

# Anonymous Chatroom (Next.js + Supabase) - Trae 实战指南

当前版本: v11.0 (Trae 实战增强版)  
项目: Anonymous Chatroom (Next.js + Supabase)

这里没有枯燥的理论。这份指南记录了我们如何用 **Trae** (字节跳动最新 AI IDE) 构建 Anonymous Chatroom。在这个模式下，你不再是单纯的码农，而是 **AI 的飞行员和质检官**。

我们只有几个核心原则：**AI 得自己检查代码**、**Prompt 先行**、**上下文要给够**。

## 1. 我们的 workflow

AI 写代码很快，但也容易出错。为了不被 AI 坑，我们总结了一套**四步走**的方法。核心就一条：充分利用 Trae 的原生能力，让 AI 自己写代码，然后自己测试，直到跑通为止。

### 第一步：立规矩 (Rule & Self-Check)

AI 第一遍写的代码通常都有 Bug。别急着复制粘贴，先在 Trae 里把规矩立好。

- 配置 **.cursorrules** (或 **Trae Rule**):
  - Trae 允许你配置全局或项目级别的 Rule。把我们的核心原则写进去，这样不用每次都啰嗦。
- 你写的代码，你自己测：
  - **Prompt 话术**: "写个音频录制功能，顺便把 Jest 测试也写了。要是测试跑不通，你自己修好再告诉我。"

### 第二步：找帮手 (Skill Acquisition)

别让 AI 裸奔。Trae 强大的地方在于它能挂载各种 Skill。

- 去哪找 Skill:
  - 访问 **\*\* https://skillsmp.com/search \*\***。这里是 Skill 的大本营。
  - **怎么用**: 搜你需要的 (比如 "react", "tailwindcss")，复制安装指令，在 Trae 的聊天框里装上。
- 本项目常用 Skill:
  - **frontend-design**: 搞不定复杂的 CSS? 用这个。粒子特效、流体光标，它在行。
  - **frontend-code-review**: 提交代码前心里没底? 用这个。它能帮你查查 Hooks 依赖对不对，Tailwind 写没写歪。
  - **test-generator**: 测试用例写得头大? 用这个帮你补全。

### 第三步：想清楚再动手 (Ideation & PRD)

写代码前，先用自然语言把逻辑理顺。

- 利用 Trae 的 Solo 模式：
  - 这是 Trae 的杀手锏。开启 **Solo 模式**，它会自己思考、自己规划、自己写代码、自己修 Bug。
  - 你只需要像产品经理一样跟它聊：“我要做一个带波形图的音频播放器”，它会生成一份简单的 PRD 给你确认。
- 我们的技术栈选择：
  - 前端: Next.js (App Router)。没别的，React 生态最强，SSR 和 SEO 都好使。
  - 后端: Supabase。开源版的 Firebase。PostgreSQL、Auth、Realtime、Storage 全都有，完美契合 Serverless，重点是免费。
  - 部署: Vercel + Cloudflare。国内访问加速方案，个人项目白嫖首选。

## 第四步：开发与验收 (Development & Verify)

- 怎么做：
  1. 一次配置，永久复用：确保 Trae 里的 Rule 和 Skill 都配置好了。
  2. 全自动驾驶：切换到 **Solo 模式**，输入需求。
  3. 看它表演：你会看到它生成代码 -> 运行测试 -> 报错 -> 读日志 -> 修复 -> 再跑。
  4. 最后把关：只有当 AI 陷入死循环或者方向彻底错了，你再介入。

---

## 2. 目录

1. [项目成品一览 \(Showcase\)](#)
2. [我们的工作流](#)
3. [架构图 \(System Architecture\)](#)
4. [实战案例：我们要怎么做](#)
5. [国内特供：Vercel + Cloudflare 部署指南](#)
6. [怎么写好 Prompt](#)
7. [Skill 怎么用](#)
8. [附录：拿来即用的 Prompt](#)

---

## 1. 项目成品一览 (Showcase)

这是一个基于 **Next.js 14** 和 **Supabase** 构建的现代化匿名聊天室，完美支持移动端。

- 核心玩法：
  - 匿名潜水: 访客无需注册，进来就能聊。
  - 上帝视角: 房主通过密码登录，拥有特殊标识和管理权限。
  - 实时互动: 基于 Supabase Realtime，消息秒推，支持图片、语音、表情包。
  - 手机党福音: 专为移动端优化的 UI，消息气泡、触感反馈细节拉满。
  - 个性化: 内置多套紫色系主题，支持深色模式。

---

## 3. 架构图 (System Architecture)

这是我们项目的全貌。生成自AI, 也应用于AI



## 4. 实战案例：我们要怎么做

光说不练假把式。来看看我们在 Anonymous Chatroom 里实际是怎么开发的。

### 案例 1: 架构设计 - 媒体消息系统

任务: 能发图片和音频。

**Prompt 怎么写:**

理论上

"我想发图片和音频。基于 Supabase 帮我设计一下。

1. messages 表要加什么字段? (type, url, duration 这些需不需要? )
2. Storage 的 Bucket 怎么建? 权限怎么设?
3. SQL 脚本直接给我。"

实际上我说的

"实现一下发图片和发视频的功能

结果: AI 搞定了枚举字段、Bucket 规划, 连 SQL 脚本都写好了。

### 案例 2: 多种方案 - 彩蛋特效系统

任务: 输入“捡手机”全屏飘花。

**Prompt 怎么写:**

理论上

"我们要搞个彩蛋给我一些方案以供选择。

结果: AI 先提供了选项, 然后再实现

### 案例 3: 做个组件 - 音频播放器

任务: 带波形动画的播放器。

**Prompt 怎么写:**

"用 frontend-design 技能帮我做一个 AudioPlayer。

1. 图标用是一个幽灵。
2. 播放的时候给我整一个 SVG 动画，像幽灵晃动那种。
3. 播完了记得重置图标。
4. 优化性能。"

结果: GhostWizard 和 MagicWand 两个子组件直接生成，效果很棒。

## 5. 国内特供：Vercel + Cloudflare 部署指南

Vercel 很好用，但 .vercel.app 域名在国内被墙了。为了让国内朋友能访问，我们需要一套组合拳：**Vercel + Cloudflare + 自定义域名**。

### 第一步：准备工作

1. 买个域名：
  - 不用买贵的
  - 注意: 不需要备案（因为服务器在海外），但需要实名认证。
2. 注册 Cloudflare:
  - 去 Cloudflare 官网注册个账号（免费版足够）。

### 第二步：域名托管给 Cloudflare

1. 在 Cloudflare 点击 **"Add a site"**，输入你刚买的域名。
2. 选择 **Free Plan**。
3. Cloudflare 会给你两个 **Nameservers** (例如 bob.ns.cloudflare.com)。
4. 回到你买域名的地方（比如阿里云控制台），把域名的 DNS 服务器修改为 Cloudflare 给你的那两个。

### 第三步：Vercel 部署

1. 代码推送到 GitHub。
2. 在 Vercel 导入项目，一路默认配置点 **Deploy**。
3. 部署成功后，Vercel 会给你一个分配的域名（国内打不开，先不管）。

### 第三步：配置 DNS (关键步骤)

1. 在 Vercel 项目设置 -> **Domains**，添加你的自定义域名（比如 chat.yourdomain.xyz）。
  - Vercel 会提示你添加 A 记录或 CNAME。
2. 回到 Cloudflare 的 DNS 设置：
  - 国内优化黑科技: 不要直接用 Vercel 默认的 IP。
  - 添加一条 **A 记录**:
    - **Name**: @ (或者 www)
    - **Content**: 76.223.126.88

- 注: 这个 IP 是 Vercel 专门为被墙地区优化的官方 IP。
- 或者使用 **CNAME**:
  - **Name**: www
  - **Content**: cname.vercel-dns.com
- **重要**: 确保 Cloudflare 的小云朵图标是 **点亮状态 (Proxied)**, 这样流量才会走 Cloudflare 的 CDN。

## 第四步: SSL 配置

1. 在 Cloudflare -> **SSL/TLS**:
  - 把模式改为 **Full (Strict)**。
  - 这是为了防止重定向循环, 因为 Vercel 强制 HTTPS。

搞定! 现在你的域名不仅能国内秒开, 还自带 Cloudflare 的免费 CDN 和防攻击保护。

---

## 6. 怎么写好 Prompt

简单说就是 **C-R-I** 原则, 别想太复杂:

1. **Context (背景)**: 告诉它你在干嘛。"我在写 AudioRecorder, 文件在 src/lib/audio-recorder.ts。"
2. **Role (角色)**: 给它戴高帽。"你是 Web Audio API 的专家。"
3. **Instruction (指令)**: 说人话。"把这段代码重构一下, iOS Safari 播不了 audio/mp4, 你给修修。"

---

## 7. Skill 怎么用

Trae 的 Skill 系统是让 AI 变得专业的关键。

### 7.1 怎么下载和安装

1. 找 **Skill**: 去 `** https://skillsmp.com/search **` 逛逛。
2. 选 **Skill**: 比如你需要写 Python, 就搜 "python"。
3. 装 **Skill**: 复制安装命令, 在 Trae 对话框里运行。
4. 用 **Skill**: 安装好后, 在对话里直接说 "帮我用 xxx skill 做...", 或者 AI 也会自己判断什么时候该用。

### 7.2 常用 Skill 表

Skill	什么时候用	怎么用
frontend-design	界面丑、缺动画	比如设计 ThemeSelector 的下拉菜单, 或者 MessageBubble 的气泡样式。
frontend-code-review	提交代码前	比如查查 src/lib/session.ts 里的 LocalStorage 操作安不安全。
test-generator	缺测试	比如 src/lib/easter-eggs.ts 正则太复杂, 让它自动生成测试用例。