

# **User Generated Short Video SDK Integration (UI Included) Product Documentation**



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

## Trademark Notice



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

## Service Statement

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

# Contents

Integration (UI Included)

UGSV Demo App

Source Code Description (iOS)

Source Code Description (Android)

# Integration (UI Included)

## UGSV Demo App

Last updated : 2022-11-14 18:18:58

### Client

You can use Xcode or Android Studio to compile and debug the UGSV demo app's source code as follows. The execution effect is as shown below:



### Step 1. Download the app source code

Download the UGSV demo app's source code [here](#).

### Step 2. Apply for the SDK license

For more information, see [License Application](#).

### Step 3. Prepare the debugging environment

- **iOS**
  - Xcode 9 or later
  - iOS 12 or later
- **Android**
  - Android SDK Tools: android-sdk\_26.0.2
    - minSdkVersion: 21
    - targetSdkVersion: 26

### Step 4. Compile and run

Click **Build** in Xcode or Android Studio to compile and run the source code. Tencent's test server address

`http://demo.vod2.myqcloud.com/lite/` is configured in the source code, so you can quickly run the UGSV app in the debugging environment.

## Backend

The UGSV demo app depends on two backend services:

- **VOD**

VOD offers video storage and online delivery capabilities for the UGSV demo app. The basic edition of the UGSV license comes with the VOD plan and a free tier of traffic.
- **CVM**

The UGSV demo app requires a simple business server to provide capabilities such as signup, login, video list storage, and video upload signature. You can set up the server in a CVM instance and modify the logic as needed.

If you use the default server address `http://demo.vod2.myqcloud.com/lite/` in the UGSV demo app's source code, both the VOD service and list server are provided by Tencent Cloud. However, they can be used only for debugging and trial as there is a limit on the concurrent requests.

You can also set up your own backend server as follows:

### Step 1. Activate VOD

1. Log in to the [VOD console](#) to activate the VOD service, which can provide video storage and online playback capabilities for the UGSV demo app.
2. In [Callback Settings](#) in the VOD console, set the callback mode to **Reliable Callback** and select **Callback on video upload completion** in **Callback Event Configuration**. The settings will take effect in about ten minutes.

Event Notification Method

☐ Normal Callback ☒ Reliable Callback

Event Notification

☒ Finished video uploading

☒ Deleted the video

☒ Finished video composition

☒ Task flow status changed

☒ Finished video editing

▲ Legacy Types①

☒ Finished video transcoding

☒ Finished screencapturing by time points in the video

☒ Finished video splicing

☒ Finished video clipping

☒ Finished screencapturing of image sprite in the video

Confirm

Cancel

## Step 2. Get the TencentCloud API key.

When the UGSV demo app uploads a video, it requires a Tencent Cloud key, i.e., `SecretId` and `SecretKey`, which need to be obtained from the Tencent Cloud console and configured on the business server.

1. Log in to the console and select **Products > Cloud Access Management> Manage API Key** to enter the **Manage API Key** page.
2. Get the TencentCloud API key. If you haven't created a key, click **Create Key** to create a pair of `SecretId` and `SecretKey`.

APPID	Key	Creation Time	Last Access Time	Status	Operation
1311	SecretId: IKIDa0/ SecretKey: ***** <a href="#">Show</a>	v3KM8f	2022-05-18 16:09:54	-	On <a href="#">Disable</a>

## Step 3. Deploy the backend code on the CVM instance

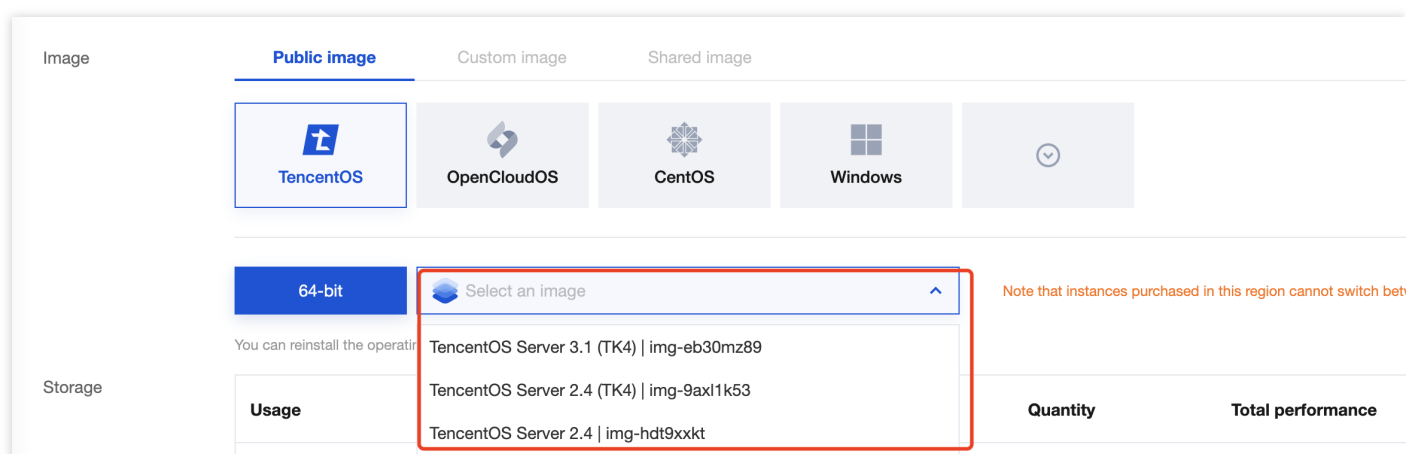
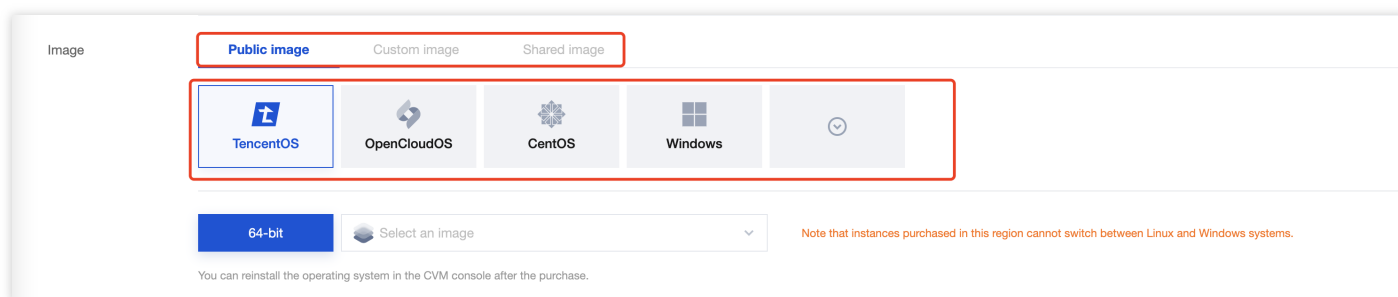
1. Create a CVM instance in the [CVM console](#).

Create a Reserved Instance

Separate keywords with "|", and separate tags using the Enter key

Reserved Instance ID/Name	Status	Availability Zone	Instance Type	Specifications	Operating System	Number of Instances	Payment Type	Effective date/Expiry date	Operation
No instances found. You can <a href="#">create</a> an instance.									
Total Items: 0									
20 / page 1 / 1 page									

## 2. Select Custom Configuration and enter the image marketplace to select an image.



## 3. Configure the disk, network, and CVM access password. Keep the password secure. Then, configure a security group.

Cloud Virtual Machine (CVM) Product

Custom configuration

✓ Select basic configurations

2 Configure network and host

3 Confirm configuration

Network and bandwidth

Network

Default-VPC (Default)

Default-Subnet (Default)

Available IPs in the subnet: 4093

The current network is the default VPC/subnet. Please adjust it based on your own needs.

If the existing VPCs/subnets do not meet your requirements, [create a VPC](#) or [a subnet](#) in the console. You can also change the VPC and subnet later.

Public network IP

☒ Get a free public IP

Bandwidth billing mode

By traffic

Note: The traffic fee is settled on an hourly basis. When your account balance becomes negative, the service will be stopped in 2 hours.

Bandwidth

1Mbps

10Mbps

20Mbps

100Mbps

5

Mbps

Security group

New security group

Existing security group

Select security group(s) (up to five)

Please make sure port 22 (Linux SSH login) and port 3389 (Windows remote login) are open in the selected security group. You can go to the CVM console to [Modify security group](#)

Other settings

Tag

Tag key

Tag value

删除

+ Add

#### 4. Log in to the created CVM instance.

You can click **Log In** in the **Operation** column of the instance to access the instance through Tencent Cloud's web shell or use **PuTTY** or **SecureCRT** to log in to it over SSH.

#### 5. Modify the CVM instance configuration information.

Set `appId`, `SecretId`, and `SecretKey` in the following script to the `APPID`, `SecretId`, and `SecretKey` obtained in [step 2](#). Then, log in to the CVM instance and directly run the modified script in the instance.

5.1 Modify and copy the following configuration locally. Then, log in to the CVM instance and paste and run the code in the console.

```
echo '{
  "dbconfig": {
    "host": "127.0.0.1",
    "user": "litvideo",
    "password": "litvideo",
    "database": "db_litvideo",
    "port": 3306,
```



```
"supportBigNumbers": true,
"connectionLimit":10
},
"tencentyunaccount":{
"appid":"Your AppId",
"SubAppId":"",
"SecretId": "Your SecretId",
"SecretKey": "Your SecretKey",
"bucket":"xiaoshipin",
"region":"ap-guangzhou"
},
"server":{
"ip":"0.0.0.0",
"port":8001,
"reliablecb":true
}
}' > /home/ubuntu/vod-xiaoshipin-server/conf/localconfig.json
```

6. ii. In the instance, run the following command to directly start the service. The default port for starting the service is 8001 .

- Start the service:

```
cd /home/ubuntu/vod-xiaoshipin-server/;pm2 start app.js --name 'litvideo';
```

- To stop or restart the service, run the following commands:

Restart the service:

```
pm2 restart litvideo;
```

Stop the service:

```
pm2 delete litvideo;
```

5.3 View the CVM instance's public IP in [step 4](#) and enter `http://IP` in the browser to check whether the service has started successfully.

## Step 4. Replace the backend address in the terminal source code

- iOS:** After decompressing the source code package, change the `kHttpServerAddr` in the `iOS/Demo/XiaoShiPin/TCConstants.h` file to the public IP address of your CVM instance.

- **Android:** After decompressing the source code package, change the `APP_SVR_URL` in the `XiaoShiPin_Professional_Android/Demo/ugckit/src/main/java/com/tencent/qcloud/ugckit/UGCKitConstants.java` file to the public IP address of your CVM instance.

Note :

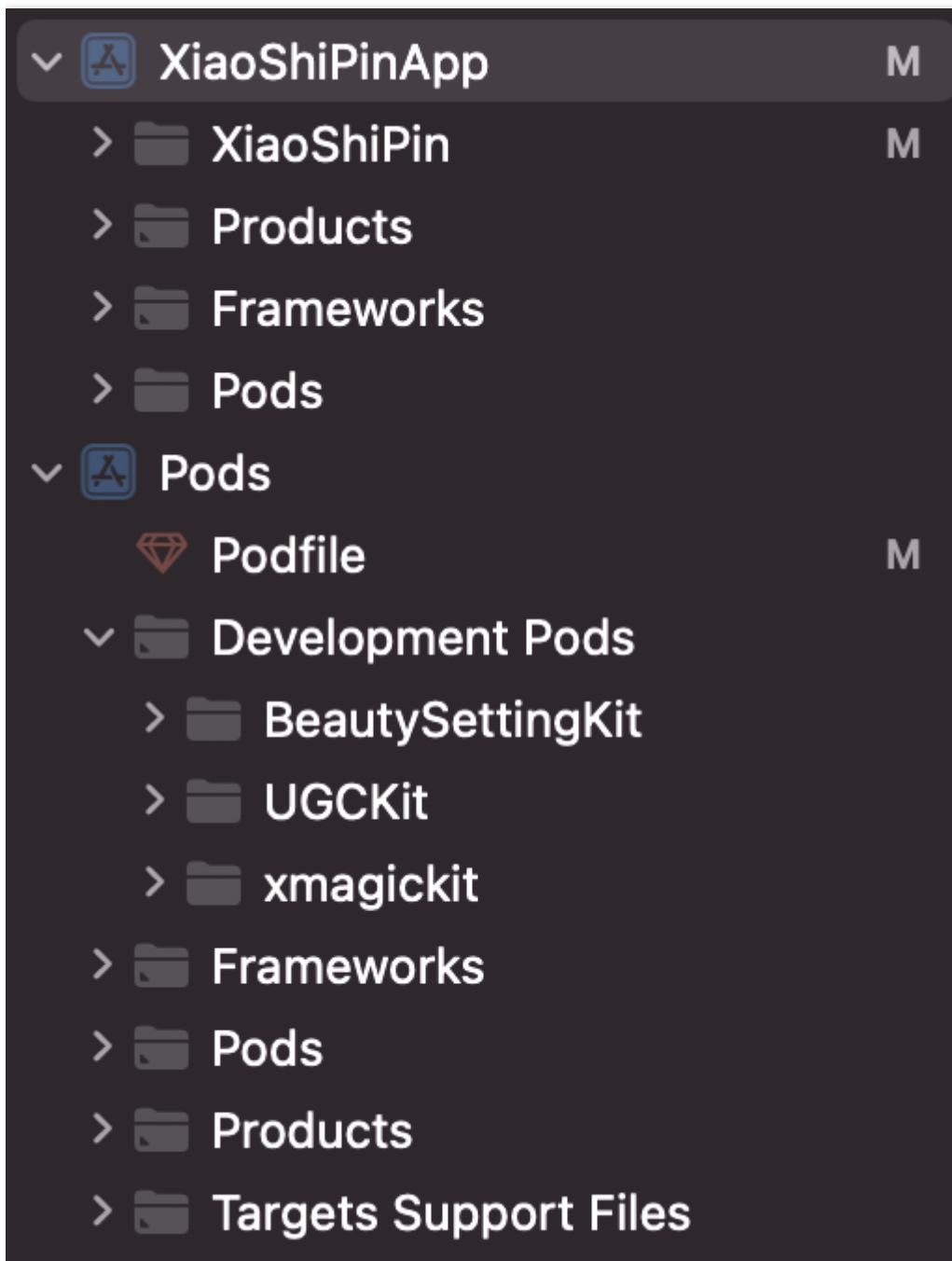
If no certificate is configured for the CVM instance, then HTTP, not HTTPS, must be used in the CVM address.

At this point, the server mode of the UGSV demo app has been configured, and you can run the app to try out its features.

# Source Code Description (iOS)

Last updated : 2022-11-14 18:18:58

## Project Structure



The UGSV demo app mainly integrates `UGCKit` as the core feature library, which is used for playback and shooting. For the integration method, see [SDK Integration \(Xcode\)](#).

The UGSV demo app's beauty filter feature consists of basic beauty filters and effects from Tencent Effect. You can implement basic beauty filters by integrating `BeautySettingkit` as instructed in [Special Effects \(iOS\)](#), and the code is in the `BeautySettingKit` directory. You can implement Tencent Effect by integrating `xmagickit` as instructed in [SDK Integration Guide \(iOS\)](#), and the code is in the `xmagickit` directory.

## Module Overview

The UGSV demo app has seven modules:

- The account, list management, publishing, and profile modules are under the `XiaoshiPinApp` directory.
- The video playback, shooting, and editing modules are parts of UGCKit.

### Account module

- The account module handles the logics of user login/registration and login data caching.
- You can replace this module with your own account system if you already have one.
- The UI of the account module is implemented by `XiaoShiPin/AppViewControllers/Account` and `XiaoShiPin/AppViewControllers/AccountInfo`. The former is responsible for login, and the latter is responsible for profile processing.
- You can view the business logic of the account module in `XiaoShiPin/Model`.

### Main UI and list management

#### Description

- The main UI offers access to three primary features of the app: video list, video shooting, and user profile.
- By default, users log in to the video list page. When they tap the shooting tab, they will be directed to the login page if they are not logged in or the publishing page if they are logged in. If they tap the profile tab, the user profile page will be displayed.
- The list management module pulls and displays the video list.

#### Code

- Model:
  - `TCLiveListModel( XiaoShiPin/AppViewControllers/mainList )`: Data-level definition and pickling/unpickling of the video list
- UI:
  - `TCNavigationController`: Controller for the navigation bar, used to customize the background color of the navigation bar

- `TCMainViewController`: Controller for the tab bar of the main UI, used to switch among the video list, shooting, and user profile pages
- `TCVideoListCell`: Cell class of the VOD list, which displays the thumbnail, title, and username
- `TCLiveListViewController`: `TableViewController` for the video list, which displays videos and directs users to the playback page when they tap a video

## Playback module

### Description

The playback module offers features including video preloading, playback, caching, sharing, etc.

### Code

You can find the code in `/XiaoShiPin/AppViewControllers/VideoPlayer` , which includes the UI logic and business logic for playback.

## Shooting module

### Description

The shooting module offers features including segment-based shooting and deletion, multi-resolution shooting, changing shooting speed, etc.

### Code

You can find the logic for shooting in `UGCKit/Source/Record` .

## Editing module

### Description

The video editing module offers features including video clipping, background music, filters, special effects, animated and static stickers, etc.

### Code

You can find the logic for video editing in `UGCKit/Source/Edit` .

## Publishing module

### Description

The publishing module offers the video publishing feature.

### Code

You can find the logic for publishing in `XiaoShiPin/AppViewController/Publish` .

## Profile module


















### Description

- The user profile module is responsible for displaying, saving, and modifying user profiles and syncing the data to the server.
- The user profile contains a user profile photo, nickname, and gender information.
- The user profile module syncs the latest user profile data from the server to your app. It allows users to view their profiles, including photos, names, and gender.
- Users can modify their profiles. The changes will be synced to the server.
- Other modules can obtain and modify user profiles from this module.

### Code

You can find the logic for user profiles in `XiaoShiPin/AppViewController/AccountInfo` .

ugckit

- ▼  UGCKit
  - >  Pod
  - ▼  Professional
    - h SDKHeader
    - h TXLiteAVSDKHeader
  - >  UGCKitResources
  - >  Localizable
  - ▼  Source
    - h UGCKit
    - >  Common
    - >  Edit
    - >  MediaPlayer
    - >  Model
    - >  Music
    - >  Record
    - >  Report
    - >  Theme
    - >  VideoCut
  - >  UGCKitResources
  - >  Support Files

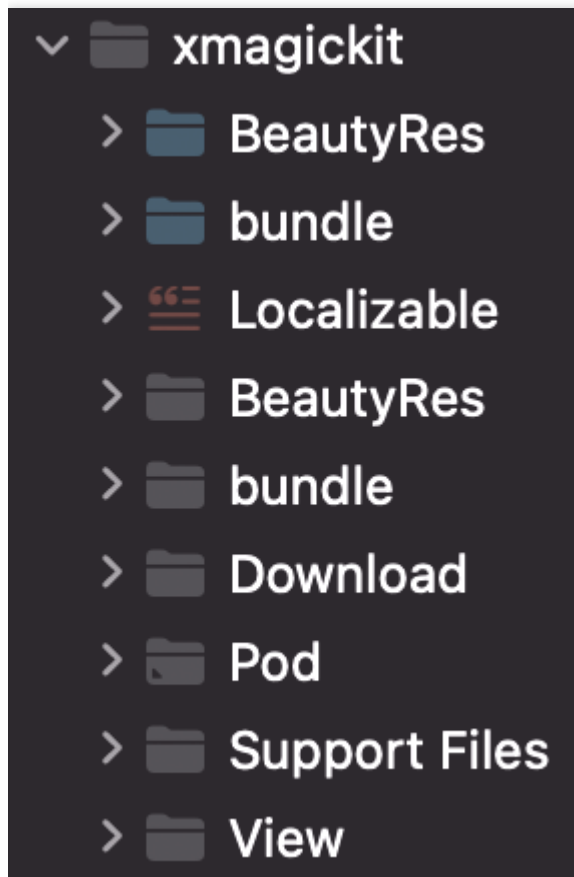
This module is the advanced encapsulation (with the UI) of the UGSV SDK to facilitate quick integration.

Directory description:

File/Directory	Description
Source/Common	The custom view module
Source/Edit	The video editing module
Source/MediaPicker	The media selection module
Source/Model	The media model module
Source/Music	The music module
Source/Record	The video shooting module
Source/Report	The data reporting module
Source/Theme	The resource theme module
Source/VideoCut	The video clipping module

## xmagickit

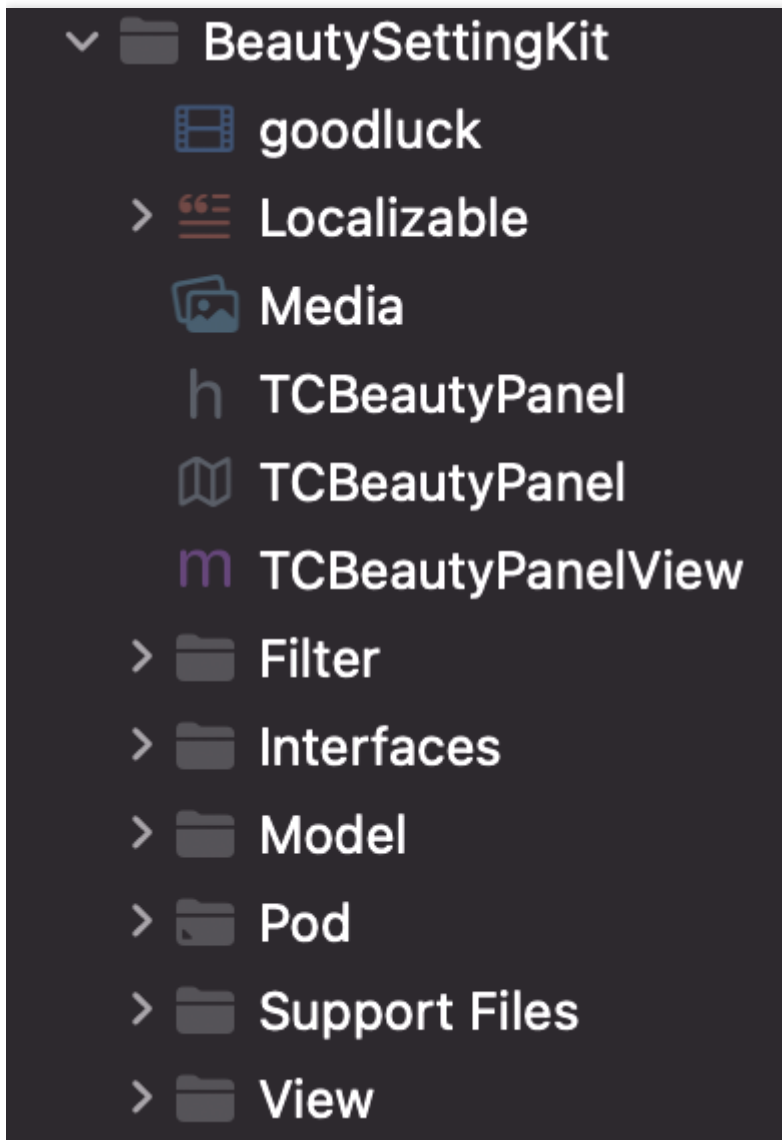




This module encapsulates Tencent Effect so that it can quickly integrated.

File/Directory	Description
BeautyRes	Image resources
bundle	Beauty filter materials
Download	The download module
View	The beauty filter panel and data

## beautysettingkit



This module is the basic beauty filter module of UGSV. To use basic beauty filters, you can use this module for quick integration.

File/Directory	Description
Filter	The filter module
Interfaces	Beauty filter APIs
Model	Beauty filter data models
View	The beauty filter panel UI

# Source Code Description (Android)

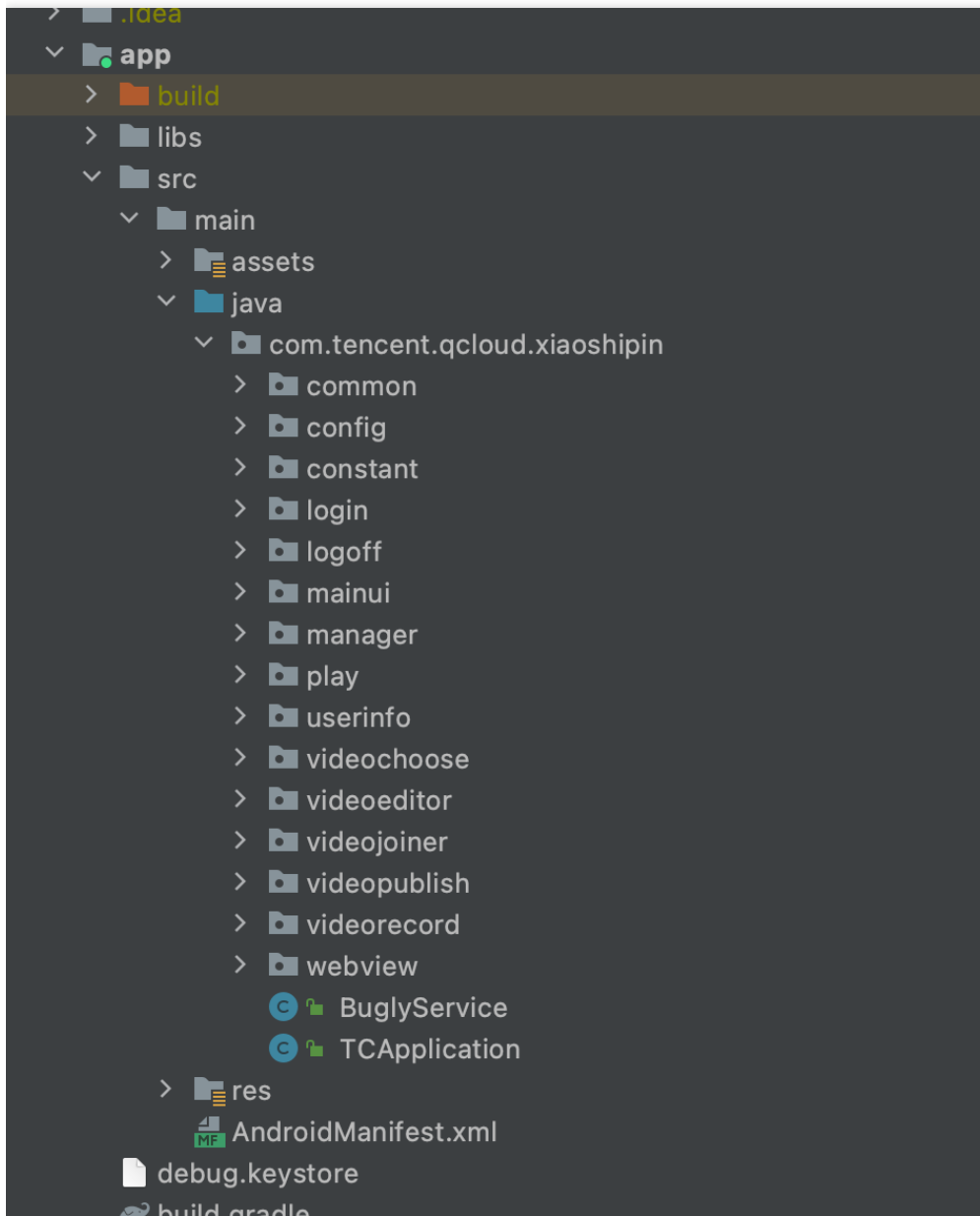
Last updated : 2022-11-14 18:18:58

## Project Structure

The project contains four modules: [app](#), [ugckit](#), [xmagickit](#), and [beautysettingkit](#).

### app

#### Directory overview



File/Directory	Description
common	Common components, including tools and custom UIs (beauty filter and player controls)
login	The account module, including login and registration
logoff	The account cancellation UI
mainui	The main UI of the UGSV demo app, including main activities and the video list
manager	The permission processing module
play	The VOD playback module

File/Directory	Description
userinfo	The profile module
videochoose	The video file selection module
videoeditor	The video editing module
videojoiner	The video composition module
videopublish	The video publishing module
videorecord	The video shooting module
webview	The module for loading HTML5 mobile web pages

## Module overview

The UGSV demo app's functional modules include account, main UI, playback, and UGSV (editing, composition, shooting, and publishing), which correspond to different codes. Below is a description of these modules plus their implementation methods.

### Account module

- The account module handles the logics of user login/registration and login data caching.
- You can replace this module with your own account system if you already have one.
- The account module registers usernames and passwords with the backend of the UGSV demo app by calling `register` of `TCUserMgr`, handles logins by calling `login` of `TCUserMgr`, caches login data locally in `SharePreference`, and clears the cached data after logout.

### Code:

Class	Description
TCApplication.java	SDK initialization class
TCLoginActivity.java	User login view
TCRegisterActivity.java	User registration view
TCUserMgr	User login/registration management class
ProgressFragmentUtil	Control for displaying the progress bar
TCUserInfoFragment.java	User profile view

Class	Description
TCAboutActivity	"About" view of the UGSV demo app

### Main UI and list management

- The main UI offers access to three primary features of the app: video list, video shooting/editing, and user profile.
- By default, users log in to the video list page. When they click **+**, a dialog box pops up asking them to select whether to shoot new videos or edit local videos. They can also click the user profile button to go to the user information page.
- The list management module pulls and displays the video list.

#### Code:

Class	Description
TCSplashActivity.java	Splash screen
TCMainActivity.java	The main UI, which includes the video list, video recording/editing and user profile pages
TCLiveListMgr.java	List management class, which provides APIs to get local videos and update the video list from the server
TCLiveListFragment.java	List view, which displays video information
TCLiveListAdapter.java	Video list adapter
TCUGCVideoListAdapter.java	Video list adapter
TCVideoInfo	Video data

### Short video shooting module

With the UGSV demo app, users can shoot [UGSV](#) not longer than one minute, but the SDK does not show video length.

#### Code:

Class	Description
TCVideoRecordActivity.java	Video shooting view
RecordProgressView	View for pressing and holding to shoot videos
ComposeRecordBtn	Progress bar for multi-segment shooting

## File selection module

This module lists all the MP4 files stored locally on phones for users to choose from.

### Code:

Class	Description
TCVideoChooseActivity.java	Local MP4 file selecting view
TCVideoEditorListAdapter.java	Local MP4 file list adapter
TCVideoEditorMgr.java	MP4 file management class, which provides an API to get the MP4 files stored on mobile phones
TCVideoFileInfo.java	Local video data

## Editing module

The [video editing](#) module offers features including video clipping, slow motion, filters, music mixing, stickers and subtitling.

### Code:

- **\*\*videoeditor/ directory: \*\***

Class	Description
TCVideoPreprocessActivity.java	Video pre-processing view, which is displayed when users start editing a video after shooting
TCVideoCutterActivity.java	Video clipping view
TCVideoEditorActivity.java	The editing view displayed after video clipping, with options including music, filters, speed adjustment, tone, stickers and subtitles at the bottom
TCVideoEffectActivity.java	The special effect adding view, which shows when users click the button at the bottom
BaseEditFragment.java	Parent class of the special effect fragment, which controls the playback of special effects on multiple views
TCVideoJoinerActivity.java	If users select multiple files to edit, the SDK will compose the videos.

- **videoeditor/cutter/directory:** Video clipping
- **videoeditor/time/ directory:** Speed adjustment, including slow motion, loop and rewind.
- **videoeditor/bgm/ directory:** Background music
- **videoeditor/paster/ directory:** Stickers, including animated and static stickers

- **videoeditor/motion/ directory:** Four animated filters. You cannot add your own filters. To use more filters, contact us by submitting a ticket.
- **videoeditor/bubble/ directory:** Bubble subtitles
- **videoeditor/utils/ directory:** Video editing tools
- **videoeditor/common/directory:** Common video editing components
- **videojoiner/directory:** Video composition

### Short video publishing module

This module allows users to publish videos to Tencent Cloud's video distribution platform (VOD system).

#### Code:

Class	Description
TCVideoPublisherActivity.java	Video publishing view

### Short video playback

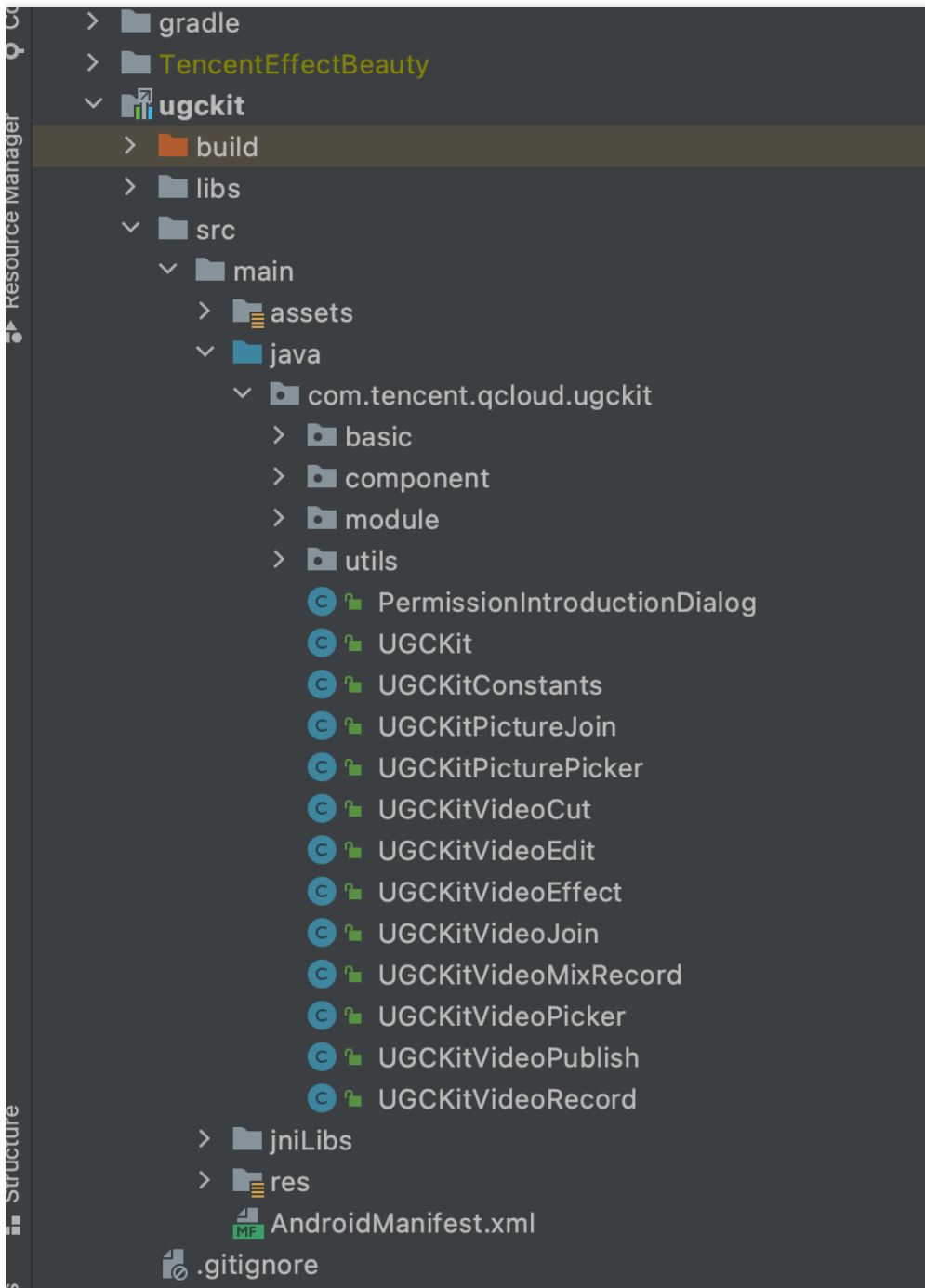
This module allows users to swipe up and down to play videos that have been published to the VOD system.

#### Code:

Class	Description
TCVodPlayerActivity.java	UGSV playback view

## ugckit





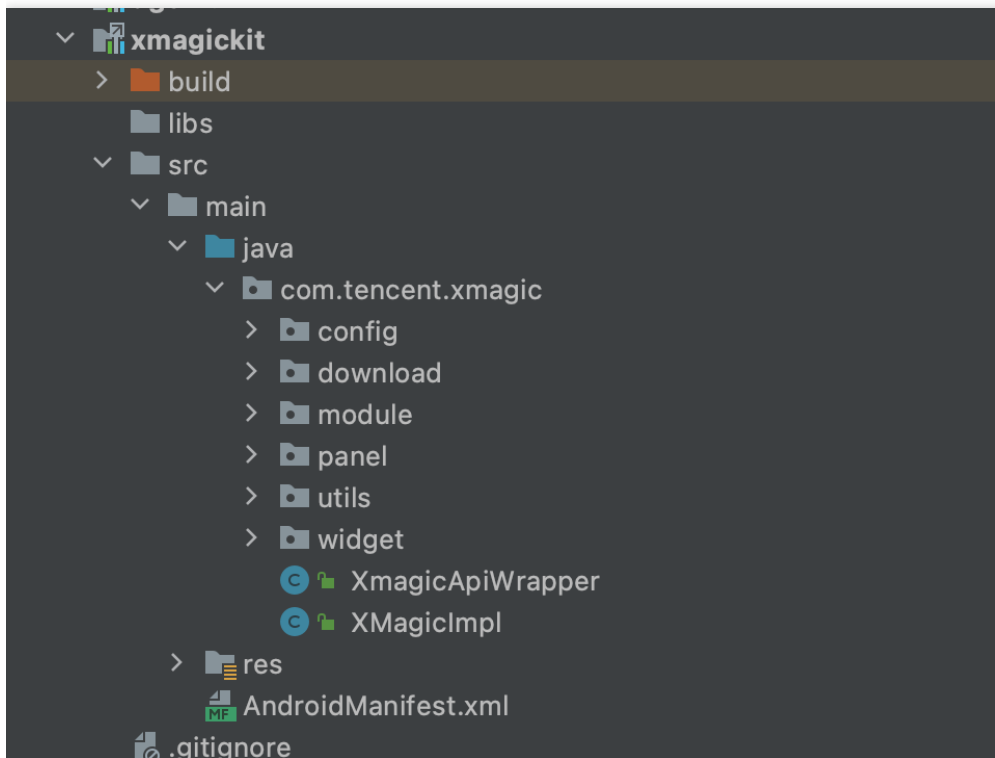
This module is the advanced encapsulation (with the UI) of the UGSV SDK to facilitate quick integration.

## Directory overview

File/Directory	Description
base	General APIs
component	Components used on the UI

File/Directory	Description
module	Encapsulated modules such as clipping, editing, special effects, composition, shooting, multi-screen shooting, upload, and video selection to be used by the corresponding views
utils	The tool module
PermissionIntroductionDialog	The permission pop-up window
UGCKit	The UGSV initialization class
UGCKitConstants	Constant configuration information
UGCKitPicturePicker	The view of the image selection module
UGCKitVideoCut	The view of the short video clipping module
UGCKitVideoEdit	The view of the short video editing module
UGCKitVideoEffect	The view of the short video special effect setting module
UGCKitVideoJoin	The view of the short video composition module
UGCKitVideoMixRecord	The view of the multi-screen shooting module
UGCKitVideoPicker	The view of the video selection module
UGCKitVideoPublish	The view of the video upload module
UGCKitVideoRecord	The view of the short video recording module

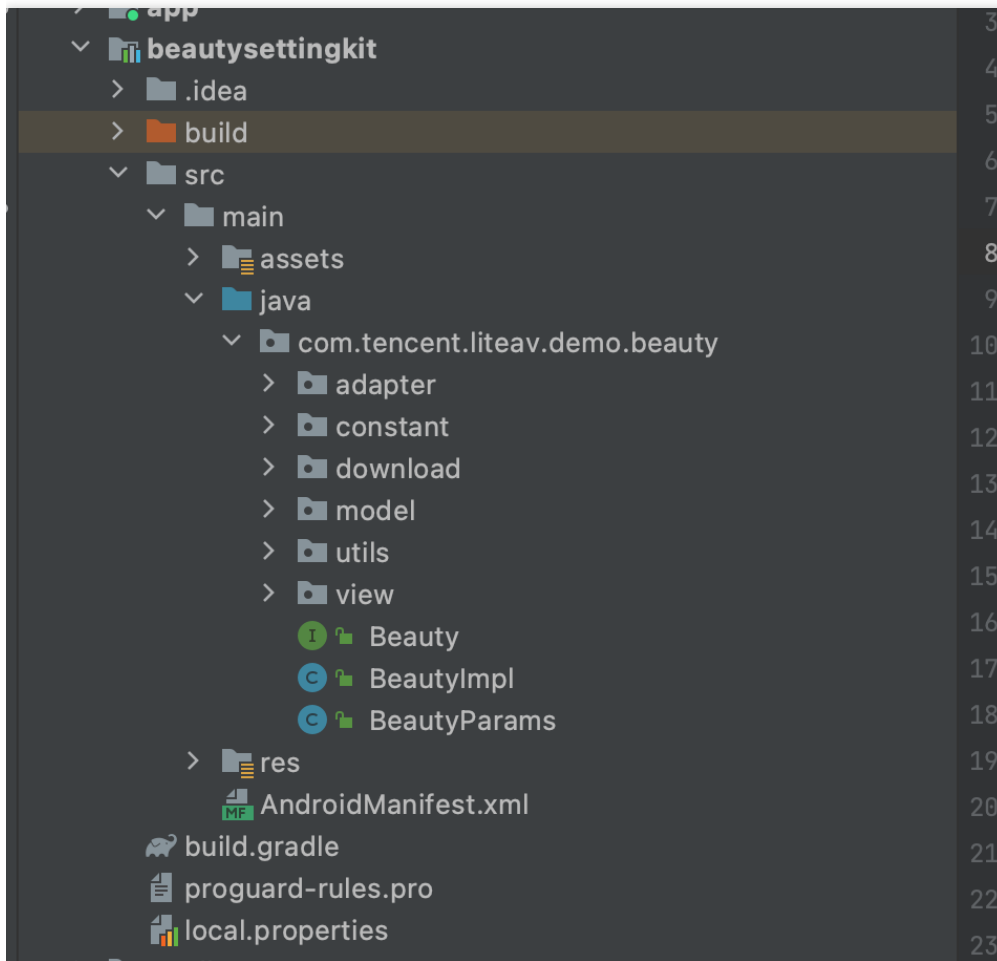
## xmagickit



This module encapsulates Tencent Effect so that it can quickly integrated.

File/Directory	Description
config	The configuration information for parsing animated effect resources downloaded dynamically in the demo project
download	The module for downloading animated effect resources
module	The class for creating and encapsulating Tencent Effect beauty filter attributes
panel	The beauty filter panel module
utils	The tool class
widget	The custom view
XMagicImpl	The basic encapsulation of beauty filter objects for easier use

## beautysettingkit



This module is the basic beauty filter module of UGSV. To use basic beauty filters, you can use this module for quick integration.

File/Directory	Description
adapter	The UI adapter of the beauty filter panel
constant	The beauty filter constant definitions
download	The beauty filter download module
model	The encapsulated module of beauty filter attributes
utils	The tool class
view	The beauty filter panel
BeautyImpl	The encapsulation class for using beauty filters
BeautyParams	The encapsulation class of beauty filter parameters