 경로/파일 이름'을 직접 실행하거나 해당 도구를 사용하십시오.
2. 패키지를 서버에 업로드하고 다운로드 URL을 가져옵니다.
3. Demo 프로젝트의 ResDownloadConfig에서 다음 상수 값을 업데이트합니다.

```
public class ResDownloadConfig {
    //Directory structure
    //
    //---mylibs.zip(You can give it a custom name)
    //------liblibpag.so
    //------liblight-sdk.so
    //------libv8jni.so
    //
    //   You, A minute ago • Uncommitted changes
    public static final String DOWNLOAD_URL_LIBS_V8A = "https:/
    public static final String DOWNLOAD_URL_LIBS_V7A = "https:/
    //MD5 checksum of the ZIP file
    public static final String DOWNLOAD_MD5_LIBS_V8A = "libs-v8
    public static final String DOWNLOAD_MD5_LIBS_V7A = "libs-v7
```

4. `ResDownloadUtil.checkOrDownloadFiles` 를 호출하여 다운로드를 시작합니다.

주의 :

- SDK 버전이 업데이트되면 해당 so 라이브러리도 변경될 수 있으며, so 라이브러리를 다시 다운로드해야 합니다. Demo의 방법을 참고하여 MD5 값을 검증하는 것을 권장합니다.
- so 라이브러리를 직접 다운로드하든 Demo에서 다운로드 서비스를 재사용하든 상관없이 SDK의 auth API를 호출하기 전에 다운로드되었는지 확인하십시오. ResDownloadUtil은 다음과 같은 확인 방법을 제공합니다. 다운로드한 경우 아래와 같이 SDK에서 경로를 설정합니다.

```
String validLibsDirectory = ResDownloadUtil.getValidLibsDirectory(LaunchActivity.
this,
isCpuV8a() ? ResDownloadConfig.DOWNLOAD_MD5_LIBS_V8A : ResDownloadConfig.DOWNLOAD
_MD5_LIBS_V7A);
if (validLibsDirectory == null) {
Toast.makeText(LaunchActivity.this,"libs는 다운로드되지 않습니다. 먼저 다운로드하십시오"
,Toast.LENGTH_LONG).show();
return;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}
XmagicApi.setLibPathAndLoad(validLibsDirectory);
auth();
```

assets 리소스 동적 다운로드

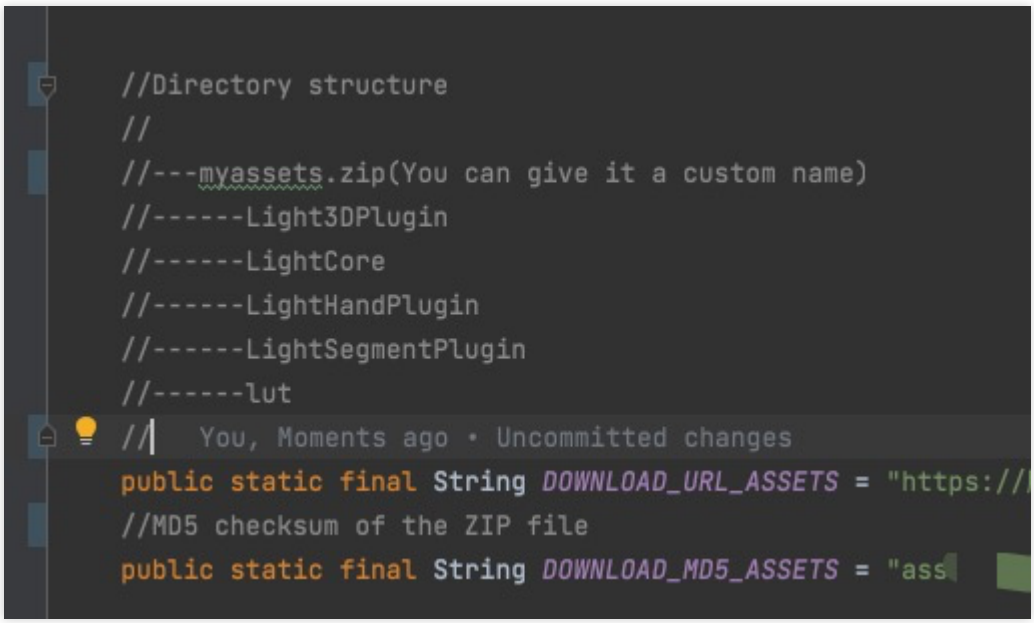
다음과 같이 assets 리소스를 동적으로 다운로드할 수 있습니다.

1. 로컬 프로젝트의 assets에서 다음을 구성합니다.
   - **2.4.0 이상**: 로컬 assets 디렉터리에 파일을 보관할 필요가 없습니다.
   - **2.4.0 이전 버전**: License 파일과 4개의 JSON 구성 파일인 `brand_name.json`, `device_config.json`, `phone_info.json`, `score_phone.json` 을 유지해야 합니다.
2. SDK에서 `download_assets.zip` 패키지를 찾습니다.

3. so 파일에 대해 위에서 설명한 것과 같은 방법으로 ZIP 패키지의 MD5 값을 계산한 다음 서버에 패키지를 업로드하여 다운로드 주소를 가져옵니다.
   - Demo에서 다운로드 서비스를 재사용하려면
   - 자체 다운로드 서비스를 이용하려면

©2013-2022 Tencent Cloud. All rights reserved.

i. 다음 그림에서 다운로드 주소와 MD5 값을 업데이트합니다.



ii. `ResDownloadUtil.checkOrDownloadFiles` 를 호출하여 다운로드를 시작하고 `ResDownloadUtil.getValidAssetsDirectory` 를 호출하여 다운로드한 assets의 path를 가져옵니다. 자세한 안내는 `LaunchActivity.java` 를 참고하십시오.

주의 :

- SDK 버전이 업데이트되면 해당 assets도 변경될 수 있으며, 호환성을 위해 assets을 다시 다운로드해야 합니다. Demo의 방법을 참고하여 MD5 값을 검증하는 것을 권장합니다.
- assets을 직접 다운로드하거나 Demo에서 다운로드 서비스를 재사용하는지 여부에 관계없이 비디오를 캡처하기 전에 assets이 다운로드되었는지 확인하십시오. ResDownloadUtil은 다음과 같은 확인 방법을 제공합니다. 다운로드한 경우 SDK에서 경로를 설정합니다. 자세한 안내는 `LaunchActivity.java` 를 참고하십시오.

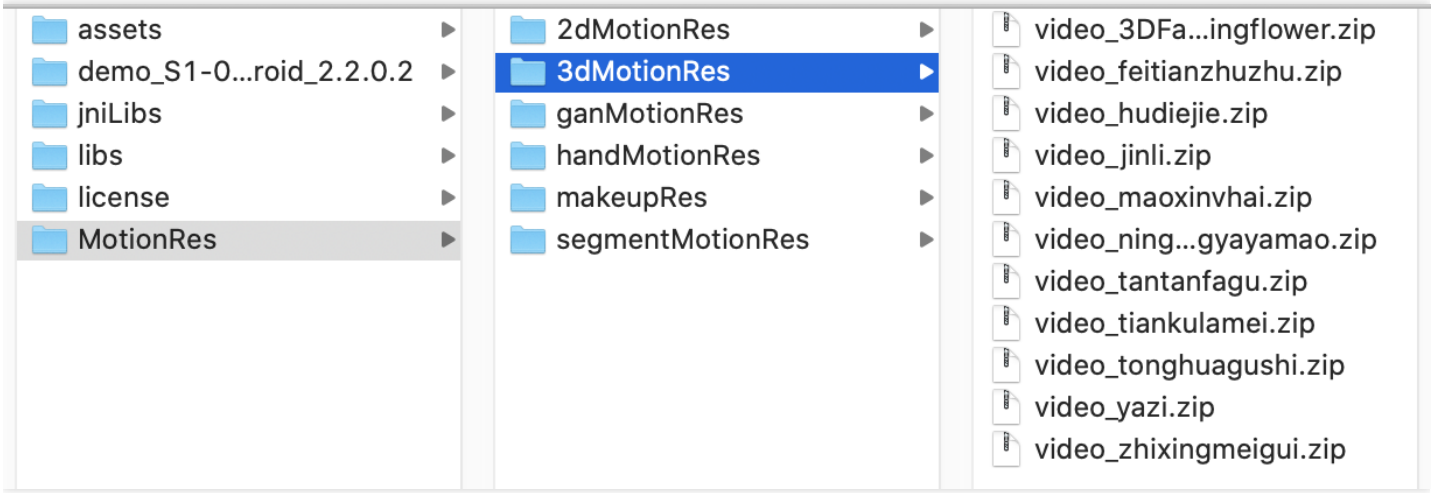
```
String validAssetsDirectory = ResDownloadUtil.getValidAssetsDirectory(LaunchActiv
ity.this,ResDownloadConfig.DOWNLOAD_MD5_ASSETS);
if (validAssetsDirectory == null) {
Toast.makeText(LaunchActivity.this,"assets이 다운로드되지 않았습니다. 먼저 다운로드하십
시오",Toast.LENGTH_LONG).show();
return;
}
XmagicResParser.setResPath(validAssetsDirectory);
startActivity(intent);
```

©2013-2022 Tencent Cloud. All rights reserved.

애니메이션 효과 리소스 MotionRes 다운로드

일부 기본 버전에는 애니메이션 효과 리소스가 없습니다. 실제 상황에 따라 이 섹션을 건너뛸 수 있습니다.

애니메이션 효과는 6가지 유형으로 나뉘며 각 유형에는 애니메이션 효과가 포함된 여러 ZIP 패키지가 있습니다. 파일 내용은 구입한 에디션에 따라 다릅니다.



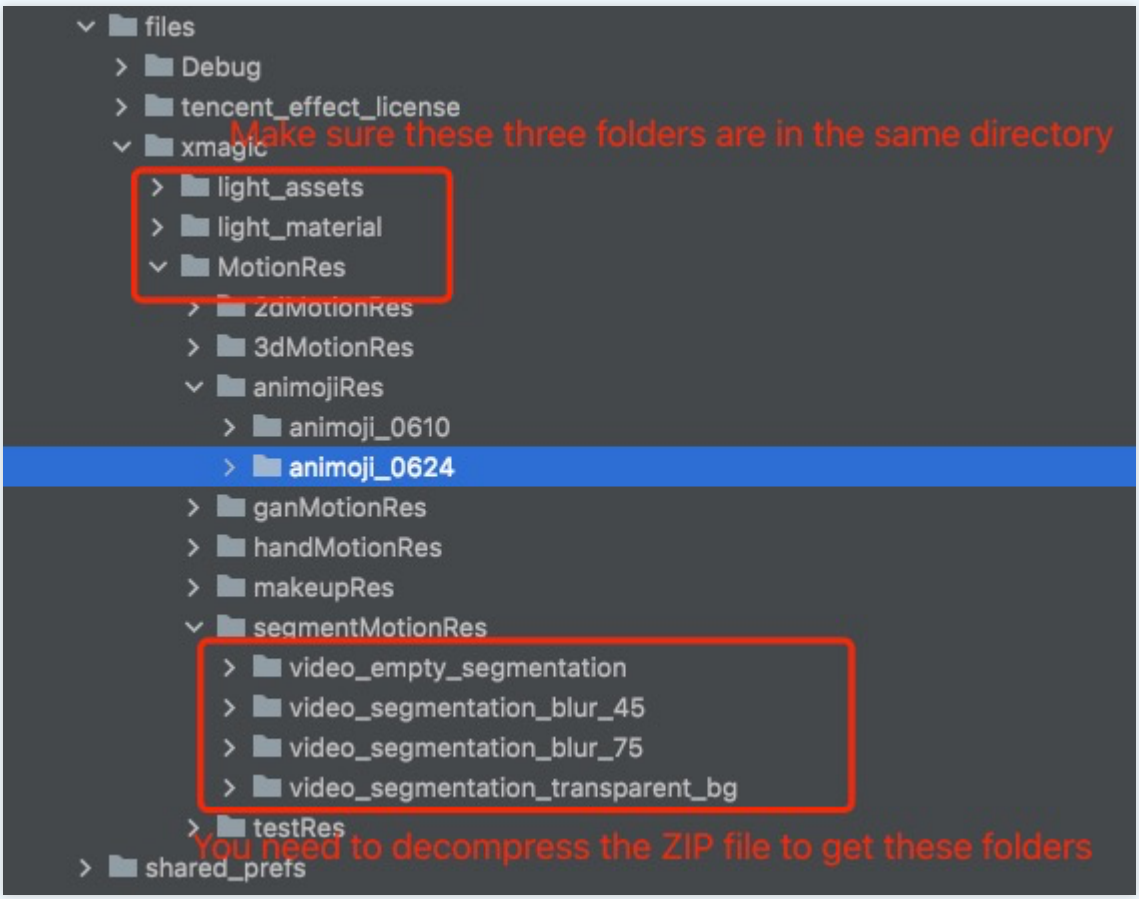
애니메이션 효과 리소스는 필요에 따라 다운로드할 수 있습니다. 예를 들어 사용자가 관련 기능 페이지에 들어간 후 또는 애니메이션 효과의 아이콘을 클릭한 후 다운로드를 시작할 수 있습니다.

이러한 ZIP 패키지를 서버에 업로드하고 각 ZIP 패키지의 다운로드 주소를 가져와야 합니다.

주의：
다운로드한 애니메이션 효과 리소스의 MotionRes 디렉터리는 이전 섹션에서 설명한 light_assets 및 light_material과 동일한 수준에 있어야 합니다. 또한 각 애니메이션 효과는 추출해야 하며 동일한 ZIP 패키지

©2013-2022 Tencent Cloud. All rights reserved.

에 넣을 수 없습니다.



MotionRes를 다운로드하려면 ResDownloadUtil.checkOrDownloadMotions 메소드를 참고하십시오. 이러한 리소스를 하나씩 다운로드하는 것이 좋습니다.

Demo에서 다운로드 서비스를 재사용하려면 ResDownloadConfig의 MOTION_RES_DOWNLOAD_PREFIX 상수 값을 다운로드 URL 접두사로 바꾸십시오.

©2013-2022 Tencent Cloud. All rights reserved.

뷰티 필터 시나리오의 권장 매개변수

최종 업데이트 날짜: : 2022-07-18 10:06:17

다음은 Tencent Effect SDK의 일부 사용 사례에 권장되는 기능 매개변수입니다.

사례1: 내츄럴 룩

내츄럴 뷰티 필터 효과를 원하는 경우 다음 매개변수를 사용합니다.

기능 유형	권장 매개변수
브라이트닝	30
피부 보정	45
안색 보정	20
눈 확대	20
갸름한 얼굴(내츄럴)	30

사례2: 여신 룩

여신 뷰티 필터 효과를 원하는 경우 다음 매개변수를 사용합니다.

기능 유형	권장 매개변수
브라이트닝	60
피부 보정	70
안색 보정	35
눈 확대	40
갸름한 얼굴(내츄럴)	40
안면 윤곽(내츄럴)	50
눈 브라이트닝	30
코 축소	10

©2013-2022 Tencent Cloud. All rights reserved.

사례3: 미남

미남 뷰티 필터 효과를 원하는 경우 다음 매개변수를 사용합니다.

기능 유형	권장 매개변수
미백	25
피부 보정	40
안색 보정	20
갸름한 얼굴(미남)	40

©2013-2022 Tencent Cloud. All rights reserved.

UGSV 엔터프라이즈 버전 마이그레이션 가이드

최종 업데이트 날짜: : 2022-07-18 10:06:18

User Generated Short Video(UGSV) 엔터프라이즈 버전은 단종되었으며 뷰티 필터 모듈은 Tencent Effect(TE) SDK를 구성하기 위해 분리되었습니다. Tencent Effect SDK는 보다 자연스러운 뷰티 효과, 보다 강력한 기능 및 보다 유연한 통합 방법을 제공합니다. 본문에서는 UGSV 엔터프라이즈 버전에서 Tencent Effect SDK로 마이그레이션하는 방법을 설명합니다.

주의 사항

1. xmagic 모듈에서 glide 라이브러리의 버전 번호를 실제 버전 번호와 동일하게 수정합니다.
2. xmagic 모듈의 가장 오래된 버전 번호를 실제 버전 번호와 동일하게 수정합니다.

통합 단계

1단계: Demo 프로젝트 압축 해제

1. TE SDK가 통합된 UGSV Demo를 다운로드합니다. 이 Demo는 TE SDK S1-04 에디션을 기반으로 제작되었습니다.
2. Demo의 SDK 파일을 실제로 사용하는 SDK용 파일로 교체합니다. 구체적 작업은 다음 단계를 따르십시오.
   - xmagic 모듈의 libs 디렉터리에 있는 `.aar` 파일을 SDK의 libs에 있는 `.aar` 파일로 교체합니다.
   - xmagic 모듈의 `../src/main/assets` 에 있는 모든 파일을 SDK의 `assets/` 에 있는 파일로 교체합니다. SDK 패키지의 MotionRes 폴더에 파일이 있으면 `../src/main/assets` 디렉터리에도 복사합니다.
   - xmagic 모듈의 `../src/main/jniLibs` 에 있는 모든 .so 파일을 SDK 패키지의 jniLibs에 있는 .so 파일로 교체합니다. (arm64-v8a 및 armeabi-v7a에 대한 .so 파일을 얻으려면 jinLibs 폴더에 있는 ZIP 파일의 압축을 해제해야 합니다.)
3. Demo 프로젝트의 xmagic 모듈을 실제 항목 프로젝트로 가져옵니다.

2단계: SDK 버전 업그레이드

SDK를 Enterprise 버전에서 Professional 버전으로 업그레이드하십시오.

- **교체 전**: `implementation 'com.tencent.liteav:LiteAVSDK_Enterprise:latest.release'`
- **교체 후**: `implementation 'com.tencent.liteav:LiteAVSDK_Professional:latest.release'`

3단계: 뷰티 필터 License 설정

©2013-2022 Tencent Cloud. All rights reserved.

1. 다음과 같이 프로젝트의 application에서 oncreate 메소드를 호출합니다.

```
XMagicImpl.init(this);
XMagicImpl.checkAuth(null);
```

2. XMagicImpl 클래스의 콘텐츠를 획득한 TE SDK *__License URL 및 Key__로 교체합니다.

4단계: 코드 구현

UGSV 녹화 페이지 TCVideoRecordActivity.java를 예로 듭니다.

1. `TCVideoRecordActivity.java` 클래스에 다음 변수 코드를 추가합니다.

```
private XMagicImpl mXMagic;
private int isPause = 0;//0 일시 중지되지 않음, 1일시 중지됨, 2일시 중지 중, 3종료 예정
```

2. `TCVideoRecordActivity.java` 클래스의 onCreate 메소드 뒤에 다음 코드를 추가합니다.

```
TXUGCRecord instance = TXUGCRecord.getInstance(this);
instance.setVideoProcessListener(new TXUGCRecord.VideoCustomProcessListener() {
@Override
public int onTextureCustomProcess(int textureId, int width, int height) {
if (isPause == 0 && mXMagic != null) {
return mXMagic.process(textureId, width, height);
}
return 0;
}
@Override
public void onDetectFacePoints(float[] floats) {
}
@Override
public void onTextureDestroyed() {
if (Looper.getMainLooper() != Looper.myLooper()) { //메인 스레드가 아님
if (isPause == 1) {
isPause = 2;
if (mXMagic != null) {
mXMagic.onDestroy();
}
initXMagic();
isPause = 0;
} else if (isPause == 3) {
if (mXMagic != null) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
mXMagic.onDestroy();
}
}
}
}
});
XMagicImpl.checkAuth((errorCode, msg) -> {
if (errorCode == TELicenseCheck.ERROR_OK) {
loadXmagicRes();
} else {
TXCLog.e("TAG", "인증 실패, 인증 url 및 key를 확인하십시오" + errorCode + " " + ms
g);
}
});
```

3. onStop 메소드에 다음 코드를 추가합니다.

```
isPause = 1;
if (mXMagic != null) {
mXMagic.onPause();
}
```

4. onDestroy 메소드에 다음 코드를 추가합니다.

```
isPause = 3;
XmagicPanelDataManager.getInstance().clearData();
```

5. onActivityResult 메소드 시작 부분에 다음 코드를 추가합니다.

```
if (mXMagic != null) {
mXMagic.onActivityResult(requestCode, resultCode, data);
}
```

6. 이 클래스의 끝에 다음 두 메소드를 추가합니다.

```
private void loadXmagicRes() {
if (XMagicImpl.isLoadedRes) {
XmagicResParser.parseRes(getApplicationContext());
initXMagic();
return;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
new Thread(() -> {
XmagicResParser.setResPath(new File(getFilesDir(), "xmagic").getAbsolutePath
());
XmagicResParser.copyRes(getApplicationContext());
XmagicResParser.parseRes(getApplicationContext());
XMagicImpl.isLoadedRes = true;
new Handler(Looper.getMainLooper()).post(() -> {
initXMagic();
});
}).start();
}
/**
```

- 뷰티 필터 SDK 초기화
- /

```
private void initXMagic() {
if (mXMagic == null) {
  mXMagic = <span class="hljs-keyword">new</span> XMagicImpl(<span class="hljs-ke
  yword">this</span>, mUGCKitVideoRecord.getBeautyPanel());
}else{
  mXMagic.onResume();
}
}
```

5단계: 다른 클래스 수정

1. AbsVideoRecordUI 클래스의 mBeautyPanel 유형을 RelativeLayout 유형으로 변경하고 getBeautyPanel() 메소드의 응답 유형을 RelativeLayout으로 변경합니다. 또한 오류를 보고하는 코드를 주석 처리하려면 해당 XML 구성을 수정해야 합니다.
2. UGCKitVideoRecord 클래스에서 오류를 보고하는 코드를 주석 처리합니다.
3. ScrollFilterView 클래스의 코드를 수정하여 mBeautyPanel 변수를 삭제하고 오류를 보고하는 코드를 주석 처리합니다.

©2013-2022 Tencent Cloud. All rights reserved.

6단계: beautysettingkit 모듈에 대한 종속성 삭제

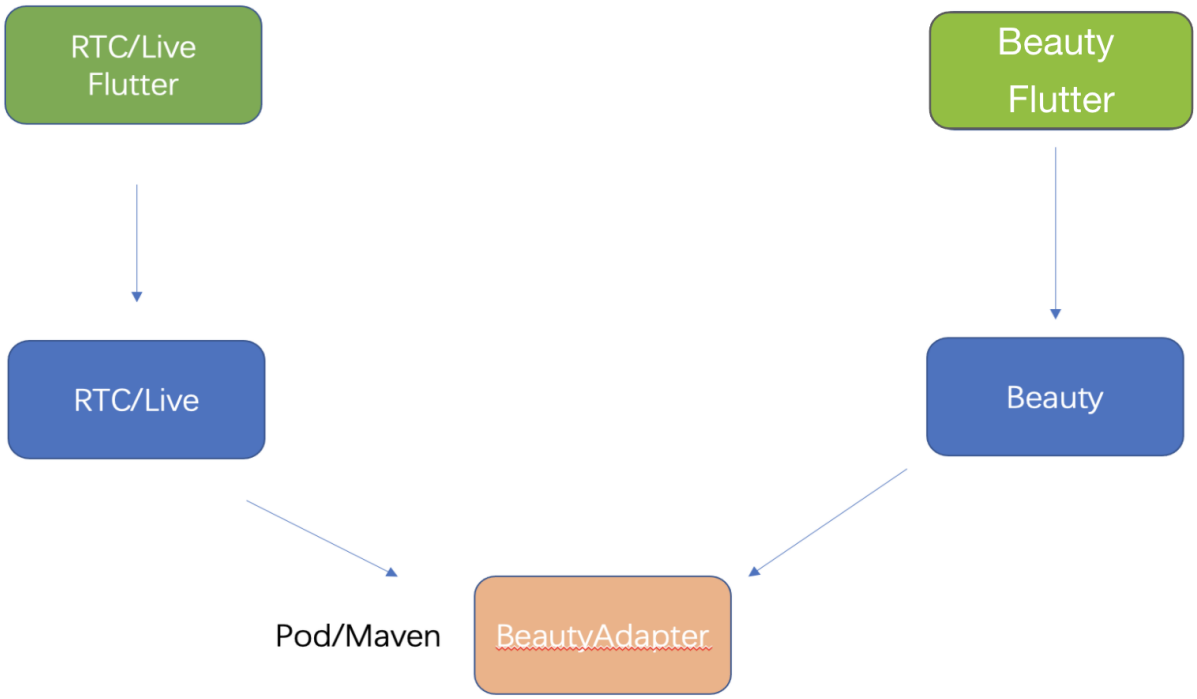
ugckit 모듈의 build.gradle 파일에서 beautysettingkit 모듈에 대한 종속성을 삭제하고 프로젝트를 컴파일하여 오류를 보고하는 코드를 주석 처리합니다.

©2013-2022 Tencent Cloud. All rights reserved.

서드 파티 퍼블리셔 Tencent Effect 통합 (Flutter)

최종 업데이트 날짜: : 2022-12-15 11:30:53

Flutter OpenGL 환경은 네이티브 환경과 격리되어 있으므로 Tencent Effect SDK를 Flutter에 직접 통합할 수 없습니다. 네이티브 측에서 이들 사이에 연결을 설정해야 합니다.



전반적인 구현 과정

1. Tencent Effect SDK 측에서 API 추상화 레이어를 생성하고 API를 구현합니다.
2. 애플리케이션이 시작되면 타사 게시자가 API를 사용하여 뷰티필터 인스턴스를 생성, 사용 및 종료할 수 있도록 타사 게시자에게 API를 등록합니다.
3. 타사 게시자는 뷰티필터 인스턴스를 만들고 종료하는 기능을 Flutter 측에 노출합니다.
4. Tencent Effect Flutter SDK를 사용하여 뷰티필터를 구성합니다.

TRTC를 예로 들면

©2013-2022 Tencent Cloud. All rights reserved.

Tencent Effect 측에서 정의한 API:

```
public interface ITXCustomBeautyProcesserFactory {
/**
* 뷰티필터 인스턴스 생성
* @return
*/
ITXCustomBeautyProcesser createCustomBeautyProcesser();
/**
* 뷰티필터 인스턴스 종료(이 API는 OpenGL 스레드에서 호출되어야 함)
*/
void destroyCustomBeautyProcesser();
}
public interface ITXCustomBeautyProcesser {
//비디오 프레임에 지원되는 픽셀 형식을 가져옵니다. Tencent Effect는 OpenGL 2D 텍스처를 지원합니다.
TXCustomBeautyPixelFormat getSupportedPixelFormat();
//비디오 프레임에 지원되는 컨테이너 형식을 가져옵니다. Tencent Effect는 최고의 성능을 제공하고 비디오 품질에 미치는 영향이 가장 적은 V2TXLiveBufferTypeTexture를 지원합니다.
TXCustomBeautyBufferType getSupportedBufferType();
//OpenGL 스레드에서 이 API를 호출합니다(srcFrame은 RGBA 텍스처와 width 및 height를 포함해야 함). 처리 후 텍스처 객체는 dstFrame의 texture.textureId에 포함됩니다.
void onProcessVideoFrame(TXCustomBeautyVideoFrame srcFrame, TXCustomBeautyVideoFrame dstFrame);
}
```

1. TRTC는 등록 방법을 제공합니다. 애플리케이션이 실행되면 ITXCustomBeautyProcesserFactory의 구현 클래스인 'com.tencent.effect.tencent_effect_flutter.XmagicProcesserFactory'를 TRTC(네이티브 측)에 등록합니다.

©2013-2022 Tencent Cloud. All rights reserved.

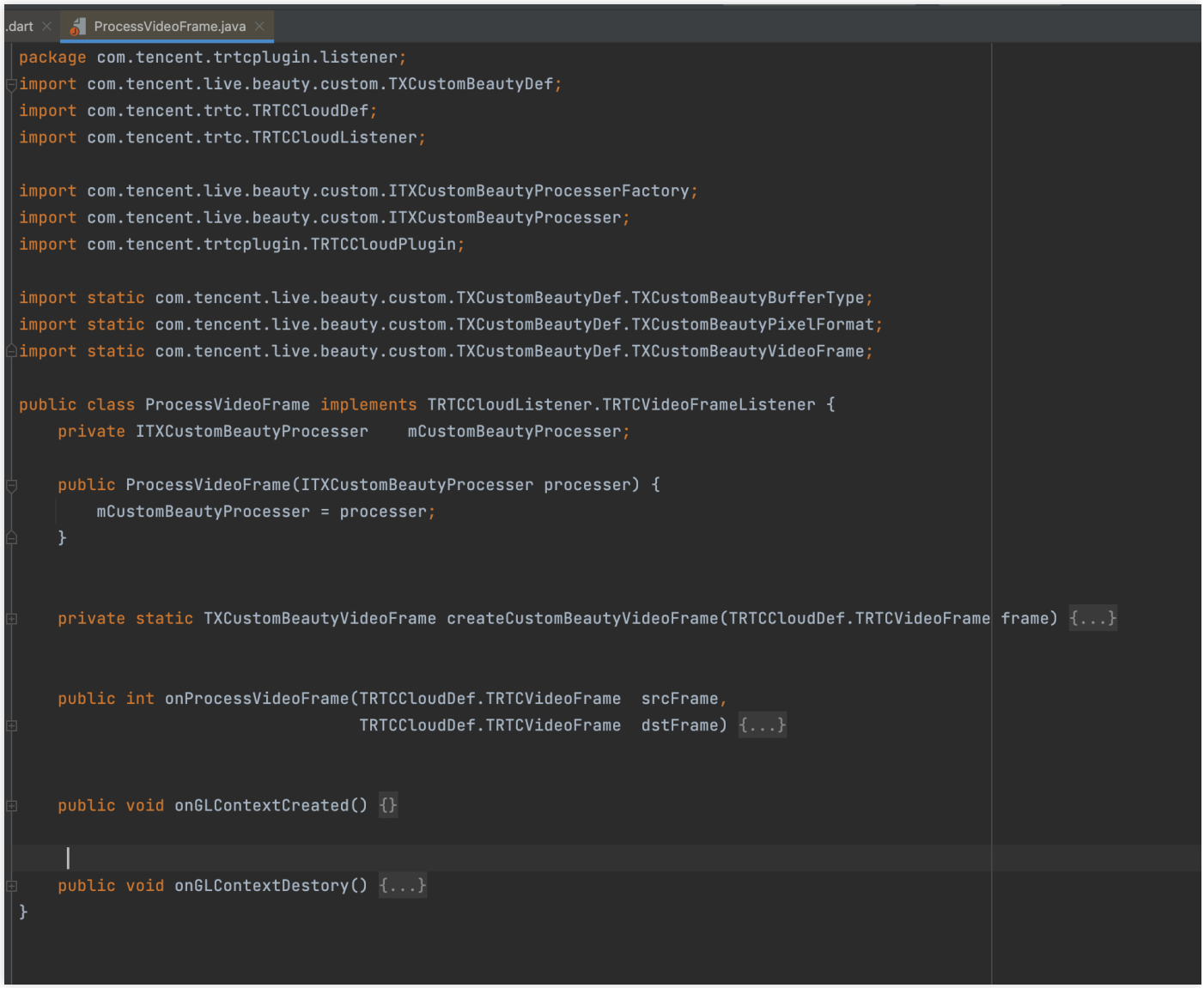
```java
package com.tencent.effect.tencent_effect_flutter_example;

import android.os.Bundle;

import androidx.annotation.Nullable;

import com.tencent.effect.tencent_effect_flutter.XmagicProcesserFactory;
import com.tencent.live.TXLivePluginManager;   kevinxlhua, 2022/5/9, 3:38 下午 • 添加对 直播flutter版本的依赖，由于现在直

import io.flutter.embedding.android.FlutterActivity;
import com.tencent.trtcplugin.TRTCCloudPlugin;

public class MainActivity extends FlutterActivity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TXLivePluginManager.register(new XmagicProcesserFactory());
        TRTCCloudPlugin.register(new XmagicProcesserFactory());
    }
}
```

2. `Flutter` 레이어에서 사용자 지정 효과를 활성화 또는 비활성화하는 데 사용되는 `Future<v2txlivecode> enableCustomVideoProcess(bool enable)` 를 제공합니다.
3. TRTC 네이티브 측에서 효과를 활성화 또는 비활성화합니다.

```java
public void enableCustomVideoProcess(MethodCall call, MethodChannel.Result result) {
    boolean enable = CommonUtil.getParam(call, result, "enable");
    ITXCustomBeautyProcesserFactory processerFactory = TRTCCloudPlugin.getBeautyProcesserFactory();
    mCustomBeautyProcesser = processerFactory.createCustomBeautyProcesser();
    TXCustomBeautyBufferType bufferType = mCustomBeautyProcesser.getSupportedBufferType();
    TXCustomBeautyPixelFormat pixelFormat = mCustomBeautyProcesser.getSupportedPixelFormat();
    if(enable) {
        ProcessVideoFrame processVideo =  new ProcessVideoFrame(mCustomBeautyProcesser);
        int ret = trtcCloud.setLocalVideoProcessListener(convertTRTCPixelFormat(pixelFormat), convertTRTCBufferType(bufferType), processVideo);
        result.success(ret);
    } else {
        int ret = trtcCloud.setLocalVideoProcessListener(convertTRTCPixelFormat(pixelFormat), convertTRTCBufferType(bufferType), null);
        // processerFactory.destroyCustomBeautyProcesser();
        mCustomBeautyProcesser = null;
        result.success(ret);
    }
}
```

When set to null, the onGlContextDestroy method of the last set processVideo will be called back

©2013-2022 Tencent Cloud. All rights reserved.



```
.dart  ProcessVideoFrame.java
package com.tencent.trtcplugin.listener;
import com.tencent.live.beauty.custom.TXCustomBeautyDef;
import com.tencent.trtc.TRTCCloudDef;
import com.tencent.trtc.TRTCCloudListener;

import com.tencent.live.beauty.custom.ITXCustomBeautyProcesserFactory;
import com.tencent.live.beauty.custom.ITXCustomBeautyProcesser;
import com.tencent.trtcplugin.TRTCCloudPlugin;

import static com.tencent.live.beauty.custom.TXCustomBeautyDef.TXCustomBeautyBufferType;
import static com.tencent.live.beauty.custom.TXCustomBeautyDef.TXCustomBeautyPixelFormat;
import static com.tencent.live.beauty.custom.TXCustomBeautyDef.TXCustomBeautyVideoFrame;

public class ProcessVideoFrame implements TRTCCloudListener.TRTCVideoFrameListener {
    private ITXCustomBeautyProcesser    mCustomBeautyProcesser;

    public ProcessVideoFrame(ITXCustomBeautyProcesser processer) {
        mCustomBeautyProcesser = processer;
    }


    private static TXCustomBeautyVideoFrame createCustomBeautyVideoFrame(TRTCCloudDef.TRTCVideoFrame frame) {...}


    public int onProcessVideoFrame(TRTCCloudDef.TRTCVideoFrame  srcFrame,
                                   TRTCCloudDef.TRTCVideoFrame  dstFrame) {...}


    public void onGLContextCreated() {}

    public void onGLContextDestory() {...}
}
```

부록

Tencent Effect에서 제공하는 추상화 레이어 의존성

```
///
implementation 'com.tencent.liteav:custom-video-processor:latest.release'
```

©2013-2022 Tencent Cloud. All rights reserved.