

实时音视频 AI Coding



腾讯云

【 版权声明 】

©2013–2026 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

AI Coding

快速入门

MCP

Skills

AI Coding 快速入门

最近更新时间：2026-04-02 11:22:09

如果你正在使用 AI 编辑器集成 Tencent RTC，这里提供完成接入所需的关键资源，包括 MCP、Skills、面向 LLM 的文档入口，以及 Live / Room / Call 的快速开始示例，支持 Web / iOS / Android / Flutter。

资源导航

- [RTC MCP Server](#)（必做）
- [RTC Skills](#)（推荐）
- [给 AI 提供文档上下文](#)

快速开始

- [直播互动（Live）](#)
- [视频会议（Room）](#)
- [音视频通话（Call）](#)

准备账号与凭证（必做）

开始前请先在腾讯云控制台创建应用，并准备以下凭证：

凭证	获取位置	用途
SDKAppID	TRTC 控制台	SDK 初始化、MCP 环境变量
SecretKey	TRTC 控制台 → 应用管理	生成 UserSig（仅用于测试）、MCP 环境变量

配置 MCP（必做）

在你的 AI 编辑器（CodeBuddy / Trae / Cursor / Codex / Claude Code / Claude Desktop 等）里配置 MCP：

```
{
  "mcpServers": {
    "tencentcloud-sdk-mcp": {
      "command": "npx",
      "args": ["-y", "@tencentcloud/sdk-mcp@latest"],
      "env": {
        "SDKAPPID": "YOUR_SDKAPPID",
        "SECRETKEY": "YOUR_SECRET_KEY"
      }
    }
  }
}
```

```
}  
}  
}  
}
```

查看 CodeBuddy、Trae、Cursor、Codex、Claude Code、Claude Desktop 等的详细安装说明，请点击 [MCP 配置](#)。

安装 Skills（推荐）

运行以下命令安装 Skills，CLI 会自动识别你正在使用的工具（CodeBuddy / Trae / Cursor / Codex / Claude Code / Claude Desktop 等）：

```
npx skills add Tencent-RTC/tencent-rtc-skills
```

建议安装 Skills，以提高 AI 调用 MCP 工具的稳定性与准确率。详细说明见：[Tencent RTC Skills](#)。

给 AI 提供文档上下文

除了使用 MCP 调用工具外，你还可以直接把文档 URL 提供给 AI，作为补充上下文。对于需要查 API、找示例、查参数的问题，这种方式通常更直接。

1. 索引文档 `llms.txt`

当你还不确定该看哪篇文档时使用。

```
https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms.txt
```

2. 产品聚合文档 `llms-full-*.txt`

当你已经知道产品时，直接提供对应产品的 `llms-full-*.txt`：

产品	文档 URL
LiveKit	https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-livekit.txt
RoomKit	https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-roomkit.txt
CallKit	https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-callkit.txt
WebRTC SDK	https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-webrtc-sdk.txt
TRTC SDK	https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-trtc-sdk.txt

Native Live
SDK

<https://gitee.com/Tencent-RTC/docs/raw/main/zh/llms-full-native-live-sdk.txt>

3. 单页 Markdown *.md

当你已经知道具体功能页时使用，先通过 `llms.txt` 或对应产品的 `llms-full-*.txt` 找到页面，再提供单页面 URL 给 AI。

示例：

```
https://gitee.com/Tencent-RTC/docs/raw/main/zh/livekit/integration-  
live.md
```

直播互动（Live）

直接复制下面 Prompt 到 IDE 对话框，集成直播互动 LiveKit：

任务：在当前项目中集成 LiveKit（先跑通视频直播基础流程）

执行要求：

1. 自动识别框架（Web Vue3/Flutter/Android/iOS）。
2. 无法识别时，需要调用 ``present_framework_choice`` 给用户选择。
3. 按平台调用 MCP 工具：
 - Web: ``get_web_live_uikit_integration``
 - Native: ``get_native_live_uikit_integration``
4. 参数: ``framework``。
5. 调用 ``get_usersig`` 完成初始化和登录。
6. 生成主播开播页、观众观看页、基础互动入口。

验收清单：

- 主播可开播
- 观众可观看
- 基础互动可用（如连麦入口）

常见问题排查：

- 黑屏：检查设备权限与渲染初始化
- 无声：检查音频权限与设备路由

视频会议（Room）

直接复制下面 Prompt 到 IDE 对话框，集成视频会议 RoomKit：

任务：在当前项目中集成 RoomKit（先跑通 quick-start 模式）

执行要求：

1. 自动识别框架（Web Vue3）。
2. 无法识别时，需要调用 `present_framework_choice` 给用户选择。
3. 按平台调用 MCP 工具：
 - Web: `get_web_room_uikit_integration`
4. 参数: `framework`。
5. 调用 `get_usersig` 完成初始化和登录。
6. 生成会议入口页、会议房间页、路由注册。

验收清单：

- 设备检测正常
- 可入会并看到音视频
- 屏幕共享可用（平台支持时）

常见问题排查：

- 进房失败：检查 roomId 与登录状态
- 无画面/无声音：检查摄像头麦克风权限

音视频通话（Call）

直接复制下面 Prompt 到 IDE 对话框，集成音视频通话 CallKit:

任务：在当前项目中集成 CallKit（先跑通 1v1 语音通话和视频通话）

执行要求：

1. 自动识别框架（React/Vue3/Flutter/Android/iOS）。
2. 无法识别时，需要调用 `present_framework_choice` 给用户选择。
3. 按平台调用 MCP 工具：
 - Web: `get_web_call_uikit_integration`
 - Native: `get_native_call_uikit_integration`
4. 参数: `framework`。
5. 调用 `get_usersig` 完成初始化和登录。
6. 生成最小可运行的呼叫入口和来电处理页面。

验收清单：

- 初始化与登录成功
- 能发起 1v1 视频通话

- 能收到来电并接听

常见问题排查：

- 60001：重新生成 `userSig`
- 通话失败：检查网络与设备权限

参考

- [MCP NPM](#)
- [Tencent RTC Skills](#)
- [直播/语聊房-开通服务](#)
- [视频会议-开通服务](#)
- [视频通话-开通服务](#)

MCP

最近更新时间：2026-03-19 16:30:22

MCP (Model Context Protocol) 标准化了应用向 LLM 提供上下文的方式。本文介绍如何使用 MCP 集成实时音视频 TRTC UIKit 组件和底层 SDK，仅支持 `studio` 模式传输。

MCP 功能

- **智能集成**：覆盖 Live / Room / Call UIKit 和底层 SDK (WebRTC 和 Native RTC) 集成工具。
- **功能详情查询**：提供 UI 组件与 API 使用详情。
- **常见问题查询**：常见问题查询工具。
- **测试 UserSig 生成**：UserSig 生成工具（仅用于测试）。
- **多平台支持**：完整的 Web、iOS、Android、Flutter 文档支持。

前置条件

MCP 已经发布在 [NPM](#) 上，您可以通过 `npx` 安装在任何支持 MCP 的 AI 编辑器中。使用 MCP 您需要先在腾讯云控制台创建应用，并获取以下凭证：

凭证	获取位置	用途
<code>SDKAppID</code>	TRTC 控制台	MCP 环境变量
<code>SecretKey</code>	TRTC 控制台 → 应用管理	MCP 环境变量，生成 UserSig（仅用于测试）

MCP 配置

在 AI 编辑器中配置

在您使用的 AI 编辑器（CodeBuddy / Trae / Cursor / Codex / Claude Code / Claude Desktop 等）配置 MCP。

CodeBuddy

单击设置 > 选择 MCP > 配置 MCP > 打开 `mcp.json` 文件，配置以下内容：

```
{
  "mcpServers": {
    "tencentcloud-sdk-mcp": {
      "command": "npx",
```

```
"args": ["-y", "@tencentcloud/sdk-mcp@latest"],
"env": {
  "SDKAPPID": "YOUR_SDKAPPID",
  "SECRETKEY": "YOUR_SECRET_KEY"
}
}
```

Trae

单击**设置** > 选择 **MCP** > **手动添加 MCP** > **原始配置(JSON)** > 打开 mcp.json 文件，配置以下内容：

```
{
  "mcpServers": {
    "tencentcloud-sdk-mcp": {
      "command": "npx",
      "args": ["-y", "@tencentcloud/sdk-mcp@latest"],
      "env": {
        "SDKAPPID": "YOUR_SDKAPPID",
        "SECRETKEY": "YOUR_SECRET_KEY"
      }
    }
  }
}
```

Cursor

单击**设置** > 选择 **Tools & MCP** > **Add Custom MCP** > 打开 mcp.json 文件，配置以下内容：

```
{
  "mcpServers": {
    "tencentcloud-sdk-mcp": {
      "command": "npx",
      "args": ["-y", "@tencentcloud/sdk-mcp@latest"],
```

```
"env": {
  "SDKAPPID": "YOUR_SDKAPPID",
  "SECRETKEY": "YOUR_SECRET_KEY"
}
}
```

Codex

1. 运行以下命令配置 MCP，将 SDKAPPID 和 SECRETKEY 替换为真实的应用信息。

```
codex mcp add tencentcloud-sdk-mcp \
  --env SDKAPPID=YOUR_SDK_APP_ID --env SECRETKEY=YOUR_SECRET_KEY \
  -- npx -y @tencentcloud/sdk-mcp@latest
```

如需项目级配置，可追加 `--scope project`，写入项目根目录的 `.mcp.json`。

2. 运行 `codex mcp list` 验证 MCP 是否配置成功，列表中出现 `tencentcloud-sdk-mcp` 表示配置成功。

Claude Code

1. 运行以下命令配置 MCP，将 SDKAPPID 和 SECRETKEY 替换为真实的应用信息。

```
claude mcp add tencentcloud-sdk-mcp \
  --env SDKAPPID=YOUR_SDK_APP_ID --env SECRETKEY=YOUR_SECRET_KEY \
  -- npx -y @tencentcloud/sdk-mcp@latest
```

如需项目级配置，可追加 `--scope project`，写入项目根目录的 `.mcp.json`。

2. 运行 `claude mcp list` 验证 MCP 是否配置成功，列表中出现 `tencentcloud-sdk-mcp` 表示配置成功。

Claude Desktop

单击设置 > 选择 Developer > Edit Config > `claude_desktop_config.json`，配置以下内容：

```
{
  "mcpServers": {
    "tencentcloud-sdk-mcp": {
      "command": "npx",
      "args": ["-y", "@tencentcloud/sdk-mcp@latest"],
      "env": {
        "SDKAPPID": "YOUR_SDKAPPID",
        "SECRETKEY": "YOUR_SECRET_KEY"
      }
    }
  }
}
```

环境变量

- **SDKAPPID:** 实时音视频 TRTC 和即时通信 IM 应用 ID。
- **SECRETKEY:** SDKAPPID 对应的密钥，用于生成测试 UserSig。

验证 MCP

直接复制下面 Prompt 到 IDE 对话框，验证 MCP 的可用性：

```
使用 tencentcloud-sdk-mcp 工具获取 test001 的 userSig
```

当 Agent 输出 SDKAppID、userID、userSig 信息时说明 MCP 配置成功。

附录：全局手动安装

如果需要全局安装，可以在命令行执行以下命令：

```
npx -y @tencentcloud/sdk-mcp@latest
```

Skills

最近更新时间：2026-03-19 16:30:14

RTC Skills 为腾讯云实时音视频场景提供结构化指引。它可以将自然语言需求转为可执行步骤，并帮助 Agent 按正确顺序调用对应的 MCP 工具。使用 RTC Skills，让 AI Agent 更高效地完成 Live / Room / Call 的集成。

前置条件

使用 Skills 前，请先完成 MCP Server 配置，[MCP 配置说明](#)。

安装

安装 RTC Skills 全量包，安装过程中需要您选择使用的工具（CodeBuddy / Trae / Cursor / Codex / Claude Code / Claude Desktop 等）：

```
npx skills add Tencent-RTC/tencent-rtc-skills
```

如需手动导入，可通过 GitHub 克隆：

```
git clone https://github.com/Tencent-RTC/tencent-rtc-skills.git
~/skills/tencent-rtc-skills
```

优势

- **结构化知识注入**：将领域知识直接注入 Agent 上下文，降低歧义，提升输出一致性。
- **与 Prompt 工程深度配合**：内置标准化执行流程，帮助 Agent 按可复用路径完成集成。
- **零运行时依赖**：Skills 为静态文本资源，无需额外启动服务。
- **场景覆盖完整**：单一技能包覆盖 Live / Room / Call 场景集成。

参考

- [Tencent RTC Skills](#)
- [MCP NPM](#)