

直播 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或 95716。

文档目录

含 UI 集成方案

TUIKit 概述

直播推流组件

集成 TUIPusher (Android)

集成 TUIPusher (iOS)

直播播放组件

集成 TUIPlayer (Android)

集成 TUIPlayer (iOS)

集成 TCPlayer (Web)

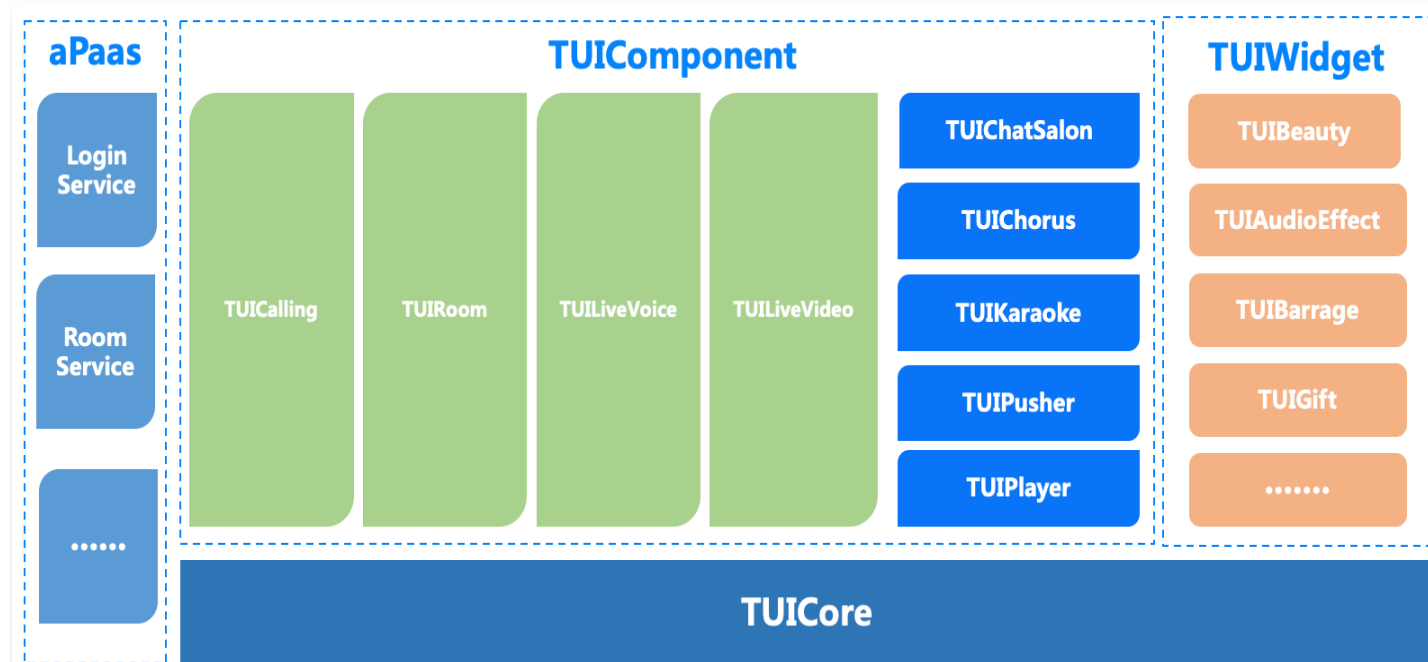
含 UI 集成方案

TUIKit 概述

最近更新时间：2025-08-08 17:16:21

TUIkit 是腾讯云音视频团队在5000+客户的服务积累中，结合业内主流的音视频场景，提炼出的开源解决方案，包含视频通话组件、直播组件、视频房间组件等多个客户端音视频组件，可以帮助开发者快速搭建诸如通话、客服、直播、语聊、教育等场景解决方案。

TUIKit 全家桶



如上图所示，TUIKit 分为 TUIComponent 和 TUIWidget 两种，同时支持可选的基础后台服务。

```
├─ TUIComponent
│   ├── TUIPusher          // 含完整UI的推流组件，支持多种协议推流，PK等功能，支持挂
│   载音效、弹幕等挂件；
│   ├── TUIPlayer          // 含完整UI的播放组件，支持多种协议播放，连麦主播等功能，
│   支持挂载礼物、弹幕等挂件；
│   ├── TUICalling         // 通话组件（类微信通话），针对视频通话、客服、金融审核等音
│   视频场景；
│   ├── TUIRoom            // 多人视频房间组件，针对教育、会议、面试等音视频场景，具备
│   禁言禁画等功能；
│   ├── TUILiveVoice       // 语聊直播组件，针对交友语聊、排队语聊，音乐房等音频场景；
│   └── TUILiveVideo        // 视频互动直播组件，具备连麦、PK、音效等功能；
```



```
|   └─ TUIChatSalon    // 语音沙龙组件，针对商务圆桌，兴趣论坛等音视频场景；
|   └─ TUICHorus       // 创新组件，针对合唱的音视频新玩法的音视频组件，可以搭配
腾讯云正版曲库解锁更多玩法；
|   └─ TUIKaraoke      // 创新组件，针对在线KTV等新玩法的音视频的组件，可以搭配
腾讯云正版曲库解锁更多玩法；
└─ TUIWidget
|   └─ TUIBeauty       // 美颜挂件
|   └─ TUIAudioEffect  // 音效挂件
|   └─ TUIBarrage      // 弹幕挂件
|   └─ TUIGift         // 礼物挂件
└─ aPass （可选）
|   └─ Room Service    // 提供房间列表、房间热度、主播列表等功能；
```

交流&反馈

当然，一样的环境，一样的步骤，可能也会有不一样的报错，这是工程师逃不开的魔咒，您可以通过如下方式来尝试解决

- TUIKit 技术交流 QQ 群：592465424



- [更多联系](#)

直播推流组件

集成 TUIPusher (Android)

最近更新时间：2025-08-12 09:25:01

组件介绍

`TUIPusher` 组件是一套开源的、完整的视频直播互动推流组件，它基于腾讯云 [直播 Live SDK](#) 和 [即时通信 IM SDK](#)，实现直播推流，直播 PK 等功能，同时支持弹幕、点赞、美颜等外挂插件，使用 `TUIPusher` 组件您可以

快速搭建诸如秀场直播、电商直播等场景化解决方案。



组件集成

步骤一：下载并导入 TUIPusher 组件

1. 单击进入 [Github](#)，选择克隆或者下载小直播工程代码。
2. 拷贝 Android/tuipusher、Android/tuiaudioeffect、Android/tuibeauty、Android/tuibarrage、Android/tuigift 等文件夹到您的工程中。
3. 在 `setting.gradle` 中完成导入，参考如下：

```
include ':tuipusher'
include ':tuiaudioeffect'
include ':tuibeauty'
include ':tuibarrage'
include ':tuigift'
```

4. 在 `app` 的 `build.gradle` 文件中添加对 `tuipusher` 等 module 的依赖：

```
api project(':tuipusher')
api project(':tuiaudioeffect')
api project(':tuibeauty')
api project(':tuibarrage')
api project(':tuigift')
```

步骤二：配置权限及混淆规则

1. 在 `AndroidManifest.xml` 中配置 `App` 的权限，`SDK` 需要以下权限（6.0以上的 Android 系统需要动态申请相机、麦克风权限等）：

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
```

2. 在 `proguard-rules.pro` 文件，将 `SDK` 相关类加入不混淆名单：

```
-keep class com.tencent.** { *; }
```

步骤三：初始化&创建组件

如果您未开通腾讯云直播相关服务，请先按照如下步骤开通相关服务：

- [开通云直播服务](#)，并在 [域名管理](#) 页面中配置推流域名和拉流域名。
- [创建应用并绑定 License](#)，并记录下 `LICENSEURL`、`LICENSEURLKEY` 等关键信息。

```
// 1. 初始化直播服务
V2TXLivePremier.setLicence(this, "您的LICENSEURL", "您的
LICENSEURLKEY");
```

```
// 2. 创建TUIPusher组件
TUIPusherView mTUIPusherView = new TUIPusherView(Context);

// 3. 设置诸如推流开始、推流结束等事件监听
mTUIPusherView.setTUIPusherViewListener(new TUIPusherViewListener() {
    ...
})
```

步骤四：有互动直播推流

1. 服务开通

因为连麦& PK 时需要更低的延时需求，需要在腾讯云直播控制台开通对应的连麦应用服务，如果您未开通，请登录云直播管理控制台选择 [连麦应用](#)，单击**新建连麦应用**输入应用名称（例如 `TUIPusher`），然后在该应用的对应操作栏中，选择**应用信息**进入应用管理页，查看并记录应用的 **SDKAppID** 和 **SECRETKEY（密钥）**。



⚠ 注意

因为在连麦/PK过程中，观众端还是需要正常观看CDN流，所以需要进入 **CDN 观看配置**页，开启旁路推流，推荐全局自动旁路。

← 连麦测试应用()

应用信息

CDN 观看配置

混流配置

录制配置

i 基于 UDP 传输协议的 TRTC 服务，通过协议转换将音视频流对接到云直播系统，这个过程称之为“旁路推流”。旁路推流把直播音视频流推送至云端后，即可通过 CDN 方式拉流观看。所产生的相关费用，请参见[云直播>流量带宽计费](#) 说明。

CDN 配置

旁路推流开关 ☐ 请前往 [trtc 控制台 - 应用管理](#) 开启旁路推流，关闭旁路推流将无法使用连麦服务

2. 组件登录

因为 PK 服务，需要直播间相互通信，所以需要进行单独登录，登录流程如下：

```
TUILogin.init(this, "您的SDKAppId", config, new V2TIMSDKListener() {  
    @Override  
    public void onKickedOffline() { // 登录被踢下线通知（示例：账号在  
其他设备登录）  
    }  
    @Override  
    public void onUserSigExpired() { // userSig过期通知  
    }  
});  
TUILogin.login("您的userId", "您的userSig", new V2TIMCallback() {  
    @Override  
    public void onError(int code, String msg) {  
        Log.d(TAG, "code: " + code + " msg:" + msg);  
    }  
    @Override  
    public void onSuccess() {  
    }  
});
```

登录组件参数说明：

- SDKAppId：即服务开通中记录到的 SDKAppId 信息。
- SECRETKEY：即服务开通中记录到的 SECRETKEY（密钥）。

- **userId**: 当前用户的 ID，字符串类型，长度不超过32字节，不支持使用特殊字符，建议使用英文或数字，可结合业务实际账号体系自行设置。
- **userSig**: 根据 SDKAppId、userId，SECRETKEY 等信息计算得到的安全保护签名，您可以单击 [这里](#) 直接在线生成一个调试的 UserSig，也可以参照我们的 [示例工程](#) 自行计算，更多信息请参见 [如何计算及使用 UserSig](#)。

3. 开始推流

基于 `RTC` 协议的推流 `URL` 的生成，可以参考 [示例工程](#) 中封装好的 `Utils` 方法，基本示例如下，具体参数信息请参见 [推拉流 URL](#)。

```
mTUIPusherView.start(String url);
```

4. 停止推流

```
mTUIPusherView.stop();
```

5. 发起 PK 请求

调用 `mTUIPusherView.sendPKRequest()` 后会向接收方发起 PK 请求，请求超时发送方设置的 `mTUIPusherView.setTUIPusherViewListener` 回调中，`onPKTimeout` 回调会收到超时通知。

```
mTUIPusherView.sendPKRequest(String userId);

public void onPKTimeout(TUIPusherView pusherView) {
    Toast.makeText(mContext, "PK request timed out",
        Toast.LENGTH_SHORT).show();
}
```

6. 接受 PK 请求

接收方设置的 `mTUIPusherView.setTUIPusherViewListener` 回调中，`onReceivePKRequest` 回调会通知接收方收到 PK 请求，可在此回调中处理 PK 请求。

```
public void onReceivePKRequest(TUIPusherView pushView, String userId,
    ResponseCallback callback) {
    showPKDialog(userId, callback);
}

private void showPKDialog(String userId, final
    TUIPusherViewListener.ResponseCallback callback) {
    if (mPKDialog == null) {
        final TextView textView = new TextView(this);
        textView.setText(userId + "invited linkMic");
    }
}
```



```
mPKDialog = new AlertDialog.Builder(this, R.style.PKDialogTheme)
    .setView(textView)
    .setPositiveButton("confirm", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which)
{
        callback.response(true);
    }
})
    .setNegativeButton("cace1", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which)
{
        callback.response(false);
    }
}).create();

mPKDialog.show();
}
```

步骤五：无互动直播推流

如果您的应用中无连麦或 **PK** 等互动场景，可以选择标准的 **RTMP** 协议进行推流，具体步骤如下：

- 开始推流

基于 RTMP 推流 URL 的生成，可以参考 [示例工程](#) 中封装好的 Utils 方法，基本规则如下图，具体参数信息请参见 [推流 URL](#)。

```
mTUIPusherView.start(String url);
```

- 停止推流

```
mTUIPusherView.stop();
```

交流&反馈

更多帮助信息，欢迎加入 QQ 群：**592465424**，进行技术交流和反馈。

集成 TUIPusher (iOS)

最近更新时间：2025-08-14 09:43:41

组件介绍

TUIPusher 组件是一套开源的、完整的视频直播互动推流组件，它基于腾讯云 [直播 Live SDK](#) 和 [即时通信 IM SDK](#)，实现直播推流，直播 PK 等功能，同时支持弹幕、点赞、美颜等外挂插件，使用 TUIPusher 组件您可以快速搭建诸如秀场直播、电商直播等场景化解决方案。



组件集成

步骤一：下载并导入 TUIPusher 组件

1. 单击进入 [Github](#)，选择克隆/下载代码。将 **iOS 目录** 下的 `TUIPusher`、`TUIGift`、`TUIBeauty`、`TUIBarrage` 和 `TUIAudioEffect` 文件夹拷贝到您在自己工程创建的 `Podfile` 文件同一级目录下。

2. 通过 **cocoapods** 导入组件：

在您的 `Podfile` 文件中添加以下依赖。之后执行 `pod install` 命令，完成导入。

```
# :path => "指向TUIPusher.podspec所在目录的相对路径"
pod 'TUIPusher', :path => "TUIPusher/"

# :path => "指向TUIAudioEffect.podspec所在目录的相对路径"
pod 'TUIAudioEffect', :path => "TUIAudioEffect/"

# :path => "指向TUIBarrage.podspec所在目录的相对路径"
pod 'TUIBarrage', :path => "TUIBarrage/"

# :path => "指向TUIBeauty.podspec所在目录的相对路径"
pod 'TUIBeauty', :path => "TUIBeauty/"

# :path => "指向TUIGift.podspec所在目录的相对路径"
pod 'TUIGift', :path => "TUIGift/"
```

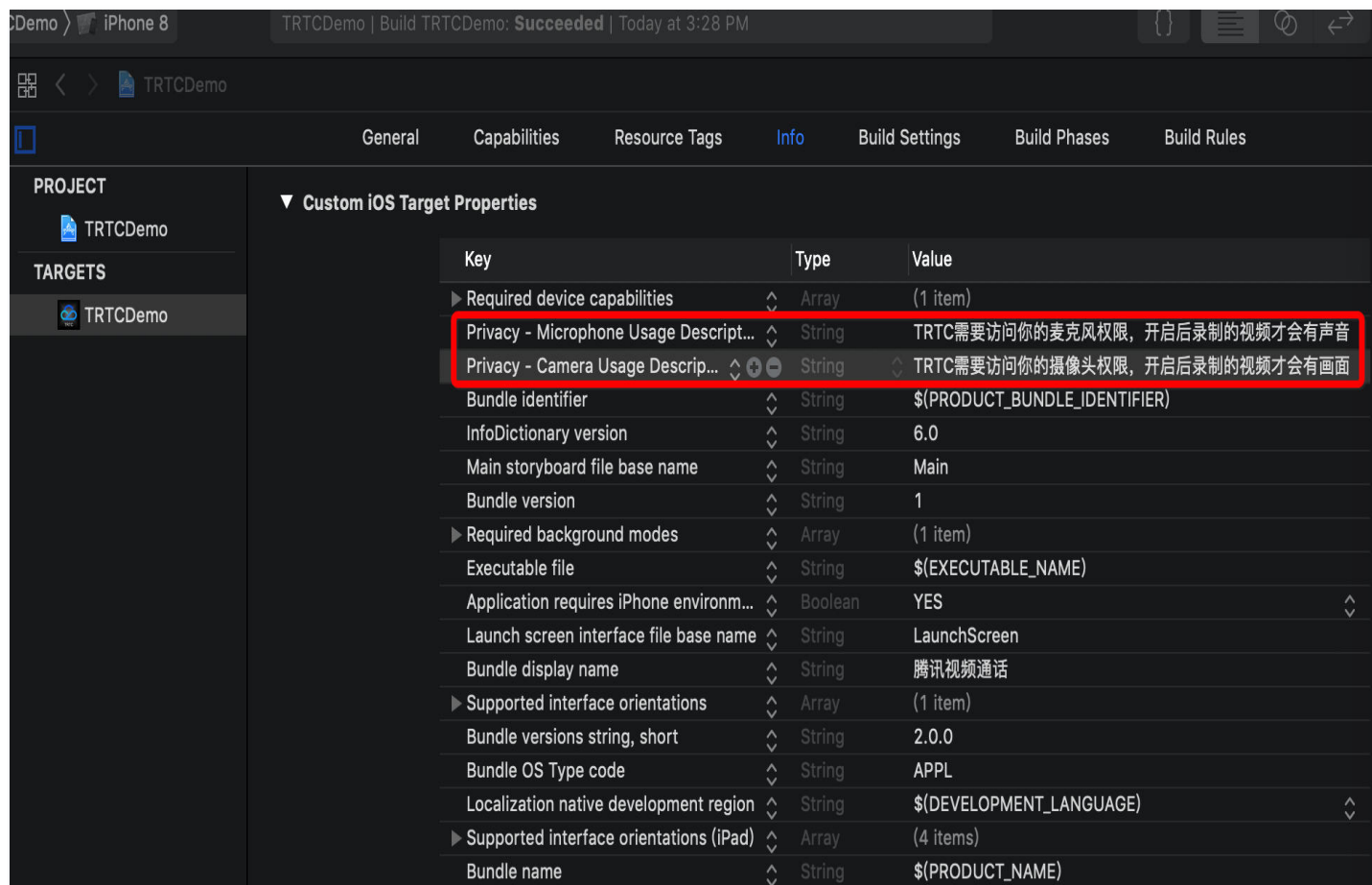
⚠ 注意

关于 `TUIBeauty` 的美颜特效依赖库编译失败问题，请前往 [美颜特效-SDK集成指引iOS](#)，下载并解压 **Demo 包**，将 SDK 目录中的 `libpag.framework`、`XMagic.framework`、`YTCommonXMagic.framework` 导入到 `TUIBeauty` 下的 `Frameworks` 目录，重新执行命令 "`pod install`"即可。

步骤二：配置权限

使用音视频功能，需要授权麦克风和摄像头的使用权限。在 App 的 `Info.plist` 中添加以下两项，分别对应麦克风和摄像头在系统弹出授权对话框时的提示信息。

```
<key>NSCameraUsageDescription</key>
<string>TUIPusherApp需要访问您的相机权限，开启后录制的视频才会有画面</string>
<key>NSMicrophoneUsageDescription</key>
<string>TUIPusherApp需要访问您的麦克风权限，开启后录制的视频才会有声音</string>
```



步骤三：初始化&创建组件

如果您未开通腾讯云直播相关服务，请先按照如下步骤开通相关服务：

- [开通云直播服务](#)，并在 [域名管理](#) 页面中配置推流域名和拉流域名。
- [创建应用并绑定License](#)，并记录下 `LICENSEURL`、`LICENSEURLKEY` 等关键信息。

```
// 1. 初始化直播服务
V2TXLivePremier.setLicence(LICENSEURL, key: LICENSEURLKEY)

// 2. 创建TUIPusher组件
let mTUIPusherView = TUIPusherView(frame: view.bounds);
view.addSubview(mTUIPusherView)

// 3. 为 TUIPusherView 设置事件回调 TUIPusherViewDelegate
mTUIPusherView.setDelegate(self)
```

步骤四：有互动直播推流

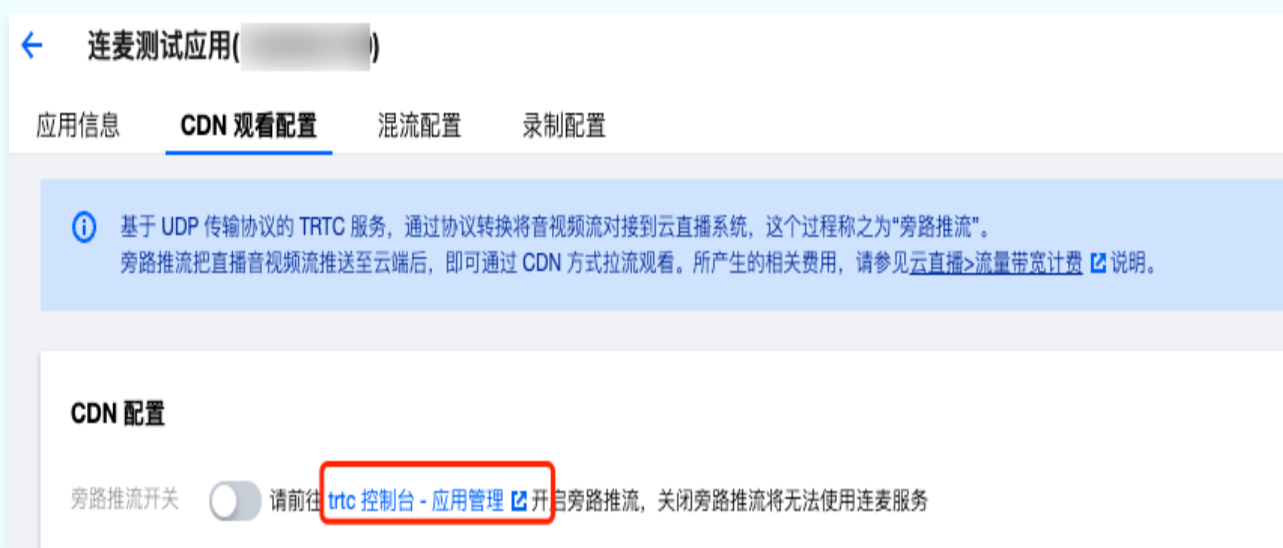
1. 服务开通

因为连麦& PK 时需要更低的延时需求，需要在腾讯云直播控制台开通对应的连麦应用服务，如果您未开通，请登录云直播管理控制台选择 [连麦应用](#)，单击新建连麦应用输入应用名称（例如 TUIPusher），然后在该应用的对应操作栏中，选择应用信息进入应用管理页，查看并记录应用的 SDKAppID 和 SECRETKEY（密钥）。



⚠ 注意

因为在连麦/PK 过程中，观众端还是需要正常观看 CDN 流，所以需要进入 **CDN 观看配置** 页，开启旁路推流，推荐全局自动旁路。



2. 组件登录

因为 PK 服务，需要直播间相互通信，所以需要进行单独登录，登录流程如下：

```
// 组件登录
TUILogin.initWithSdkAppID(SDKAPPID)
TUILogin.login(userId, userSig: userSig) {
    print("login success")
} fail: { code, message in
    print("login failed, code:\(code), error: \(message ?? "nil")")
}
```

○ 登录组件参数说明：

- **SDKAppID**：服务开通中记录到的 SDKAppID 信息。
- **SECRETKEY**：服务开通中记录到的 SECRETKEY（密钥）。
- **userId**：当前用户的 ID，字符串类型，长度不超过32字节，不支持使用特殊字符，建议使用英文或数字，可结合业务实际账号体系自行设置。
- **userSig**：根据 SDKAppId、userId，Secretkey 等信息计算得到的安全保护签名，您可以单击[这里](#)直接在线生成一个调试的 UserSig，也可以参照我们的[示例工程](#)自行计算，更多信息请参见[如何计算及使用 UserSig](#)。

3. 开始推流

```
mTUIPusherView.start(url: "xxxx");
```

基于 RTC 协议的推流 URL 的生成，可以参见我们的[示例工程](#)中封装好的 Utils 方法，基本示例如下，具体参数信息详情请参见[推拉流 URL](#)。

4. 停止推流

```
mTUIPusherView.stop();
```

5. 发起 PK 请求

调用 `mTUIPusherView.sendPKRequest()` 后会向接收方发起 Pk 请求，请求超时 `TUIPusherViewDelegate` 会收到 `onPKTimeout` 回调。

```
mTUIPusherView.sendPKRequest(userID: "xxxx");
```

6. 接受 PK 请求

接收方设置的 `mTUIPusherView.setDelegate` 回调中，`TUIPusherViewDelegate` 回调会通知接收方收到 PK 请求，可在此回调中处理 PK 请求。


```
//接收方收到PK请求回调
func onReceivePKRequest(_ pusherView: TUIPusherView, userId: String,
responseCallback completion: @escaping Response) {
    let alert = UIAlertController(title: "收到 PK 邀请, 是否接受?",
message: "", preferredStyle: .alert)
    let accept = UIAlertAction(title: "接受", style: .default) { _ in
        completion(true)
    }
    let reject = UIAlertAction(title: "拒绝", style: .cancel) { _ in
        completion(false)
    }
    alert.addAction(accept)
    alert.addAction(reject)
    present(alert, animated: true, completion: nil)
}
```

步骤五：无互动直播推流

如果您的应用中无连麦或 `PK` 等互动场景，可以选择标准的 `RTMP` 协议进行推流，具体步骤如下：

- 开始推流

基于 RTMP 推流 URL 的生成，可以参见我们的 [示例工程](#) 中封装好的 Utils 方法，基本规则如下图，具体参数信息请参见 [推拉流 URL](#)。

```
mTUIPusherView.start(url: "xxxx");
```

- 停止推流

```
mTUIPusherView.stop();
```

交流&反馈

更多帮助信息，欢迎加入 QQ 群：**592465424**，进行技术交流和反馈。

直播播放组件

集成 TUIPlayer (Android)

最近更新时间：2025-08-14 09:43:41

组件介绍

`TUIPlayer` 组件是一套开源的、完整的视频直播互动播放组件，它基于腾讯云 [直播 Live SDK](#) 和 [即时通信 IM SDK](#)，实现直播播放，直播连麦等功能，同时支持弹幕、点赞、美颜等外挂插件，通过 `TUIPlayer` 组件您可以轻松实现直播视频拉流，快速搭建诸如秀场直播、电商直播等场景化解决方案。



组件集成

步骤一：下载并导入 TUIPlayer 组件

1. 单击进入 [Github](#)，选择克隆或者下载小直播工程代码。
2. 拷贝 Android/tuoplayer、Android/tuibarrage、Android/tuigift 等文件夹到您的工程中。
3. 在 `setting.gradle` 中完成导入，参考如下：

```
include ':tuoplayer'
include ':tuibarrage'
```



```
include ':tuigift'
```

4. 在 `app` 的 `build.gradle` 文件中添加对 `tuoplayer` 等 module 的依赖：

```
api project(':tuoplayer')
api project(':tuibarrage')
api project(':tuigift')
```

步骤二：配置权限及混淆规则

1. 在 `AndroidManifest.xml` 中配置 App 的权限，SDK 需要以下权限（6.0以上的 Android 系统需要动态申请相机、麦克风权限等）：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
```

2. 在 `proguard-rules.pro` 文件，将 SDK 相关类加入不混淆名单：

```
-keep class com.tencent.** { *; }
```

步骤三：初始化&创建组件

```
// 1. 创建TUIPlayer组件
TUIPlayerView mTUIPlayerView = new TUIPlayerView(Context);

// 2. 设置诸如拉流开始、拉流结束等事件监听
mTUIPlayerView.setTUIPlayerViewListener(new TUIPlayerViewListener() {
    ...
})
```

步骤四：开始&停止播放

● 开始播放

```
mTUIPlayerView.startPlay(String url);
```

腾讯云直播支持两种播放协议：WebRTC 协议（即快直播，更低延时，更强抗性）、RTMP 协议、此处参数 URL 的生成，您可以参考 [示例工程](#) 中封装好的 Utils 方法，更多参数信息请参见 [推拉流 URL](#)。

● 停止播放

```
mTUIPlayerView.stopPlay();
```

步骤五：实现连麦功能（可选）

1. 开通云直播服务：

如果您未开通腾讯云直播相关服务，请先按照如下步骤开通相关服务并完成SDK鉴权：

- [开通云直播服务](#)
- [创建应用并绑定 License](#)，并记录下 `LICENSEURL`、`LICENSEURLKEY` 等关键信息。
- 按照如下方式，完成 SDK 推流鉴权：

```
V2TXLivePremier.setLicence(this, "您的LICENSEURL", "您的  
LICENSEURLKEY");
```

2. 开通连麦服务

因为连麦时需要更低的延时需求，需要在腾讯云直播控制台开通对应的连麦应用服务，如果您未开通，请登录云直播管理控制台选择 [连麦应用](#)，单击 [新建连麦应用](#) 输入应用名称（例如 `TUIPlayer`），然后在该应用的对应操作栏中，选择 [应用信息](#) 进入应用管理页，查看并记录应用的 `SDKAppID` 和 `SECRETKEY`（密钥）。



3. 组件登录

因为连麦服务，需要观众与主播相互通信，所以需要进行单独登录，登录流程如下：

```
TUILogin.init(this, "您的SDKAppId", config, new V2TIMSDKListener() {  
    @Override
```

```
public void onKickedOffline() { // 登录被踢下线通知（示例：账号在其他设备登录）
}
@Override
public void onUserSigExpired() { // userSig过期通知
}
});
TUILogin.login("您的userId", "您的userSig", new V2TIMCallback() {
    @Override
    public void onError(int code, String msg) {
        Log.d(TAG, "code: " + code + " msg:" + msg);
    }
    @Override
    public void onSuccess() {
    }
});
```

登录组件参数说明：

- **SDKAppID**：即服务开通中记录到的 **SDKAppID** 信息。
- **SECRETKEY**：即服务开通中记录到的 **SECRETKEY**（密钥）。
- **userId**：当前用户的 ID，字符串类型，长度不超过32字节，不支持使用特殊字符，建议使用英文或数字，可结合业务实际账号体系自行设置。
- **userSig**：根据 SDKAppId、userId，Secretkey 等信息计算得到的安全保护签名，您可以单击 [这里](#) 直接在线生成一个调试的 UserSig，也可以参照我们的 [示例工程](#) 自行计算，更多信息见 [如何计算及使用 UserSig](#)。

4. 开始&停止连麦

在 TUIPlayer 组件内部已经封装好了连麦功能，完成上述步骤后，即可单击[连麦按钮](#)发起&停止连麦，如您需要修改连麦 icon 或显示位置，详情请参见 [连麦按钮](#) 文件。

交流&反馈

更多帮助信息，欢迎加入 QQ 群：**592465424**，进行技术交流和反馈。

集成 TUIPlayer (iOS)

最近更新时间：2025-08-14 09:43:41

组件介绍

TUIPlayer 组件是一套开源的、完整的视频直播互动播放组件，它基于腾讯云 [直播 Live SDK](#) 和 [即时通信 IM SDK](#)，实现直播播放，直播连麦等功能，同时支持弹幕、点赞、美颜等外挂插件，通过 TUIPlayer 组件您可以轻松实现直播视频拉流，快速搭建诸如秀场直播、电商直播等场景化解决方案。



组件集成

步骤一：下载并导入 TUIPlayer 组件

单击进入 [Github](#)，选择克隆/下载代码。将 **iOS 目录** 下的 `TUIPlayer`、`TUIGift` 和 `TUIBarrage` 文件夹拷贝到您在自己工程创建的 `Podfile` 文件同一级目录下，并导入组件。

通过 **cocoapods** 导入组件：

在您的 `Podfile` 文件中添加以下依赖。之后执行 `pod install` 命令，完成导入。

```
# :path => "指向TUIPlayer.podspec所在目录的相对路径"
pod 'TUIPlayer', :path => "TUIPlayer/"

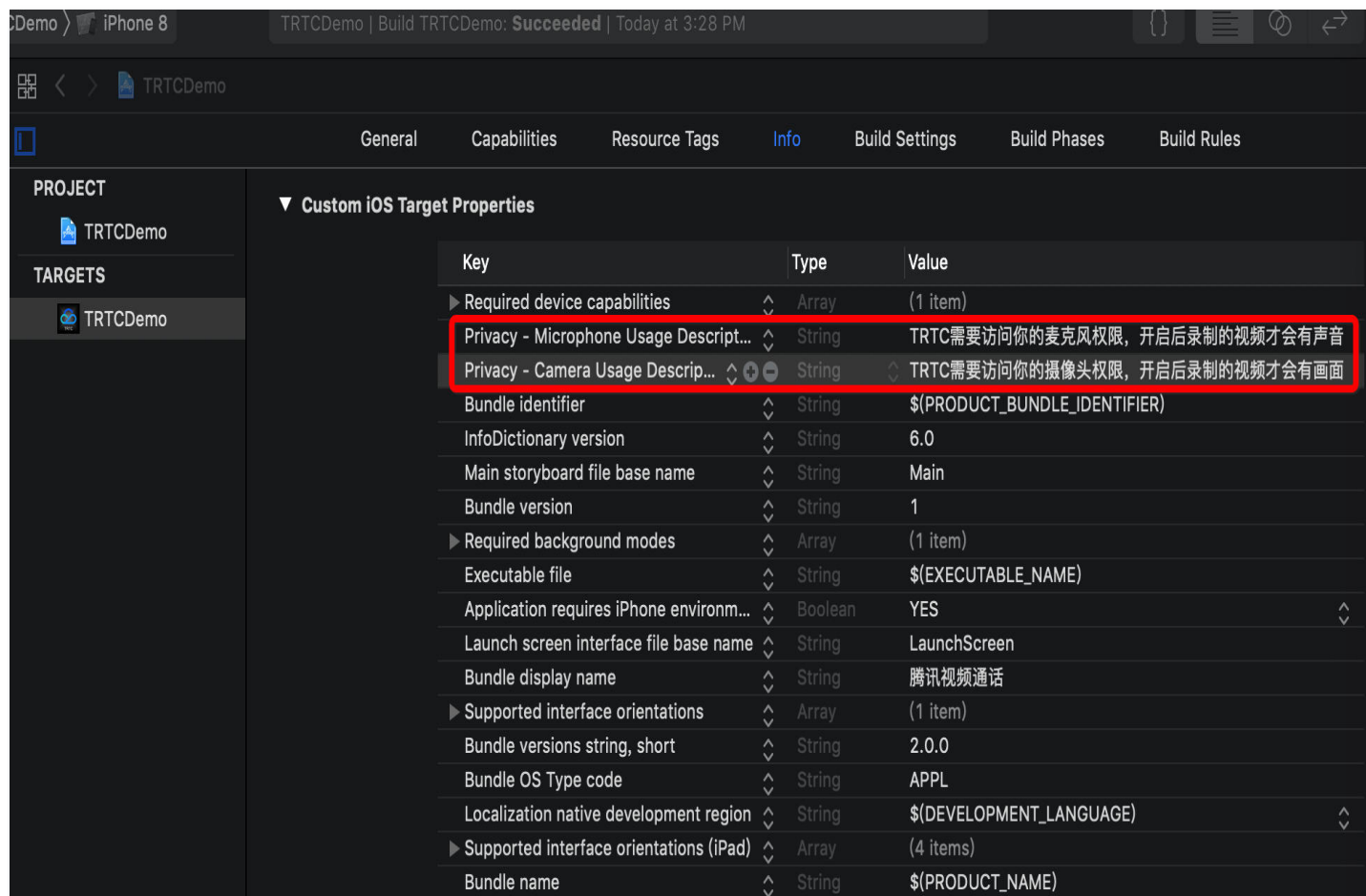
# :path => "指向TUIBarrage.podspec所在目录的相对路径"
pod 'TUIBarrage', :path => "TUIBarrage/"

# :path => "指向TUIGift.podspec所在目录的相对路径"
pod 'TUIGift', :path => "TUIGift/"
```

步骤二：配置权限

使用音视频功能，需要授权麦克风和摄像头的使用权限。在 App 的 `Info.plist` 中添加以下两项，分别对应麦克风和摄像头在系统弹出授权对话框时的提示信息。

```
<key>NSCameraUsageDescription</key>
<string>TUIPlayerApp需要访问您的相机权限，开启后录制的视频才会有画面</string>
<key>NSMicrophoneUsageDescription</key>
<string>TUIPlayerApp需要访问您的麦克风权限，开启后录制的视频才会有声音</string>
```



步骤三：初始化&创建组件

```
// 1. 创建TUIPlayer组件
let mTUIPlayerView = TUIPlayerView(frame: view.bounds);
view.addSubview(mTUIPlayerView);

// 2. 为 TUIPlayerView 设置事件回调 TUIPlayerViewDelegate
mTUIPlayerView.setDelegate(self)
```

步骤四：开始&停止播放

● 开始播放

```
mTUIPlayerView.startPlay(url: "xxxx")
```

腾讯云直播支持两种播放协议：WebRTC 协议（即快直播，更低延时，更强抗性）、RTMP 协议、此处参数 URL 的生成，您可以参考 [示例工程](#) 中封装好的 Utils 方法，更多参数信息请参见 [推拉流 URL](#)。

● 停止播放

```
mTUIPlayerView.stopPlay();
```

步骤五：实现连麦功能（可选）

1. 开通云直播服务：

如果您未开通腾讯云直播相关服务，请先按照如下步骤开通相关服务并完成SDK鉴权：

- [开通云直播服务](#)
- [创建应用并绑定 License](#)，并记录下 `LICENSEURL`、`LICENSEURLKEY` 等关键信息。
- 按照如下方式，完成 SDK 推流鉴权：

```
V2TXLivePremier.setLicence(LICENSEURL, key: LICENSEURLKEY)
```

2. 开通连麦服务

因为连麦时需要更低的延时需求，需要在腾讯云直播控制台开通对应的连麦应用服务，如果您未开通，请登录云直播管理控制台选择 [连麦管理 > 连麦应用](#)，单击新建连麦应用，输入应用名称（例如 `TUIPlayer`），然后在该应用的对应操作栏中，选择应用信息进入应用管理页，查看并记录应用的 `SDKAppID` 和 `SECRETKEY`（密钥）。



3. 组件登录

因为连麦服务，需要观众与主播相互通信，所以需要进行单独登录，登录流程如下：

```
// 组件登录
```



```
TUILogin.initWithSdkAppID (SDKAPPID)
TUILogin.login(userId, userSig: userSig) {
    print("login success")
} fail: { code, message in
    print("login failed, code:\(code), error: \(message ?? "nil")")
}
```

○ 登录组件参数说明：

- **SDKAppID**：服务开通中记录到的 SDKAppID 信息。
- **SECRETKEY**：服务开通中记录到的 SECRETKEY（密钥）。
- **userId**：当前用户的 ID，字符串类型，长度不超过32字节，不支持使用特殊字符，建议使用英文或数字，可结合业务实际账号体系自行设置。
- **userSig**：根据 SDKAppId、userId、Secretkey 等信息计算得到的安全保护签名，您可以单击[这里](#)直接在线生成一个调试的 UserSig，也可以参照我们的[示例工程](#)自行计算，更多信息请参见[如何计算及使用 UserSig](#)。

4. 开始&停止连麦

在 `TUIPlayer` 组件内部已经封装好了连麦功能，完成上述步骤后，即可单击[连麦按钮](#)发起&停止连麦，如您需要修改连麦 icon 或显示位置，详情请参见[连麦按钮](#)文件。

交流&反馈

更多帮助信息，欢迎加入 QQ 群：**592465424**，进行技术交流和反馈。

集成TCPlayer（Web）

最近更新时间：2024-02-29 17:19:51

本文档将介绍适用于点播播放和直播播放的 Web 播放器 SDK（TCPlayer），它可快速与自有 Web 应用集成，实现视频播放功能。Web 播放器 SDK（TCPlayer）内默认包含部分 UI，您可按需取用。

概述

Web 播放器是通过 HTML5 的 `<video>` 标签以及 Flash 实现视频播放。在浏览器不支持视频播放的情况下，实现了视频播放效果的多平台统一体验，并结合腾讯云点播视频服务，提供防盗链和播放 HLS 普通加密视频等功能。

协议支持

音视频协议	用途	URL 地址格式	PC 浏览器	移动浏览器
MP3	音频	http://xxx.vod.myqcloud.com/xxx.mp3	支持	支持
MP4	点播	http://xxx.vod.myqcloud.com/xxx.mp4	支持	支持
HLS（M3U8）	直播	http://xxx.liveplay.myqcloud.com/xxx.m3u8	支持	支持
	点播	http://xxx.vod.myqcloud.com/xxx.m3u8	支持	支持
FLV	直播	http://xxx.liveplay.myqcloud.com/xxx.flv	支持	部分支持
	点播	http://xxx.vod.myqcloud.com/xxx.flv	支持	部分支持
WebRTC	直播	webrtc://xxx.liveplay.myqcloud.com/live/xxx	支持	支持

❗ 说明：

- 视频编码格式仅支持 H.264 编码。
- 播放器兼容常见的浏览器，播放器内部会自动区分平台，并使用最优的播放方案。
- HLS、FLV 视频在部分浏览器环境播放需要依赖 Media Source Extensions。

- 在不支持 WebRTC 的浏览器环境，传入播放器的 WebRTC 地址会自动进行协议转换来更好的支持媒体播放。

功能支持

功能\浏览器	Chrome	Firefox	Edge	QQ 浏览器	Mac Safari	iOS Safari	微信	Android Chrome	IE 11
播放器尺寸设置	✓	✓	✓	✓	✓	✓	✓	✓	✓
续播功能	✓	✓	✓	✓	✓	✓	✓	✓	✓
倍速播放	✓	✓	✓	✓	✓	✓	✓	✓	✓
缩略图预览	✓	✓	✓	✓	—	—	—	—	✓
切换 fileID 播放	✓	✓	✓	✓	✓	✓	✓	✓	✓
镜像功能	✓	✓	✓	✓	✓	✓	✓	✓	✓
进度条标记	✓	✓	✓	✓	✓	—	—	—	✓
HLS 自适应码率	✓	✓	✓	✓	✓	✓	✓	✓	✓
Referer 防盗链	✓	✓	✓	✓	✓	✓	✓	—	✓
清晰度切换提示	✓	✓	✓	✓	—	—	—	✓	✓
试看功能	✓	✓	✓	✓	✓	✓	✓	✓	✓
HLS 标准加密播放	✓	✓	✓	✓	✓	✓	✓	✓	✓
HLS 私有加密播放	✓	✓	✓	—	—	—	Android: ✓	✓	✓

							• iOS: -		
视频统计信息	✓	✓	✓	✓	-	-	-	-	-
视频数据监控	✓	✓	✓	✓	-	-	-	-	-
自定义提示文案	✓	✓	✓	✓	✓	✓	✓	✓	✓
自定义UI	✓	✓	✓	✓	✓	✓	✓	✓	✓
弹幕	✓	✓	✓	✓	✓	✓	✓	✓	✓
水印	✓	✓	✓	✓	✓	✓	✓	✓	✓
幽灵水印	✓	✓	✓	✓	✓	✓	✓	✓	✓
视频列表	✓	✓	✓	✓	✓	✓	✓	✓	✓
弱网追帧	✓	✓	✓	✓	✓	✓	✓	✓	✓

- ❗ 说明:
- 视频编码格式仅支持 H.264 编码。
 - Chrome、Firefox 包括 Windows、macOS 平台。
 - Chrome、Firefox、Edge 及 QQ 浏览器播放 HLS 需要加载 `hls.js`。
 - Referer 防盗链功能是基于 HTTP 请求头的 Referer 字段实现的，部分 Android 浏览器发起的 HTTP 请求不会携带 Referer 字段。

播放器兼容常见的浏览器，播放器内部会自动区分平台，并使用最优的播放方案。例如：在 Chrome 等现代浏览器中优先使用 HTML5 技术实现视频播放，而手机浏览器上会使用 HTML5 技术或者浏览器内核能力实现视频播放。

准备工作

播放器 SDK Web端（TCPlayer）自 5.0.0 版本起需获取 License 授权后方可使用。若您无需使用新增的高级功能，可直接申请基础版 License 以**继续免费使用 TCPlayer**；若您需要使用新增的高级功能则需购买高级版 License。具体信息如下：

TCPlayer 功能	功能范围	所需 License	定价	授权单位
基础版功能	包含 5.0.0 以前版本提供的全部功能，详	播放器 Web 端基础版 License	0元 免费申请	精准域名（1个 License 最多可授

	情见 产品功能			权10个精准域名)
高级版功能	基础版功能、 VR 播放 、 安全检查	播放器 Web 端高级版 License	399元/月 立即购买	泛域名 (1个 License 最多可授权1个泛域名)

❗ 说明:

1. 播放器 Web 端基础版 License 可免费申请，申请后有效期默认1年；在有效期剩余时间小于30天时，可免费续期。
2. 为方便本地开发，播放器不会校验 localhost 或者 127.0.0.1，因此申请 License 时不需要申请这类本地服务域名。

集成指引

通过以下步骤，您就可以在网页上添加一个视频播放器。

步骤1：在页面中引入文件

播放器 SDK 支持 cdn 和 npm 两种集成方式：

1. 通过 cdn 集成

在本地的项目工程内新建 index.html 文件，html 页面内引入播放器样式文件与脚本文件：

```
<link
href="https://web.sdk.qcloud.com/player/tcplayer/release/v5.1.0/tcplayer
.min.css" rel="stylesheet"/>
<!--播放器脚本文件-->
<script
src="https://web.sdk.qcloud.com/player/tcplayer/release/v5.1.0/tcplayer.
v5.1.0.min.js"></script>
```

建议在使用播放器 SDK 的时候自行部署资源，[单击下载播放器资源](#)。部署解压后的文件夹，不能调整文件夹里面的目录，避免资源互相引用异常。

如果您部署的地址为 `aaa.xxx.ccc`，在合适的地方引入播放器样式文件与脚本文件，自行部署情况下，需要手动引用资源包文件夹 `libs` 下面的依赖文件，否则会默认请求腾讯云 cdn 文件。

```
<link href="aaa.xxx.ccc/tcplayer.min.css" rel="stylesheet"/>
<!--如果需要在 Chrome 和 Firefox 等现代浏览器中通过 H5 播放 HLS 格式的视频，需要
在 tcplayer.vx.x.x.min.js 之前引入 hls.min.x.xx.m.js。-->
<script src="aaa.xxx.ccc/libs/hls.min.x.xx.m.js"></script>
<!--播放器脚本文件-->
```

```
<script src="aaa.xxx.ccc/tcplayer.vx.x.x.min.js"></script>
```

2. 通过 npm 集成

首先安装 tcplayer 的 npm 包：

```
npm install tcplayer.js
```

导入 SDK 和样式文件：

```
import TCPlayer from 'tcplayer.js';
import 'tcplayer.js/dist/tcplayer.min.css';
```

步骤2：放置播放器容器

在需要展示播放器的页面位置加入播放器容器。例如，在 index.html 中加入如下代码（容器 ID 以及宽高都可以自定义）。

```
<video id="player-container-id" width="414" height="270" preload="auto"
playsinline webkit-playsinline>
</video>
```

❗ 说明：

- 播放器容器必须为 `<video>` 标签。
- 示例中的 `player-container-id` 为播放器容器的 ID，可自行设置。
- 播放器容器区域的尺寸，建议通过 CSS 进行设置，通过 CSS 设置比属性设置更灵活，可以实现例如铺满全屏、容器自适应等效果。
- 示例中的 `preload` 属性规定是否在页面加载后载入视频，通常为了更快的播放视频，会设置为 `auto`，其他可选值：`meta`（当页面加载后只载入元数据），`none`（当页面加载后不载入视频），移动端由于系统限制不会自动加载视频。
- `playsinline` 和 `webkit-playsinline` 这几个属性是为了在标准移动端浏览器不劫持视频播放的情况下实现行内播放，此处仅作示例，请按需使用。
- 设置 `x5-playsinline` 属性在 TBS 内核会使用 X5 UI 的播放器。

步骤3：播放器初始化

页面初始化后，即可播放视频资源。播放器同时支持点播和直播两种播放场景，具体播放方式如下：

- 点播播放：播放器可以通过 FileID 播放腾讯云点播媒体资源，云点播具体流程请参见 [使用播放器播放](#) 文档。

- 直播播放：播放器通过传入 URL 地址，即可拉取直播音视频流进行直播播放。腾讯云直播 URL 生成方式可参见 [自主拼装直播 URL](#)。

通过 URL 播放（直播、点播）

在页面初始化之后，调用播放器实例上的方法，将 URL 地址传入方法。

```
// player-container-id 为播放器容器 ID，必须与 html 中一致
var player = TCPlayer('player-container-id', {
  sources: [{
    src: 'path/to/video', // 播放地址
  }],
  licenseUrl: 'license/url', // license 地址，参考准备工作部分，在视立方
控制台申请 license 后可获得 licenseUrl
});

// player.src(url); // url 播放地址
```

通过 FileID 播放（点播）

在 index.html 页面初始化的代码中加入以下初始化脚本，传入在准备工作中获取到的 fileID（[媒资管理](#)）中的视频 ID 与 appID（在账号信息>[基本信息](#)中查看）。

```
var player = TCPlayer('player-container-id', { // player-container-
id 为播放器容器 ID，必须与 html 中一致
  fileID: '3701925921299637010', // 请传入需要播放的视频 fileID（必须）
  appID: '1500005696', // 请传入点播账号的 appID（必须）
  //私有加密播放需填写 psign，psign 即播放器签名，签名介绍和生成方式参见链
接：https://cloud.tencent.com/document/product/266/42436

  psign:'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBHJZCI6MTUwMDAwNTY5
NiwZmlsZUlkIjoimZcwMTkyNTkyMTI5OTYzNzAxMCIsImN1cnJlbnRUaW1lU3RhbXAi
OjE2MjY4NjAxNzYsImV4cGlyZVRpbWVtdGFtcCI6MjYyNjg1OTE3OSwicGNmZyI6InBy
aXZhdGUiLCJ1cmxBY2Nlc3NjbmZvIjp7InQioiI5YzkyYjBhYiJ9LCJkcm1MaWNlbnNl
SW5mbyI6eyJleHBpcmVUaW1lU3RhbXAiOjE2MjY4NTkxNzksInN0cm1jdE1vZGUiojJ9
fQ.Bo5K5ThInc4n8AlzIZQ-CP9a49M2mEr9-zQLH9ocQgI',
  licenseUrl: 'license/url', // 参考准备工作部分，在视立方控制台申请
license 后可获得 licenseUrl
});
```


⚠ 注意：

要播放的视频建议使用腾讯云转码，原始视频无法保证在浏览器中正常播放。

步骤4: 更多功能

播放器可以结合云点播的服务端能力实现高级功能，比如自动切换自适应码流、预览视频缩略图、添加视频打点信息等。这些功能在 [播放长视频方案](#) 中有详细的说明，可以参考文档实现。

此外，播放器还提供更多其他功能，功能列表和使用方法请参见 [功能展示](#) 页面。