

# 短视频 SDK 含 UI 集成方案



腾讯云

**【版权声明】**

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

**【服务声明】**

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

**【联系我们】**

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

## 文档目录

含 UI 集成方案

小视频 App

源码解读 (iOS)

源码解读 (Android)

微信小程序多端框架

# 含 UI 集成方案 小视频 App

最近更新时间：2024-02-01 21:46:21

## 终端部分

按照如下三步操作，可以用 XCode 或者 Android Studio 编译和调试小视频 App 的客户端代码，运行效果如下：



### 步骤1：下载 App 源码

单击 [小视频源码](#) 可以下载到小视频 App 的源代码。

### 步骤2：申请 SDK 用的 License

请参见 [申请 License](#)。

### 步骤3：准备调试环境

- iOS 平台
  - XCode 9 或更高版本
  - iOS12 或更高版本
- Android 平台
  - Android SDK Tools: android-sdk\_26.0.2
    - minSdkVersion: 21
    - targetSdkVersion: 26

### 步骤4：编译运行

单击 XCode 或 Android Studio 的 Build 按钮，即可完成编译和运行工作，源码中默认配置了腾讯云提供的测试服务器地址 <http://demo.vod2.myqcloud.com/lite/>，以便您快速在调试环境中运行我们的 App。

## 后台部分

小视频 App 的运行依赖两种后台服务：

### 点播云服务 (VOD)

点播 (VOD) 服务可以为小视频提供视频的存储和在线分发的能力, 您在购买短视频的基础版 License 时, 会一并购买腾讯云的点播服务套餐包, 因此包含一定量的流量可以使用。

### 业务服务器 (CVM)

小视频 App 需要一台简单的业务服务器, 该服务器可以为 App 提供注册、登录、视频列表存储、视频上传签名等能力, 您可以将其搭建在腾讯云 CVM 云服务器上, 并可以自行修改里面的逻辑。

#### 警告:

当前小视频源码包提供的默认服务器地址及云点播相关账号资源 (以下简称为测试资源) 皆为腾讯云所有, 且有并发限制, 仅用于帮助您快速完成调试体验, 在您使用时请注意:

- 1、上述测试资源有随时停用/更新的可能性, 以源码包内实际提供的内容为准。
- 2、上述云点播账号内的数据将不定期清除且不予恢复。
- 3、禁止将上述测试资源用于违规业务。
- 4、请勿直接将上述测试资源用于正式环境。如您需要正式使用, 请参考下述指引, 将方案中涉及的服务器、云点播相关账号等替换为您的专属资源, 保证您正式业务的安全与稳定。

如果您希望自己搭建 App 的后台服务器, 可以按照如下步骤自行搭建:

## 步骤1: 开通点播服务 (VOD)

1. 登录 [云点播控制台](#) 开通云点播服务, 云点播服务可以为小视频提供视频存储和在线播放的能力。
2. 在云点播控制台的应用管理 > 进入应用 > 回调设置中设置回调模式为可靠回调, 事件回调配置中选择上传完成回调, 该配置需要10分钟左右能生效。

## 步骤2: 获取云 API 密钥

小视频 App 在上传视频时, 需要使用腾讯云密钥, 即 SecretId 和 SecretKey, 这两个 Key 要从腾讯云控制台中获取并配置到业务服务器上。

1. 登录控制台, 选择云产品 > 访问管理 > [API 密钥管理](#), 进入“API 密钥管理”页面。
2. 获取云 API 密钥。如果您尚未创建密钥, 则单击新建密钥即可创建一对 SecretId 和 SecretKey。

| APPID | 密钥  | 创建...   | 最近访问时间     | 状态  | 操作 |
|-------|---|---------|------------|-----|----|
|       | SecretId: [REDACTED]<br>SecretKey: ***** <a href="#">显示</a> | 2020... | 2021-06-03 | 已启用 | 禁用 |
|       | SecretId: [REDACTED]<br>SecretKey: ***** <a href="#">显示</a> | 2021... | -          | 已启用 | 禁用 |

## 步骤3: 在云服务器上部署后台代码

1. 新建 CVM 云服务器。

多个关键字用竖线“|”分隔，多个过滤标签用回车键分隔

| ID/名称                    | 监控 | 状态  | 可用区  | 实例类型          | 实例配置  | 主IPv4地址 |
|--------------------------|----|-----|------|---------------|---|---------|
| <input type="checkbox"/> |    | 运行中 | 广州三区 | 标准网络优化型 SN3ne | 1核 2GB 18Mbps<br>系统盘: 高性能云硬盘<br>网络: Default-VPC |         |
| <input type="checkbox"/> |    | 运行中 | 广州三区 | 标准网络优化型 SN3ne | 1核 2GB 1Mbps<br>系统盘: 高性能云硬盘<br>网络: Default-VPC  |         |
| <input type="checkbox"/> |    | 已关机 | 广州四区 | 标准型S5         | 1核 2GB 1Mbps<br>系统盘: SSD云硬盘<br>网络: Default-VPC  |         |

2. 选择自定义配置，进入镜像市场选取镜像。

镜像:

系统盘:   GB [选购指引](#)  
 购买成功后，系统盘不支持更换介质

数据盘: [新建云硬盘数据盘](#) 还可增加20块数据盘

镜像市场

推荐镜像:

全部

运行环境:

操作系统: Ubuntu Server 16.04.1 LTS 64位 系统架构: 64位 0元 免费使用

集成软件: nginx1.10.3 MariaDb10.0.34 nodejs9.11.2

开发者工具

3. 配置硬盘和网络，以及云服务器访问密码，妥善保管好密码，然后设置安全组。

1.选择机型
2.设置主机
3.确认配置信息

所属项目  ?

安全组 新建安全组 已有安全组 ? [使用指引](#)

?

如您有业务需要放通其他端口，您可以 [新建安全组](#)

---

实例名称  支持批量连续命名或指定模式串命名，你还可以输入60个字符 ?

登录方式 设置密码 立即关联密钥 自动生成密码 ?

用户名

SSH密钥  ?

如您现有的密钥不合适，可以[现在创建](#)

安全加固  免费开通 ?

安装组件免费开通DDoS防护和云镜主机防护 [详细介绍](#)

云监控  免费开通 ?

免费开通云产品监控、分析和实施告警，安装组件获取主机监控指标 [详细介绍](#)

4. 登录生成的云服务器。

单击实例操作栏的登录，可以通过腾讯云的网页 shell 进行访问，也可以用 **putty** 或 **SecretCRT** 采用 ssh 登录到云服务器。

**登录Linux实例** ×

标准登录方式 推荐 [登录不上?](#)

支持复制粘贴、中文输入法，需要开启 SSH 端口 (TCP:22)

立即登录

其他方式 (VNC)

该方式暂不支持复制粘贴、中文输入法。

提示：采用VNC方式登录，请务必开启 MFA 二次验证提高安全保障级别

立即登录

更多方式: [登录Linux云服务器](#)

## 5. 修改云服务器配置信息。

将如下脚本中的 `appId`、`SecretId` 和 `SecretKey` 配置 [步骤2](#) 中获取到的 APPID、SecretId 和 SecretKey。然后登录云服务器，直接在云服务器上执行修改后的脚本。

5.1 请在本地修改以下配置并复制，然后登录云服务器在控制台粘贴回车执行。

```
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
```

5.2 在服务器输入启动服务命令直接启动服务，服务启动默认端口为：`8001`。

○ 启动服务：

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

○ 如需关闭或者重启服务，可以使用以下命令：

重启服务：

```
pm2 restart litvideo;
```

关闭服务：

```
pm2 stop litvideo;
```

5.3 查看 [步骤4](#) 中云服务器的公网 IP，在浏览器中输入 `http://IP` 查看服务是否启动成功。

## 步骤4：替换终端源代码中的后台地址

● **iOS**：源码包解压后在 `iOS/Demo/XiaoShiPin/TCConstants.h`，将文件里的 `kHttpServerAddr` 改成您的云服务器公网 IP 地址。

● **Android**：源码包解压后在

`XiaoShiPin_Professional_Android/Demo/ugckit/src/main/java/com/tencent/qcloud/ugckit/UGCKitConstants.java`，将文件里的 `APP_SVR_URL` 改成您的云服务器公网 IP 地址。

 **注意**

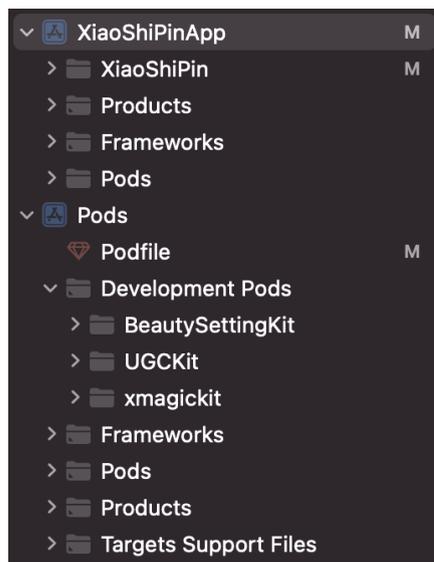
如果服务器没有配置证书，这里的云服务器地址需要用 HTTP，而不能用 HTTPS。

至此，小视频的服务器模式配置完成，您可以运行 App 体验小视频的各项功能。

# 源码解读 (iOS)

最近更新时间: 2024-04-01 15:09:51

## 工程结构



小视频主要集成了 UGCKit 作为核心功能库。集成方式参考 [UGCKit](#) (主要负责播放录制)。

小视频的美颜功能, 分为基础美颜和腾讯特效。基础美颜主要集成 BeautySettingkit 实现, 集成方式请参见 [类抖音特效](#) 文档, 相关代码都可以在 BeautySettingKit 目录下找到。腾讯特效主要集成xmagickit实现, 实现方式请参见 [短视频SDK集成腾讯特效](#) 文档, 相关代码都可以在xmagickit目录下找到。

## 模块介绍

小视频按照功能不同划分了7个模块, 分别为:

- 账号、列表管理、发布以及资料模块 (处于小视频目录下)。
- 播放、录制、编辑模块由 UGCKit 负责。

### 账号模块

- 账号模块负责处理用户登录/注册, 以及登录缓存的逻辑。
- 如果您已经有自己的账号体系, 可以直接替换该模块。
- 账号模块的 UI 逻辑分为 `XiaoShiPin/AppViewControllers/Account` 及 `XiaoShiPin/AppViewControllers/AccountInfo`, 前者涉及登录的 UI 模块, 后者涉及登录后的资料处理 UI 模块。
- 账号模块的业务逻辑可以在 `XiaoShiPin/Model` 下查看。

## 主界面和列表管理

### 模块简介

- 主界面主要负责点播列表、录制、个人资料三个一级功能的切换。
- 进入 App 后, 默认展示列表界面。单击录制按钮后, 如果未登录, 会先跳转到登录界面; 如果已经登录, 将跳转到推流的界面; 单击个人资料按钮, 将跳转到个人资料页面。
- 列表管理包含列表的拉取和展示。

### 相关代码

- Model:
  - `TCLiveListModel`(`XiaoShiPin/AppViewControllers/mainList`): 点播列表的数据层定义以及序列化/反序列化实现。
- UI:

- TCNavigationController: 定制的导航栏, 主要是设置导航栏的背景颜色。
- TCMainViewController: 主界面的 tab bar 控件, 用于切换列表、录制和个人资料页面。
- TCVideoListCell: 点播列表的 Cell 类, 主要展示封面、标题、昵称。
- TCLiveListViewController: 点播列表的 TableViewController, 负责展示点播列表, 单击后跳转播放界面。

## 播放模块

### 模块简介

播放模块主要包括: 短视频预加载、播放、缓存、分享等功能。

### 相关代码

`/XiaoShiPin/AppViewControllers/VideoPlayer` 目录下, 主要处理播放相关的业务逻辑, 包含 UI 和业务逻辑。

## 录制模块

### 模块简介

录制模块主要包括: 短视频多段录制、多段回删、多分辨率录制、变速录制等。

### 相关代码

`UGCKit/Source/Record` 在该目录下可以找到录制相关的所有逻辑。

## 编辑模块

### 模块简介

编辑模块主要包括: 短视频裁剪、BGM、滤镜风格、特效、动态贴纸、静态贴纸等。

### 相关代码

`UGCKit/Source/Edit` 目录下可以找到编辑相关的所有逻辑。

## 发布模块

### 模块简介

发布模块主要包括: 短视频发布。

### 相关代码

`XiaoShiPin/AppViewController/Publish` 目录下可以找到发布相关的所有逻辑。

## 资料模块

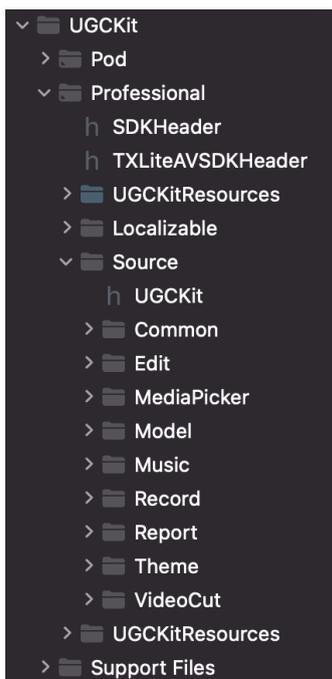
### 模块简介

- 资料模块主要负责用户资料的展示、存储和修改, 并负责将这些操作同步到服务器。
- 用户资料主要包括: 用户头像、昵称、性别等。
- 资料模块会从服务器同步用户最新资料到 App, 用户可以通过资料模块来浏览自己的相关资料, 包括用户头像、昵称和性别等。
- 用户可以通过资料模块修改自己的相关资料, 资料模块会将这些操作同步到服务器。
- 其他模块也可以通过资料模块获取, 修改用户资料。

### 相关代码

`XiaoShiPin/AppViewController/AccountInfo` 目录下可以找到用户资料相关的所有逻辑。

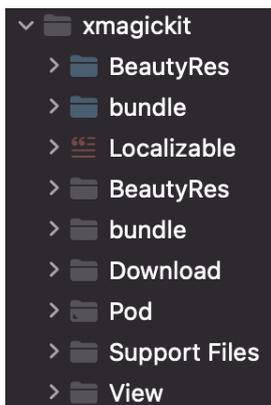
## ugckit



此模块主要是对短视频SDK的高级封装（包含了UI），方便快速接入。  
各个目录简介：

| 文件/目录              | 说明            |
|--------------------|---------------|
| Source/Common      | 短视频自定义的View模块 |
| Source/Edit        | 短视频编辑模块       |
| Source/MediaPicker | 短视频媒体选择模块     |
| Source/Model       | 短视频媒体模型模块     |
| Source/Music       | 短视频音乐模块       |
| Source/Record      | 短视频录制模块       |
| Source/Report      | 短视频数据上报模块     |
| Source/Theme       | 短视频资源主题模块     |
| Source/VideoCut    | 短视频裁剪模块       |

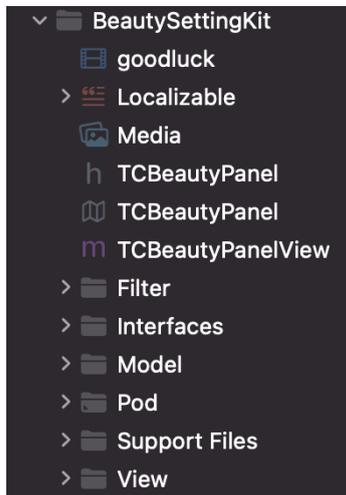
## xmagickit



此模块用于快速接入腾讯特效，主要是对腾讯特效的封装，方便快速接入。

| 文件/目录     | 说明      |
|-----------|---------|
| BeautyRes | 图片资源    |
| bundle    | 美颜素材    |
| Download  | 下载模块    |
| View      | 美颜面板和数据 |

## beautysettingkit



此模块是短视频中的基础美颜模块，如果客户选择使用基础美颜，则可以使用此模块进行快速集成。

| 文件/目录      | 说明     |
|------------|--------|
| Filter     | 滤镜模块   |
| Interfaces | 美颜接口   |
| Model      | 美颜数据模型 |
| View       | 美颜面板UI |

# 源码解读 (Android)

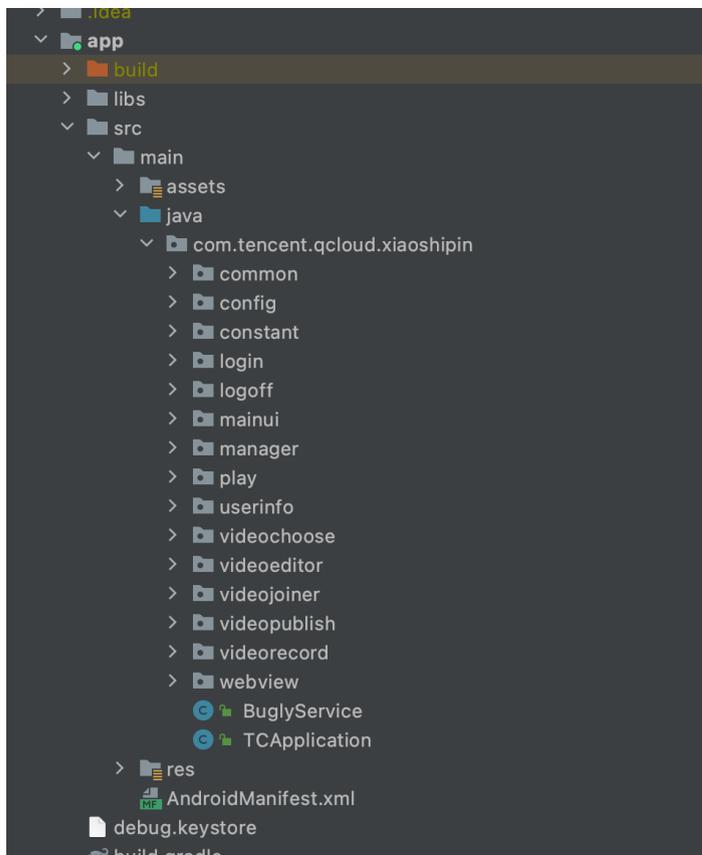
最近更新时间: 2025-06-04 11:02:42

## 工程结构

工程共包含四个模块, [app](#)、[ugckit](#)、[xmagickit](#)、[beautysettingkit](#)。

## app

### 目录介绍



| 文件/目录       | 说明                            |
|-------------|-------------------------------|
| common      | 通用组件, 包括各种工具类、自定义界面(美颜、播放器控件) |
| login       | 账号模块, 包括登录以及注册                |
| logoff      | 账号注销页面                        |
| mainui      | 小视频主界面, 包括主 Activity 以及视频列表   |
| manager     | 权限处理模块                        |
| play        | 点播播放模块                        |
| userinfo    | 个人资料模块                        |
| videochoose | 短视频文件选择模块                     |
| videoeditor | 短视频编辑模块                       |
| videojoiner | 短视频合成模块                       |

|              |          |
|--------------|----------|
| videopublish | 短视频发布模块  |
| videorecord  | 短视频录制模块  |
| webview      | 用于承载H5页面 |

## 模块介绍

小视频按照功能划分成账号、主界面、播放、短视频（编辑、合成、录制、发布），代码上也是按照这种划分进行分类，下面我们将分别介绍这些模块以及相应实现。

### 账号模块

- 账号模块负责处理用户登录/注册以及登录缓存的逻辑。
- 如果您已经有自己的账号体系，可以直接替换该模块。
- 账号模块通过调用 TCUserMgr 的 register 将用户名，密码注册到小视频的业务后台，调用 TCUserMgr 的 login 方法进行登录，并将登录信息缓存到本地 SharedPreferences 中，退出登录，清空本地缓存。

### 相关代码：

| 类名                      | 描述         |
|-------------------------|------------|
| TCAApplication.java     | sdk 初始化类   |
| TCLoginActivity.java    | 用户登录页面     |
| TCRegisterActivity.java | 用户注册页面     |
| TCUserMgr               | 用户登录注册管理类  |
| ProgressFragmentUtil    | 显示进度条的控件   |
| TCUserInfoFragment.java | 用户资料展示页面   |
| TCAboutActivity         | 小视频介绍的关于页面 |

## 主界面和列表管理

- 主界面主要负责短视频列表、短视频拍摄/编辑和个人资料三个一级功能的切换。
- 登录成功后，默认展示列表界面；单击+号按钮后，将弹出对话框让您选择录制小视频或编辑本地小视频；单击个人资料按钮，将跳转到个人资料页面。
- 列表管理包含列表的拉取和展示。

### 相关代码：

| 类名                         | 描述                           |
|----------------------------|------------------------------|
| TCSplashActivity.java      | 闪屏界面                         |
| TCMainActivity.java        | 主界面，用于呈现短视频列表，短视频拍摄/编辑，用户信息页 |
| TCLiveListMgr.java         | 列表管理类，提供接口获取本地内存列表及从服务器更新列表  |
| TCLiveListFragment.java    | 列表展示界面，负责呈现短视频的数据            |
| TCLiveListAdapter.java     | 短视频列表适配层                     |
| TCUGCVideoListAdapter.java | 短视频列表适配层                     |
| TCVideoInfo                | 视频数据                         |

## 短视频录制模块

小视频提供录制一分钟的 [短视频](#) 功能，但 SDK 本身不显示录制时长。

相关代码：

| 类名                         | 描述           |
|----------------------------|--------------|
| TCVideoRecordActivity.java | 短视频录制界面      |
| RecordProgressView         | 按住拍摄短视频 View |
| ComposeRecordBtn           | 短视频多段拍摄进度条   |

### 文件选择模块

提供本地文件选择功能，列出了手机中所有 mp4 视频文件。

相关代码：

| 类名                            | 描述                               |
|-------------------------------|----------------------------------|
| TCVideoChooseActivity.java    | 本地 mp4 文件选择界面                    |
| TCVideoEditorListAdapter.java | 本地 mp4 文件列表适配器                   |
| TCVideoEditorMgr.java         | mp4 视频文件管理类，提供接口获取存储在手机中的 mp4 文件 |
| TCVideoFileInfo.java          | 本地视频数据                           |

### 编辑模块

[视频编辑](#) 包括视频裁剪、慢动作、滤镜、音乐混音、贴纸、添加字幕等功能。

相关代码：

- [videoeditor/](#) 目录：

| 类名                             | 描述                                 |
|--------------------------------|------------------------------------|
| TCVideoPreprocessActivity.java | 录制后的视频进入编辑时预处理类的界面                 |
| TCVideoCutterActivity.java     | 短视频裁剪界面                            |
| TCVideoEditorActivity.java     | 短视频裁剪后编辑界面，底部有音乐，滤镜，速度，色调，贴纸，字幕等功能 |
| TCVideoEffectActivity.java     | 单击底部按钮，进入短视频添加特效界面                 |
| BaseEditFragment.java          | 特效 Fragment 的父类，用于控制多个界面特效的播放状态    |
| TCVideoJoinerActivity.java     | 当选择多个文件进行编辑时，先将多个视频文件合成一个视频界面      |

- [videoeditor/cutter/](#)目录： 视频裁剪相关。
- [videoeditor/time/](#) 目录： 时间特效相关，包括慢动作，重复，视频倒放。
- [videoeditor/bgm/](#) 目录： 背景音相关。
- [videoeditor/paster/](#) 目录： 贴纸相关，包括动态贴纸和静态贴纸。
- [videoeditor/motion/](#) 目录： 动态滤镜相关，包括四种动态滤镜（不支持自定义扩展，如需更多滤镜特效，请提交工单联系我们）。
- [videoditor/bubble/](#) 目录： 气泡字幕相关。
- [videoeditor/utils/](#) 目录： 短视频编辑工具相关。
- [videoeditor/common/](#)目录： 短视频编辑通用组件。
- [videojoiner/](#)目录： 短视频合成相关。

### 短视频发布模块

将录制的文件发布到腾讯云视频分发平台（点播系统）中。

相关代码：

| 类名                            | 描述      |
|-------------------------------|---------|
| TCVideoPublisherActivity.java | 短视频发布界面 |

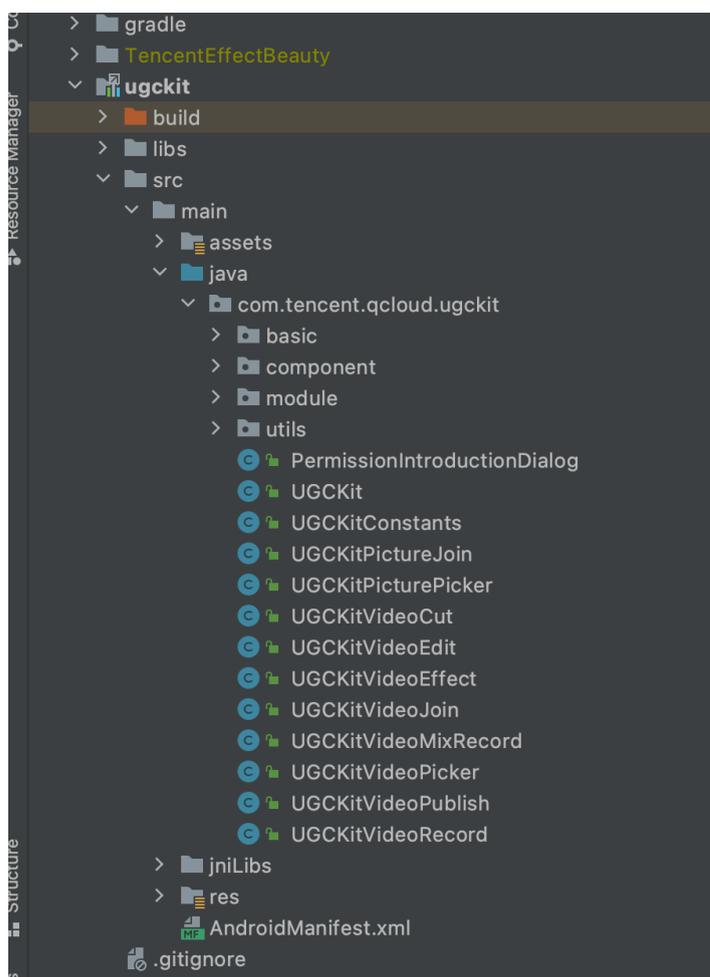
### 短视频播放

播放已经发布到点播系统中的视频，在小视频列表页面上下滑动可快速切换上/下一个视频。

相关代码：

| 类名                       | 描述      |
|--------------------------|---------|
| TCVodPlayerActivity.java | 短视频播放界面 |

### ugckit



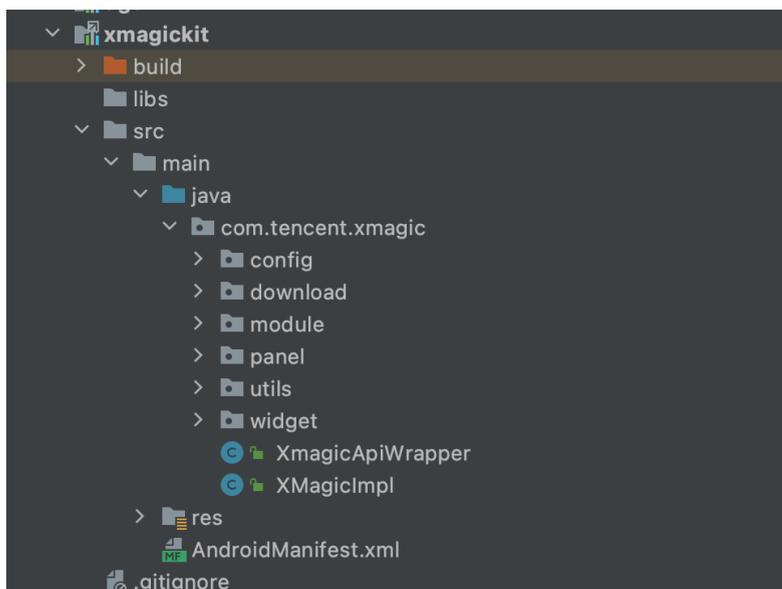
此模块主要是对短视频 SDK 的高级封装（包含了 UI），方便快速接入。

### 目录介绍

| 文件/目录     | 说明   |
|-----------|--|
| base      | 通用接口   |
| component | 页面上使用的组件                                     |
| module    | 裁剪、编辑、特效、合成、多屏合拍、录制、上传、视频选择 等模块封装，方便对应view使用 |
| utils     | 工具类模块  |

|                              |               |
|------------------------------|---------------|
| PermissionIntroductionDialog | 权限对话框         |
| UGCKit                       | ugc初始化类       |
| UGCKitConstants              | 常量配置信息        |
| UGCKitPicturePicker          | 图片选择模块View    |
| UGCKitVideoCut               | 短视频裁剪模块View   |
| UGCKitVideoEdit              | 短视频编辑模块View   |
| UGCKitVideoEffect            | 短视频特效设置模块View |
| UGCKitVideoJoin              | 短视频合成模块View   |
| UGCKitVideoMixRecord         | 多屏录制模块View    |
| UGCKitVideoPicker            | 视频选择模块View    |
| UGCKitVideoPublish           | 上传视频模块View    |
| UGCKitVideoRecord            | 短视频录制模块View   |

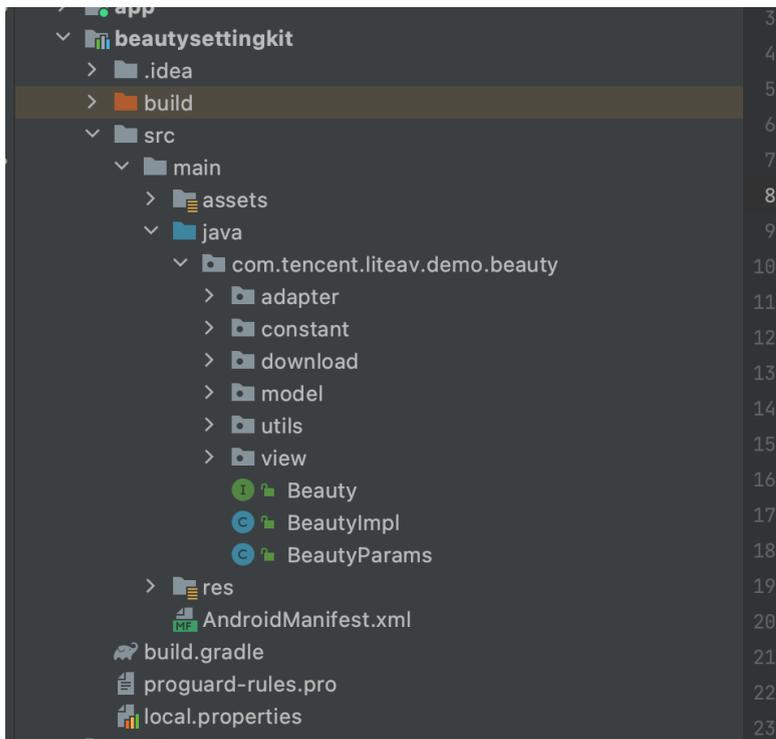
## xmagickit



此模块用于快速接入腾讯特效，主要是对腾讯特效的封装，方便快速接入。

| 文件/目录      | 说明                               |
|------------|----------------------------------|
| config     | demo工程中的动效资源是动态下载的，用于解析动效资源的配置信息 |
| download   | 动效资源的下载模块                        |
| module     | 腾讯特效美颜属性的创建和封装类                  |
| panel      | 美颜面板模块                           |
| utils      | 工具类                              |
| widget     | 自定义view                          |
| XMagicImpl | 对美颜对象的初级封装，方便使用                  |

## beautysettingkit



此模块是短视频中的基础美颜模块，如果客户选择使用基础美颜，则可以使用此模块进行快速集成。

| 文件/目录        | 说明             |
|--------------|----------------|
| adapter      | 美颜面板UI Adapter |
| constant     | 美颜常量定义         |
| download     | 美颜下载模块         |
| model        | 美颜属性封装模块       |
| utils        | 工具类            |
| view         | 美颜面板           |
| BeautyImpl   | 使用美颜的封装类       |
| BeautyParams | 美颜参数封装类        |

# 微信小程序多端框架

最近更新时间：2025-05-26 11:05:42

## 服务引入

## 环境配置

### 环境要求

多端框架 SDK 的版本要求：iOS SDK  $\geq$  1.4.18；Android SDK  $\geq$  1.5.2。

### 权限要求

- Android 端要求在适应 SDK 接口前申请好权限：`android.permission.WRITE_EXTERNAL_STORAGE`。
- iOS 端要求在适应 SDK 接口前申请好权限 `NSAppleMusicUsageDescription`。

## 插件配置

打开多端应用，找到 `project.miniapp.json` 配置文件，找到其中的多端插件配置选项。

**多端插件配置**

### Android 插件配置

- 是否启用此插件
- 插件 ID:
- 插件版本:
- 插件配置文件:  [浏览](#)

### iOS 插件配置

- 是否启用此插件
- 插件 ID:
- 插件版本:
- 是否从本地读取此插件(无需配置插件版本)
- 是否在 App 启动的时候就自动加载
- 将文件添加到原生插件中:  [浏览](#)

## Android 插件配置

在 Android 插件配置中，需要填写下面选项：

1. 是否启用此插件：勾选
2. 插件 ID：wx9cfd2cd2a9497bf6
3. 插件版本：1.0.0(示例值)
4. 插件配置文件：不填

## iOS 插件配置

在 iOS 插件配置中，需要填写下面选项：

1. 是否启用此插件：勾选
2. 插件 ID：wx9cfd2cd2a9497bf6
3. 插件版本：1.0.0(示例值)
4. 插件配置文件：不填

## 服务使用

### 集成 JS API

在 miniprogram 目录下执行如下命令安装 donut-ugc-editor：

```
npm i donut-ugc-editor
```

在微信开发者工具中构建 npm（路径：微信开发者工具 > 工具 > 构建 npm）。

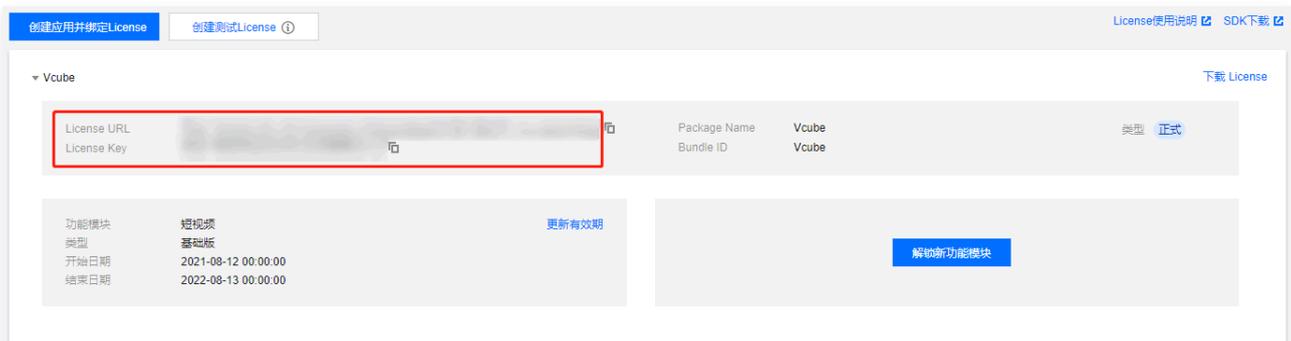
### 使用方法

#### 1. 引入 jsapi

```
import UGCEditor from "donut-ugc-editor"
```

#### 2. 设置 License

2.1 参见 [License 申请](#) 的指引申请 License 后，从 [云点播控制台](#) 复制 License Key 和 License URL，如下图所示：



2.2 在项目中调用 `UGCEditor#setUGCLicense` 方法来设置 License：

```
UGCEditor.setUGCLicense(  
  $licenseUrl,  
  $licenseKey,  
  (code,msg) => {  
    console.log("setUGCLicense listener => code:"+code+",msg:"+msg);  
  }  
);
```

调用 `UGCEditor#setUGCLicense` 方法来设置 License 信息，需要的参数为：

- `licenseUrl`：腾讯云申请到的 `licenseUrl` 信息。
- `licenseKey`：腾讯云申请到的 `licenseKey` 信息。
- `callback`：(code: number, msg: string) => void。License 设置的回调信息，code = 0 时为成功，否则失败。

### 3. 进入 UGC 编辑页

```
UGCEditor.openUGCEditor(  
    $wxFilePath,  
    $minDurationInMilliSec,  
    $maxDurationInMilliSec,  
    (resultJSON) => {  
        console.log("openUGCEditor success listener => resultJSON:" + JSON.stringify(resultJSON,  
null, 2));  
    },  
    (errorMsg) => {  
        console.log("openUGCEditor fail listener => errorMsg:" + errorMsg);  
    },  
    () => {  
    }  
});
```

调用 `UGCEditor.openUGCEditor` 方法来进入到 UGC 编辑页面，它所需要的参数如下：

- `wxFilePath`：要编辑的文件路径 (MP4格式)，示例：`wxfile://xxxxx.mp4`。
- `minDurationInMilliSec`：编辑后视频的最小时长，单位 ms。
- `maxDurationInMilliSec`：编辑的最大时长，单位 ms。
- `success`: (jsonStr: string) => void，编辑完成之后的回调，返回编辑后视频的信息，格式为 JSON，内容包括：

```
duration: 编辑后的视频时长，单位ms。  
size: 编辑后视频的文件大小，单位byte。  
tempFilePath: 编辑后视频路径，示例：wxfile://xxxxx。  
tempThumbPath: 编辑后视频封面图路径，示例：wxfile://xxxxx。
```

- `failed`: (errMsg: string) => void，编辑失败的错误信息。

### 运行效果示例

该方法执行之后，会调起 UGC 视频编辑页面，运行示例截图如下：

