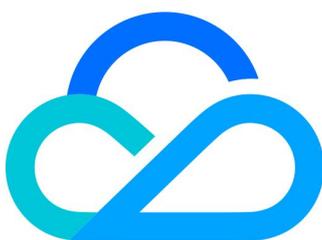


播放器 SDK

Flutter 端集成

产品文档



腾讯云

【 版权声明 】

©2013–2022 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内容。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。

文档目录

Flutter 端集成
超级播放器

Flutter 端集成 超级播放器

最近更新时间：2022-06-30 15:40:56

腾讯云视立方 Flutter 播放器是腾讯云开源的一款播放器组件，简单几行代码即可拥有类似腾讯视频强大的播放功能。包括横竖屏切换、清晰度选择、手势、小窗等基础功能，还支持视频缓存，软硬解切换，倍速播放等特殊功能。相比系统播放器，支持格式更多，兼容性更好，功能更强大。同时还支持直播流（FLV + RTMP）播放，具备首屏秒开、低延迟的优点，清晰度无缝切换、直播时移等高级能力。

本播放器是基于点播和直播播放的一个 Flutter 插件，同时支持 Android 和 iOS 两个平台。完全免费开源，不对播放地址来源做限制，可放心使用。

SDK 下载

腾讯云视立方 Flutter 播放器项目的地址是 [Player Flutter](#)。

项目简介

腾讯云视立方·播放器 SDK 是音视频终端 SDK（腾讯云视立方）的子产品 SDK 之一，基于腾讯云强大的后台能力与 AI 技术，提供视频点播和直播播放能力的强大播放载体。结合腾讯云点播或云直播使用，可以快速体验流畅稳定的播放性能。充分覆盖多类应用场景，满足客户多样需求，让客户轻松聚焦于业务发展本身，畅享极速高清播放新体验。

此项目提供了点播播放和直播播放，您可以基于播放器搭建自己的播放业务：

- **点播播放**：TXVodPlayerController 对 Android 和 iOS 两个平台的点播播放器 SDK 进行接口封装，你可以通过集成 TXVodPlayerController 进行点播播放业务开发。详细使用例子可以参考 DemoTXVodPlayer。
- **直播播放**：TXLivePlayerController 对 Android 和 iOS 两个平台的直播播放器 SDK 进行接口封装，你可以通过集成 TXLivePlayerController 进行直播播放业务开发。详细使用例子可以参考 DemoTXLivePlayer。

为了减少接入成本，在 example 里提供了播放器组件（带 UI 的播放器），基于播放器组件简单的几行代码就可以搭建视频播放业务。您可以根据自己项目的需求，把播放组件的相关代码应用到项目中去，根据需求进行调整 UI 和交互细节。

- **播放器组件**：SuperPlayerController 播放器组件，对点播和直播进行了二次封装，可以方便你快速简单集成。目前是 Beta 版本，功能还在完善中。详细使用例子可以参考 DemoSuperplayer。

快速集成

pubspec.yaml 配置

1. 集成 LiteAVSDK_Player 版本，默认情况下也是集成此版本。在pubspec.yaml中增加配置：

```
super_player:  
  git:  
    url: https://github.com/LiteAVSDK/Player\_Flutter  
    path: Flutter
```

2. 如果要集成 LiteAVSDK_Professional 版本，则pubspec.yaml中配置改为：

```
super_player:  
  git:  
    url: https://github.com/LiteAVSDK/Player\_Flutter  
    path: Flutter  
    ref: Professional
```

3. 更新依赖包：

```
flutter packages get
```

添加原生配置

Android 配置

在 Android 的 AndroidManifest.xml 中增加如下配置：

```
<!--网络权限-->  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />  
<!--点播播放器悬浮窗权限-->  
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />  
<!--存储-->  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

iOS 配置

1. 在 iOS 的Info.plist中增加如下配置:

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
```

2. iOS 原生采用pod方式进行依赖, 编辑podfile文件, 指定你的播放器 SDK 版本, 默认集成的是 Player 版 SDK。

```
pod 'TXLiteAVSDK_Player' //Player版
```

3. Professional 版 SDK 集成:

```
pod 'TXLiteAVSDK_Professional' //Professional版
```

如果不指定版本, 默认会安装最新的 TXLiteAVSDK_Player 最新版本。

集成过程中常见问题

- 执行flutter doctor命令检查运行环境, 知道出现” No issues found! “。
- 执行flutter pub get确保所有依赖的组件都已更新成功。

申请视频播放能力 License 和集成

若您已获得相关 License 授权, 需在 [腾讯云视立方控制台](#) 获取 License URL 和 License Key:



The screenshot shows the '正式 License' (Official License) page in the Tencent Cloud Video License Management Console. It displays the following information:

- Package Name:** yamii_coco
- Bundle ID:** [Redacted]
- 创建时间 (Creation Time):** 2022-05-20 17:11:51
- 基本信息 (Basic Information):**
 - License URL:** [Redacted] /v_cube.license
 - License Key:** [Redacted]
- 功能模块-短视频 (Short Video Module):**
 - 当前状态: 正常
 - 功能范围: 短视频制作基础版+视频播放
 - 有效期: 2022-05-20 00:00:00 到 2023-05-21 00:00:00
- 功能模块-直播 (Live Streaming Module):**
 - 当前状态: 正常
 - 功能范围: RTMP推流+RTC推流+视频播放
 - 有效期: 2022-05-20 15:23:35 到 2023-05-20 15:23:35
- 功能模块-视频播放 (Video Playback Module):**
 - 当前状态: 正常
 - 功能范围: 视频播放
 - 有效期: 2022-05-20 17:45:54 到 2023-05-21 00:00:00

A blue button labeled '解锁新功能模块' (Unlock New Function Modules) is visible at the bottom right of the console interface.

若您暂未获得 License 授权，需先参见 [视频播放License](#) 获取相关授权。

集成播放器前，需要 [注册腾讯云账户](#)，注册成功后申请视频播放能力 License，然后通过下面方式集成，建议在应用启动时进行。

如果没有集成 License，播放过程中可能会出现异常。

```
String licenceURL = ""; // 获取到的 licence url
String licenceKey = ""; // 获取到的 licence key
SuperPlayerPlugin.setGlobalLicense(licenceURL, licenceKey);
```

点播播放器使用

点播播放器核心类 TXVodPlayerController，详细 Demo 可参见 DemoTXVodPlayer。

```
import 'package:flutter/material.dart';
import 'package:super_player/super_player.dart';

class Test extends StatefulWidget {
  @override
  State<StatefulWidget> createState() => _TestState();
}

class _TestState extends State<Test> {
  late TXVodPlayerController _controller;

  double _aspectRatio = 16.0 / 9.0;
  String _url =
    "http://1400329073.vod2.myqcloud.com/d62d88a7vodtranscq1400329073/59c68fe75285890800381567412/adp.10.m3u8";

  @override
  void initState() {
    super.initState();
    String licenceUrl = ""; // 获取到的 licence url
    String licenseKey = ""; // 获取到的 licence key
    SuperPlayerPlugin.setGlobalLicense(licenceUrl, licenseKey);
    _controller = TXVodPlayerController();
    initPlayer();
  }
}
```

```
}

Future<void> initPlayer() async {
  await _controller.initialize();
  await _controller.startPlay(_url);
}

@override
Widget build(BuildContext context) {
  return Container(
    height: 220,
    color: Colors.black,
    child: AspectRatio(aspectRatio: _aspectRatio, child: TXPlayerVideo(controller: _controller)));
}
}
```

播放器组件使用

播放器组件核心类 SuperPlayerVideo，创建后即可播放视频。

```
import 'package:flutter/material.dart';
import 'package:super_player/super_player.dart';

/// flutter superplayer demo
class DemoSuperplayer extends StatefulWidget {
  @override
  State<StatefulWidget> createState() => _DemoSuperplayerState();
}

class _DemoSuperplayerState extends State<DemoSuperplayer> {
  List<SuperPlayerModel> videoModels = [];
  bool _isFullScreen = false;
  SuperPlayerController _controller;

  @override
  void initState() {
    super.initState();
  }
}
```

```

String licenceUrl = "填入您购买的 license 的 url";
String licenseKey = "填入您购买的 license 的 key";
SuperPlayerPlugin.setGlobalLicense(licenceUrl, licenseKey);
_controller = SuperPlayerController(context);
FTXVodPlayConfig config = FTXVodPlayConfig();
config.preferredResolution = 720 * 1280;
_controller.setPlayConfig(config);
_controller.onSimplePlayerEventBroadcast.listen((event) {
String evtName = event["event"];
if (evtName == SuperPlayerViewEvent.onStartFullScreenPlay) {
setState(() {
_isFullScreen = true;
});
} else if (evtName == SuperPlayerViewEvent.onStopFullScreenPlay) {
setState(() {
_isFullScreen = false;
});
} else {
print(evtName);
}
});
initData();
}

@override
Widget build(BuildContext context) {
return WillPopScope(
child: Container(
child: Scaffold(
backgroundColor: Colors.transparent,
appBar: _isFullScreen
? null
: AppBar(
backgroundColor: Colors.transparent,
title: const Text('SuperPlayer'),
),
body: SafeArea(
child: Builder(

```

```
builder: (context) => getBody(),
),
),
),
),
onWillPop: onWillPop);
}

Future<bool> onWillPop() async {
return !_controller.onBackPressed();
}

Widget getBody() {
return Column(
children: [_getPlayArea()],
);
}

Widget _getPlayArea() {
return SuperPlayerView(_controller);
}

Widget _getListArea() {
return Container(
margin: EdgeInsets.only(top: 10),
child: ListView.builder(
itemCount: videoModels.length,
itemBuilder: (context, i) => _buildVideoItem(videoModels[i]),
),
);
}

Widget _buildVideoItem(SuperPlayerModel playModel) {
return Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
ListTile(
leading: Image.network(playModel.coverUrl),
```

```
title: new Text(
  playModel.title,
  style: TextStyle(color: Colors.white),
),
onTap: () => playCurrentModel(playModel),
Divider()
],
);
}

void playCurrentModel(SuperPlayerModel model) {
  _controller.playWithModel(model);
}

void initData() async {
  SuperPlayerModel model = SuperPlayerModel();
  model.videoURL = "http://1500005830.vod2.myqcloud.com/6c9a5118vodcq1500005830/48d0f1f9387702299774251236/gZyqjgaZmnwA.m4v";
  model.playAction = SuperPlayerModel.PLAY_ACTION_AUTO_PLAY;
  model.title = "腾讯云音视频";
  _controller.playWithModel(model);
}

@override
void dispose() {
  // must invoke when page exit.
  _controller.releasePlayer();
  super.dispose();
}
}
```

深度定制开发指引

腾讯云播放器 SDK Flutter 插件对原生播放器能力进行了封装，如果您要进行深度定制开发，建议采用如下方法：

- 基于点播播放，接口类为 `TXVodPlayerController` 或直播播放，接口类为 `TXLivePlayerController`，进行定制开发，项目中提供了定制开发 Demo，可参考 `example` 工程里的 `DemoTXVodPlayer` 和

DemoTXLivePlayer。

- 播放器组件 SuperPlayerController 对点播和直播进行了封装，同时提供了简单的 UI 交互，由于此部分代码在 example 目录。如果您有对播放器组件定制化的需求，您可以进行如下操作：

把播放器组件相关的代码，代码目录：example/lib/superplayer，复制到您的项目中，进行定制化开发。

更多功能

完整功能可扫码下载视频云工具包体验，或直接运行工程 Demo。

