

TTT Miniapp SDK Tutorial

简介

三体云 小程序 SDK可实现 微信小程序 与 三体云其他各端之间的音视频通话。

1. 扫码体验

登录 [3t 官网](#)，微信扫码体验。

2. demo 源码

托管于 [github](#)。

开发环境

1. 请确保本地已安装微信开发者工具

2. 请确保在微信公众平台账号的开发设置中，设置以下域名请求权限：

[request合法域名]中配置如下：

- <https://apiusercenter.3ttech.cn>
- <https://log.wushuangtech.com:15100>

[socket合法域名]中配置如下：

- <wss://webdispatch.3ttech.cn>
- <wss://webmedia1.3ttech.cn>
- <wss://webmedia2.3ttech.cn>
- <wss://webmedia3.3ttech.cn>
- <wss://webmedia4.3ttech.cn>
- <wss://webmedia5.3ttech.cn>
- <wss://webmedia6.3ttech.cn>
- <wss://webmedia7.3ttech.cn>
- <wss://webmedia8.3ttech.cn>
- <wss://shenzhen1.3ttech.cn>
- <wss://miniapp1.3ttech.cn>

1. 微信小程序组件

请确保有一个支持 **live-pusher** 和 **live-player** 组件的微信公众平台账号

只有特定行业的认证企业账号才可使用这两个组件。详情请[点击这里](#)

注：在开始开发之前，请先参阅[微信小程序官方开发文档](#)

在微信小程序中实现音视频功能，需要使用微信的 live-player 组件和 live-pusher 组件

- live-player 组件

实现微信小程序实时音视频播放功能。

创建 live-player 的示例代码如下：

- live-pusher 组件

实现微信小程序实时音视频录制功能。

创建 live-pusher 的示例代码如下：

运行示例程序

本 Demo 基于 TTT Miniapp SDK 开发，开发者可以通过微信小程序实现与 3T 其他各端间视频通话互动等功能。

1. 首先应具备 三体云 AppID
2. 下载本页示例程序
3. 打开 *utils* 文件夹，在 *config.js* 文件中设置正确的 App ID：

```
const APPID = 'xxxxxx'
```

4. 启动微信开发者工具并导入该示例程序
5. 输入频道名，加入频道。邀请你的朋友加入同一个频道，就可以开始视频互通了。

小程序 SDK

1. SDK 下载

下载 [TTT Miniapp SDK](#)，解压改压缩包，将js文件重新命名为“3T_Miniapp_SDK_for_WeChat.js”，保存到本示例程序的 *lib* 目录下

2. 集成

```
const TTTMAEngine = require('../lib/3T_Miniapp_SDK_for_WeChat');
```

API 使用概述

本节对 3t miniapp sdk 的使用流程概述如下：

1. 创建 3t client 并初始化

```
// appId : AppID  
// userId : 用户ID  
client = new TTTMAEngine.Client(appId, userId);
```

```
// appId : AppID
// userId : 用户ID
// onSuccess : 成功时回调
// onFailure : 失败时回调
client.init(appId, userId, onSuccess, onFailure);
```

2. 加入房间

待 client 初始化成功后, 在其 onSuccess 回调中, 调用 join 登入房间, 如下 :

```
// roomId : 房间号
// userId : 用户ID
// onSuccess : 成功时回调
// onFailure : 失败时回调
join(roomId, userId, onSuccess, onFailure);
```

3. 发布本地音视频

加入房间后, 用户可以调用 publish 方法发布本地音视频流, 代码如下 :

```
// onSuccess : 成功时回调
// onFailure : 失败时回调
publish(onSuccess, onFailure)
```

4. 订阅远端音视频

注 : 首先, 本端需要通过监听如下事件, 来感知有流发布 :

```
client.on("stream-added", callback);
```

callback 的参数 e, 其中携带了 e.uid 用户ID参数 ;

当监听到 'stream-added' 时, 说明此时有用户发布流, 本端就可以通过 subscribe 方法, 订阅该用户音视频流, 代码如下 :

```
// userId : 用户ID
// onSuccess : 成功时回调
// onFailure : 失败时回调
subscribe(userId, onSuccess, onFailure)
```

5. 离开房间

```
// onSuccess : 成功时回调  
// onFailure : 失败时回调  
leave(onSuccess, onFailure)
```

还有更多细节，可以完整的 API 文档请参考 [开发者中心](#)

代码许可

MIT 许可 (MIT)