

描述想法，直接生成可部署应用

v0 实战手册

Vercel 出品的 AI 全栈构建器。React + Next.js + Tailwind，
用中文说清楚想要什么，几十秒生成可运行代码，一键部署上
线。本书 8 章原文实战：Design Mode、Git 工作流、
Supabase 全栈、Platform API、生产上线 checklist，含真实
代码 + 踩坑记录。

FREE \$5 免费额度即可上手

⚡ 10 分钟从零到线上

🔑 Supabase 认证 + 数据库一键接

UI 搭建不是瓶颈，v0 替你搞定前端，专注业务逻辑



jiangren.com.au · 在线版随官网更新 /wiki/v0-guide

目录 (8 章)

- 01 🤖 v0 是什么：Vercel 出品的 AI 全栈构建器
- 02 🚀 注册到第一个项目上线
- 03 ⚙️ 核心功能：Design Mode、Git 工作流、数据库连接
- 04 🎯 高手进阶：Prompt 技巧与实战案例
- 05 😊 常见问题：定价、省钱策略与适用场景
- 06 📁 v0 + Supabase 全栈实战
- 07 ⚡ v0 Platform API 与自动化
- 08 ✅ 从 v0 到生产环境



📖 这本手册怎么读

这本书和官网 wiki《v0 实战指南》同源 (jiangren.com.au/wiki/v0-guide, 免费、不用注册, 官网那份持续更新)。完全没接触过 v0 的, 第 1、2 章看清楚它能干什么、10 分钟跑通第一个应用; 想把 UI 做得精的, 第 3 章 Design Mode + shadcn/ui 集成; 想接真实后端的, 第 6 章专门讲 Supabase 全栈 (认证 + 数据库 + 实时订阅); 要在自己工具链里批量用 v0 生成代码的, 第 7 章 Platform API; 准备上线了, 第 8 章安全清单逐项过。书里代码全可复制, 命令直接能跑。



💬 用 v0 卡住了? 扫码进群问

群里每周分享真实的 v0 项目案例和 AI 工具最新玩法。





v0 是什么：Vercel 出品的 AI 全栈构建器

一句话说清楚

v0（原域名 v0.dev，2026 年 1 月迁移到 v0.app）是 Vercel 做的 AI 应用构建平台。你用中文或英文描述想要什么，它直接生成可运行的 React + Next.js 代码，点一下就部署到线上。

2023 年 10 月上线时只能生成 UI 组件，到 2026 年已经进化成全栈应用构建器——能连数据库、加认证、写 API、一键部署。目前有超过 600 万开发者在用。

技术栈

v0 生成的代码固定用这套技术栈，没得选：

```
React + Next.js (App Router)
TypeScript
Tailwind CSS
shadcn/ui (基于 Radix UI 的组件库)
```

这也是它最大的优势和限制——如果你的项目用 Vue 或 Angular，v0 帮不了你。但如果你在 React 生态里，v0 生成的代码质量是同类工具中最高的，拿来直接用不丢人。

shadcn/ui 的作者已经加入 Vercel 团队，所以 v0 对这套组件库的理解比任何竞品都深。生成的组件开箱就有无障碍支持、暗色模式、响应式布局。

跟其他工具怎么选

维度	v0	Lovable	Bolt.new	Cursor
定位	有想法 → 生成应用	非技术人员建 MVP	多框架全栈构建	有代码 → AI 辅助编辑
前端质量	最好，生产级	好	好	取决于你自己
后端能力	有限，需接外部服务	Supabase 集成	内置 Node.js	完整
框架	只有 React/Next.js	React	多框架	任意
部署	一键到 Vercel	内置托管	内置	自己搞
免费额度	\$5/月	有限	较多	无免费

如果你做 React 项目、追求 UI 质量、不介意绑定 Vercel 生态，v0 是目前最优解。它不是万能的，但在“AI 生成前端代码”这个赛道上，确实没有对手。

v0 背后的 AI 模型

v0 不是套壳 GPT-4 或 Claude。它用的是自己训练的复合模型系列：

- v0-1.5-md：基于 Claude Sonnet，128K 上下文，日常够用
- v0-1.5-lg：512K 上下文，处理复杂多步任务更强，但偶尔出错率稍高

加上专门优化的 RAG 检索、Quick Edit 管线和自动修复模型，整体效果比直接用基础模型好不少。

生成的项目结构

v0 生成的代码遵循标准 Next.js App Router 目录结构，拉到本地后跟手写项目没有区别：

```
my-v0-project/
├── app/
│   ├── layout.tsx      # 根布局，全局样式和字体
│   ├── page.tsx        # 首页
│   └── api/
│       └── route.ts     # API 路由（如果有后端逻辑）
├── components/
│   ├── ui/             # shadcn/ui 基础组件 (Button, Card, Dialog...)
│   └── custom-widget.tsx # 业务组件
├── lib/
│   └── utils.ts         # cn() 等工具函数
├── public/              # 静态资源
├── tailwind.config.ts
├── next.config.mjs
└── package.json
```

v0 CLI 快速入门

除了浏览器里用，v0 还提供 CLI 工具，可以在终端里直接跑：

```
# 安装
npm i -g @vercel/v0

# 用 Prompt 生成新项目
v0 generate "一个带暗色模式的 Todo 应用"

# 把 v0 生成的组件拉到现有 Next.js 项目
npx v0 add https://v0.app/chat/xxx
```

CLI 适合已经有本地项目的开发者——在浏览器里调好 UI，一条命令同步到本地 codebase。



注册到第一个项目上线

注册

v0 注册不要钱，不要信用卡。打开 v0.app，用以下任一方式登录：

- GitHub 账号（推荐，后面 Git 集成方便）
- Google 账号
- 邮箱
- 已有的 Vercel 账号

登录后自动获得每月 \$5 免费额度，够跑完这个教程。

点击 "Continue with GitHub" 后，浏览器跳转到 GitHub OAuth 授权页，v0 只申请读取账号基本信息和邮箱，不会要求 repo 写权限（那是后面 Git 集成才需要的）。点 "Authorize" 后秒跳回 v0 主界面，没有额外表单。首次进入会看到空白对话框和几张示例卡片。**v0 账号就是 Vercel 账号**，两边共用一套身份，不用分别管理。

界面布局

区域	功能
左侧聊天区	输入描述、迭代需求
右侧 Preview 标签	实时预览生成的应用
右侧 Code 标签	VS Code 风格编辑器，逐文件查看代码
顶部栏	项目名称、Deploy 按钮、Git 面板
侧边栏	环境变量（Vars）、模板、MCP 集成

创建第一个项目

点击聊天区的 "+" 或直接输入描述。试试这个：

创建一个个人作品集网站，包含：

- 顶部导航栏（首页、项目、关于我、联系方式）
- Hero 区域带我的名字和一句话介绍
- 项目卡片网格，每个卡片有标题、描述、截图、链接
- 底部联系表单
- 暗色主题
- 响应式布局，手机端自动适配

v0 会花 10–30 秒生成完整的 Next.js 应用。右侧 Preview 直接能看效果，Code 标签能看到生成的文件结构。

第一个 React 组件实战

在聊天框输入：

写一个 PriceCard React 组件, 接受 props:

- title (字符串)
 - price (数字)
 - currency (字符串, 默认"AUD")
 - features (字符串数组)
 - highlighted (boolean, 高亮时用不同边框颜色)
- 用 TypeScript + Tailwind CSS。

v0 生成的结果大概长这样:

```
interface PriceCardProps {
  title: string
  price: number
  currency?: string
  features: string[]
  highlighted?: boolean
}

export function PriceCard({
  title,
  price,
  currency = "AUD",
  features,
  highlighted = false,
}: PriceCardProps) {
  return (
    <div
      className={`rounded-2xl border p-6 flex flex-col gap-4 ${
        highlighted
          ? "border-indigo-500 shadow-lg shadow-indigo-500/20"
          : "border-zinc-700"
      }`}
    >
      <h3 className="text-xl font-semibold">{title}</h3>
      <p className="text-4xl font-bold">
        {currency} {price}
        <span className="text-sm font-normal text-zinc-400">/月</span>
      </p>
      <ul className="flex flex-col gap-2 text-sm text-zinc-300">
        {features.map((f) => (
          <li key={f} className="flex items-center gap-2">
            <span className="text-green-400">✓</span> {f}
          </li>
        ))}
      </ul>
    </div>
  )
}
```

几个值得注意的地方: 可选 prop 的默认值在解构时直接写 `= "AUD"`; 条件 className 用模板字符串拼 Tailwind class; `key` prop 是 React 列表渲染必填, v0 不会漏掉。

迭代修改

不满意？直接在聊天里说："把 Hero 区域的背景换成渐变色"、"项目卡片加上悬浮放大效果"。每次修改 v0 都会生成新版本，所有历史版本都保存着，随时回滚。还有个省钱技巧：切换到 **Design Mode**，直接点击页面上的元素修改颜色、字体、间距——这些操作不消耗额度。只有 AI 生成才扣 credits。

部署上线

满意了？点右上角 **Deploy** 按钮：

1. v0 自动创建 Vercel 项目（名称前缀 `v0-`）
2. 几秒钟后给你一个 `xxx.vercel.app` 的线上地址
3. 自带 SSL 证书、全球 CDN、Serverless Functions
4. 后续修改再点 Deploy 就更新同一个项目

与 GitHub + Vercel 联动

如果要正经维护项目，v0 内建的 Git 工作流更合适：

第一步：在 v0 顶部找到 Git 面板，点 "Connect to GitHub"，授权后选择要推送的仓库，可以新建空仓库或连已有的 repo。

第二步：连上 GitHub 后，v0 顶部会出现 "Create Branch" 按钮。点击后把当前这轮对话的所有改动推到一个独立分支，比如 `v0/feat-price-card`。

第三步：在 GitHub 里对 main 开 PR，Vercel 自动为这个 PR 生成预览链接——每次 push 都刷新预览。确认没问题了 merge 进 main，Vercel 触发正式部署，全程不用手动操作。

到这一步，你已经从零到上线了一个网站，全程没写一行代码。



核心功能：Design Mode、Git 工作流、数据库连接

Design Mode：免费的可视化编辑

v0 最被低估的功能。切换到 Design Mode 后，你可以直接点击页面上的任何元素，修改颜色（背景、文字、边框）、字体大小和字重、间距（padding、margin）、文本内容、布局方式。

关键点：Design Mode 的所有操作不消耗 credits。微调样式全在这里做，把 AI 生成留给结构性改动，每月 \$5 额度能多撑好几倍。

Design Mode 底层是直接操作 Tailwind class，改完后点「Apply」，右侧 diff 面板会显示对应的 className 变化。每一次点选都对应着真实的代码改动，合并到项目里不会有任何“魔法属性”残留。

Git 集成：正经的工程化工作流

2026 年 2 月加的 Git 面板让 v0 从玩具变成了正经工具：

```
# v0 在幕后做的事情：
git checkout -b v0/feature-dashboard # 每个聊天自动创建分支
git commit -m "Add KPI cards"        # 每条消息自动 commit
# 永远不会直接推到 main
```

工作流：每个聊天会话自动创建独立分支 → 每条消息生成的改动自动 commit → 通过 Git 面板查看 diff → 直接在 v0 里对 main 开 Pull Request → 合并后自动触发 Vercel 部署。

实战技巧：给 v0 的每个聊天起有意义的名字（比如"dashboard-kpi-cards"），这样 Git 分支名会带上这个名字，review 时一眼知道这个 PR 做了什么。

数据库连接

v0 通过 Vercel Marketplace 支持三个数据库：

数据库	类型	适合场景
Neon	Serverless PostgreSQL	需要 SQL 的常规应用
Supabase	PostgreSQL + Auth + 实时订阅	需要认证和实时功能
Upstash	Redis 兼容 KV 存储	缓存、限流、会话管理

跟 v0 说“帮我加一个数据库”，它引导你从 Marketplace 选择并一键配置，自动生成 schema、API 路由和 CRUD 操作。比如加用户认证，v0 会生成：

```
// app/api/auth/route.ts
import { createClient } from '@supabase/supabase-js'

const supabase = createClient(
  process.env.NEXT_PUBLIC_SUPABASE_URL!,
  process.env.SUPABASE_SERVICE_ROLE_KEY!
)

export async function POST(request: Request) {
  const { email, password } = await request.json()
  const { data, error } = await supabase.auth.signUp({ email, password })
  if (error) return Response.json({ error: error.message }, { status: 400 })
  return Response.json({ user: data.user })
}
```

Figma 导入

两种方式：截图导入（分析布局颜色生成代码）和 Figma 链接导入（Premium 及以上，提取设计 token，保真度更高）。建议把设计拆成小组件逐个生成再组合，别一次丢整个页面。

MCP 集成

v0 支持 Model Context Protocol，可连接 Stripe、Supabase 等外部服务。连接后 v0 生成代码时自动考虑已接入的服务，省去手动配置。

Generative UI 实战：Analytics Dashboard

第一轮提示词：

```
Create an analytics dashboard with:
- 4 KPI cards: Visitors (124,500), Conversion rate (3.2%), Revenue ($48,200), Active users (8,940)
- A line chart showing 30-day visitor trend
- A table showing top 10 pages by pageviews
- Left sidebar navigation with: Dashboard, Analytics, Reports, Settings

Use shadcn/ui components. Dark header, white content area.
```

v0 一次性生成完整的组件树，用到的 shadcn 组件包括：Card、CardHeader、CardContent、Table、TableRow、TableCell，图表部分用 Recharts。

第二步把数据抽出来：

```
Refactor the KPI cards to accept a `metrics` prop.
Extract the table data into a `pageData` array.
Make the chart accept a `chartData` prop instead of hardcoded values.
```

第三步接真实 API：

```
Add a useEffect to fetch KPI data from /api/analytics/metrics
and page data from /api/analytics/pages.
Show skeleton loaders while fetching.
```

v0 加上 Suspense/skeleton 状态，用 shadcn 的 Skeleton 组件占位，让加载体验不再是白屏。Dashboard 这种组件重、样式繁的场景，v0 节省的时间集中在结构搭建和样式对齐，逻辑层仍需人工处理。

credits 消耗对比

操作	消耗 credits
发一条新的生成请求	✔ 是
Design Mode 调整样式	✘ 否
查看历史版本	✘ 否
Fork 别人的 v0 项目	✘ 否
在 v0 里运行代码 (Preview)	✘ 否
重新生成 (Regenerate)	✔ 是
追加修改 (在同一聊天里发新消息)	✔ 是

每月免费额度 (Free 计划) 大约够跑 20-30 次生成。Premium 计划 (\$20/月) 额度显著更高，且支持更复杂的生成 (多文件项目、更长的上下文)。



高手进阶：Prompt 技巧与实战案例

Prompt 三要素框架

v0 的 Prompt 写法直接决定输出质量。Vercel 官方推荐的三要素：

1. **Product Surface（具体界面）**——列出每个组件和数据。别说“做个仪表盘”，要说清楚里面有什么：

创建运营仪表盘：

- 4 个 KPI 卡片 (MRR、活跃用户数、流失率、NPS 分数)
- 趋势折线图，最近 30 天数据
- 最近订单表格，列：订单号、客户、金额、状态
- 日期范围筛选器
- 右上角通知铃铛

2. **Context of Use（使用场景）**——谁在用、什么时候用：“给运营团队每天早上看，一眼看到指标变化，快速筛选异常订单。”

3. **Constraints（约束）**——颜色、风格、“不要什么”：“深色主题，主色调 #6366f1。不用 Tab 切换，所有信息一页。表格按时间倒序。”

Tailwind 精确控制

v0 的底层思维基于 Tailwind CSS，所以直接用 Tailwind 类名比用自然语言描述更精确：

模糊描述	精确 Tailwind 指令
"图片小一点"	"make the image <code>size-4</code> " (精确 1rem × 1rem)
"文字大一点"	"set heading to <code>text-2xl font-semibold</code> "
"加点间距"	"add <code>p-6</code> to the card container"
"数字排列不整齐"	"add <code>tabular-nums</code> to all numeric cells"

迭代策略

- 第一次生成是草稿，别指望一步到位
- 2-3 轮紧凑迭代比超长 Prompt 效果好
- 每轮改 2-3 处，同时改 15 处 v0 会顾此失彼
- 5 轮还不满意就重写 Prompt，别补丁叠补丁
- 样式微调用 Design Mode（不扣钱）

Instructions 系统

v0 支持保存可复用指令，三个层级叠加生效：Account Instructions（全局）→ Project Instructions（项目级）→ Chat Instructions（当前聊天）。

设置路径：点右上角头像 → Settings → Instructions。写好 Instructions 能省大量重复描述。示例：

```
# 账户级 Instructions
- 所有代码用 TypeScript strict mode
- 组件用 named export, 不用 default export
- 颜色变量用 CSS custom properties, 不硬编码 hex
- 中文注释, 英文变量名
- 表单验证用 zod + react-hook-form
```

```
# 项目级 Instructions
- 品牌主色 --brand: #6366f1, 辅色 --accent: #f59e0b
- 字体 Inter, 中文用 Noto Sans SC
- 所有页面带面包屑导航
- API 路由统一返回 { data, error } 格式
```

版本管理与回滚

v0 会为每条消息自动存一个版本快照。点击消息左侧的版本号 (v1, v2, v3...) 可以预览历史版本；点 "Restore" 回滚到该版本；从历史版本 fork 出新分支继续迭代。

一条经验：如果连续 3 轮迭代都在修同一个问题，回滚到出问题前的版本，换个描述方式重来，比继续打补丁效果好。

实战场景效果对比

项目类型	效果	原因
着陆页 / 营销页	极好	最强项，接近专业设计师水准
SaaS 管理后台	很好	表格、图表、表单组件成熟
MVP 原型验证	很好	速度快，投资人演示够用
复杂业务逻辑	一般	前端没问题，后端要自己补
非 React 项目	不行	不支持 Vue/Angular

真实场景：某团队用 v0 生成 CRM 仪表盘原型，3 小时从概念到可交互 demo，客户确认后再用 Cursor 做完整实现。v0 负责快速试错，Cursor 负责正式开发。

读到这章，你大概已经想到几个自己想做的东西——把 v0 真的用起来，第一个原型往往比你想的快。

Amelia

IT Career Consultant

在澳洲 IT 业务咨询领域拥有10年丰富经验和专业知识。对 IT 行业的发展趋势和技术需求有着深入的了解，擅长提供转码就业问题解决方案，能够准确把握最新的技术趋势和就业需求，为您提供最具前瞻性的建议。

探索IT领域，点亮你的科技未来

提供问题解决方案

工作内推机会

IT就业群提供

IT技术提升服务

300+

已帮助300+人成功就业

1000+

帮助1000+学员答疑解惑

200+

已帮助数百人成功转码



想用 AI 工具做出真实产品？

匠人学院 AI 工程师方向，从 v0、Cursor 到完整项目落地，有人带着做比自己摸索快很多。

[了解课程 →](#)



常见问题：定价、省钱策略与适用场景

定价一览

v0 在 2026 年初改成 token 计费制：

计划	月费	每月 Credits	核心差异
Free	\$0	\$5	基础模型、最多 200 个项目、用完等下月
Premium	\$20	\$20	高级模型、Figma 导入、v0 API、可加购
Team	\$30/人	\$30/人	团队协作、共享额度
Enterprise	按需	按需	SSO、专属支持

Credits 按三级模型定价：

```
Mini 模型： 输入 $1/百万 token | 输出 $5/百万 token ← 最省钱
Pro 模型：  输入 $3/百万 token | 输出 $15/百万 token ← 日常推荐
Max 模型：  输入 $5/百万 token | 输出 $25/百万 token ← 复杂任务用
```

省钱攻略

免费 \$5 说实话不多，一个复杂项目一下午就能烧完。几个实测有效的策略：

1. Design Mode 能做的别用 AI——改颜色、改文字、调间距全在 Design Mode 里搞，零消耗
2. 简单任务用 Mini 模型——改个按钮颜色不需要 Max 模型
3. Prompt 写得越具体、迭代越少——模糊 prompt 导致反复修改，每轮都扣钱
4. 拆小项目——一个大需求拆成多个小聊天，每个聊天上下文短，token 消耗低
5. 及时停手——如果 v0 在某个 bug 上转圈修不好，手动去 Code 编辑器里改，比让 AI 循环消耗 credits 划算

常见问题

Q: v0 支持中文 Prompt 吗？

支持，v0 会自动用你提问的语言回复，包括代码注释也会写中文。但体感上英文 Prompt 的生成质量稍好一些，建议技术描述用英文，需求描述可以用中文。

Q: 生成的代码能直接用到自己项目吗？

能。v0 生成的是标准 Next.js + shadcn/ui 代码，跟手写的没区别。可以直接 `npx shadcn@latest add` 到现有项目，或者通过 Git 集成提 PR 合并。

Q: 安全性怎么样？

AI 生成的代码可能有安全隐患，v0 也不例外。认证、支付、敏感数据处理必须人工审查。

Q: 被 Vercel 绑定了怎么办？

代码是标准 Next.js，能部署到任何 Node.js 平台。迁移成本不大。

Q: v0 跟 Cursor 冲突吗？

不冲突，互补。典型工作流：v0 生成初始 UI → Git 拉到本地 → Cursor 做深度开发：

```
# 1. v0 生成初始代码，通过 Git 面板推到 GitHub
# 2. 本地拉取
git clone https://github.com/you/v0-project.git
cd v0-project && npm install

# 3. 用 Cursor 打开，做深度开发
cursor .

# 4. 加业务逻辑、测试、CI
npm run test && npm run build

# 5. 推到 GitHub, Vercel 自动部署
git push origin main
```

Q: 环境变量怎么管理？

在 v0 聊天侧边栏点 Vars 面板，添加的环境变量自动同步到 Vercel 项目。敏感信息（API Key、数据库密码）放这里，不要写在代码里。

```
// v0 生成的代码会自动引用环境变量
const apiKey = process.env.OPENAI_API_KEY!
const dbUrl = process.env.DATABASE_URL!

// 本地开发时，在 .env.local 里配置
// OPENAI_API_KEY=sk-xxx
// DATABASE_URL=postgresql://...
```

谁该用 v0

你是...	推荐度	理由
React 前端开发者	强烈推荐	提速 3-5 倍，代码质量高
产品经理	推荐	自己做可交互原型
Vue/Angular 开发者	不推荐	只支持 React

v0 是目前 AI 生成前端代码质量最高的工具。\$0 注册试一下，10 分钟就能感受到它的威力。



 **实战遇到问题？扫码加群**

AI 工具实操群，每周有人分享 v0 + Cursor + Next.js 的真实项目踩坑记录。



v0 + Supabase 全栈实战

为什么选 Supabase

v0 通过 Vercel Marketplace 支持三个数据库，Supabase 是功能最全的——Postgres 数据库 + Auth 认证 + Realtime 订阅 + Storage 文件存储，一个服务全包。免费额度够跑 MVP。

一键接入

在 v0 聊天侧边栏点 Connect → Supabase → Create。第一次用会跳到 Vercel Marketplace 授权，同意后自动：

1. 创建 Supabase 项目（选离你最近的 region）
2. 把 `SUPABASE_URL` 和 `SUPABASE_ANON_KEY` 注入 v0 环境变量
3. v0 后续生成代码时自动使用这些变量

不需要手动复制粘贴任何 key。

建表 + CRUD

跟 v0 说"创建一个任务管理应用，用 Supabase 存数据"，它会自动生成建表 SQL 和完整的 CRUD 逻辑：

```
-- v0 自动生成的 Supabase SQL (在 Supabase Dashboard → SQL Editor 执行)
create table tasks (
  id uuid default gen_random_uuid() primary key,
  title text not null,
  completed boolean default false,
  user_id uuid references auth.users(id),
  created_at timestamptz default now()
);

-- 开启 Row Level Security
alter table tasks enable row level security;

-- 只允许用户访问自己的数据
create policy "Users can CRUD own tasks"
  on tasks for all
  using (auth.uid() = user_id);
```

v0 生成的前端代码会直接对接这张表：

```

// lib/supabase.ts - v0 自动生成
import { createClient } from '@supabase/supabase-js'

export const supabase = createClient(
  process.env.NEXT_PUBLIC_SUPABASE_URL!,
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!
)

// 查询任务列表
const { data: tasks } = await supabase
  .from('tasks')
  .select('*')
  .order('created_at', { ascending: false })

// 新增任务
await supabase.from('tasks').insert({ title: '写周报', user_id: session.user.id })

// 标记完成
await supabase.from('tasks').update({ completed: true }).eq('id', taskId)

```

加用户认证

跟 v0 说"加上邮箱登录和 Google 登录", 它生成完整的认证流程:

```

// app/login/page.tsx 核心逻辑
import { supabase } from '@lib/supabase'

// 邮箱注册
await supabase.auth.signUp({ email, password })

// 邮箱登录
await supabase.auth.signInWithPassword({ email, password })

// Google OAuth
await supabase.auth.signInWithOAuth({
  provider: 'google',
  options: { redirectTo: `${window.location.origin}/auth/callback` }
})

// 登出
await supabase.auth.signOut()

```

Google OAuth 需要在 Supabase Dashboard → Authentication → Providers 里配置 Client ID 和 Secret (从 Google Cloud Console 获取)。

实时订阅

Supabase Realtime 让多个用户同时看到数据变化, 适合协作类应用:

```
// 监听 tasks 表的实时变更
supabase
  .channel('tasks')
  .on('postgres_changes', { event: '*', schema: 'public', table: 'tasks' },
    (payload) => {
      if (payload.eventType === 'INSERT') setTasks(prev => [payload.new, ...prev])
      if (payload.eventType === 'UPDATE') {
        setTasks(prev => prev.map(t => t.id === payload.new.id ? payload.new : t))
      }
    }
  )
  .subscribe()
```

实际用下来，v0 + Supabase 能在 30 分钟内搭出一个带认证、数据库、实时同步的完整应用——以前这套至少要两天。

能把前端 + 数据库 + 认证在一天内做出来的，是会用工具的人，不是会写代码的人。

Rain

Senior IT Career Consultant

在澳洲近20年，Master of HRM，15年企业HR及猎头招聘工作经验，熟悉中国及澳洲就业市场情况。尤其在IT领域中拥有超过6年的咨询及招聘经验。

—— 是你拿下offer的最佳辅助！

毕业生求职做规划

辅助在职人士跳槽晋升

已帮助数千余人成功就业

免费简历诊断

3000+

3000+咨询案例

1200+

1200+辅助成功
斩获offer

1000+

1000+简历诊断

 <http://linkedin.com/in/rainqiu>



想系统学 AI 全栈开发？

匠人学院 AI 工程师方向，v0 + Supabase + Vercel 实战项目一条线，从会用到能落地。

[了解课程 →](#)



v0 Platform API 与自动化

Platform API 是什么

v0 Platform API 把浏览器里的“对话生成代码”能力开放成了 REST 接口。你可以用代码调 v0，批量生成组件、自动迭代、把 v0 嵌入自己的工具链。需要 Premium 计划（\$20/月）才能用 API，免费版不行。

获取 API Key

1. 打开 `v0.app/chat/settings/keys`
2. 点 Create API Key
3. 复制 key，存到环境变量里，不要提交到代码仓库

```
# .env.local
V0_API_KEY=v0_key_XXXXXXXXXXXXXXXXXXXX
```

SDK 安装与基础用法

```
npm install @anthropic-ai/v0-sdk
# 或
pnpm add @anthropic-ai/v0-sdk
```

最基本的用法——用 Prompt 生成一个 React 组件：

```
import { V0Client } from '@anthropic-ai/v0-sdk'

const v0 = new V0Client({ apiKey: process.env.V0_API_KEY })

// 创建聊天会话
const chat = await v0.chats.create({
  prompt: '创建一个带搜索和分页的用户列表组件',
  model: 'v0-1.5-md'
})

// 获取生成结果
console.log(chat.id)           // 聊天 ID
console.log(chat.result)       // 生成的代码
console.log(chat.files)        // 文件列表
```

多轮迭代

```
// 继续迭代
const updated = await v0.chats.update(chat.id, {
  prompt: '表格加上排序功能, 点击列头切换升序降序'
})

// 再次迭代
const final = await v0.chats.update(chat.id, {
  prompt: '加一个导出 CSV 按钮, 放在表格右上角'
})

// 获取最终的文件列表
for (const file of final.files) {
  console.log(`${file.path}: ${file.content.length} chars`)
}
```

批量生成组件

```
const pages = [
  '用户登录页, 带邮箱和 Google OAuth',
  '仪表盘, 4 个 KPI 卡片 + 趋势图',
  '用户管理列表, 支持搜索、筛选、批量操作',
  '设置页, 个人信息 + 通知偏好 + 安全设置',
  '404 页面, 带搜索框和热门链接'
]

const results = await Promise.all(
  pages.map(prompt =>
    v0.chats.create({
      prompt,
      instructions: '使用 shadcn/ui, 品牌色 #6366f1, 暗色主题'
    })
  )
)

// 把生成的文件写入本地项目
for (const result of results) {
  for (const file of result.files) {
    await fs.writeFile(`src/${file.path}`, file.content)
  }
}
```

5 个页面并行生成, 2 分钟搞定, 手写至少两天。

集成 CI/CD

把 v0 API 接入 GitHub Actions, 实现 PR 里自动生成 UI 预览:

```
# .github/workflows/v0-preview.yml
name: v0 UI Preview
on:
  pull_request:
    paths: ['docs/ui-specs/*.md']

jobs:
  generate:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - run: npm install @anthropic-ai/v0-sdk
      - run: node scripts/generate-ui.js
      env:
        V0_API_KEY: ${ secrets.V0_API_KEY }
      - uses: actions/upload-artifact@v4
      with:
        name: generated-ui
        path: generated/
```

PM 在 docs/ui-specs/ 里写需求文档，推 PR 后自动生成 UI 代码供开发 review——这是 v0 API 最实用的场景之一。





✓ 从 v0 到生产环境

v0 生成 ≠ 生产就绪

v0 能在 10 分钟生成一个能跑的应用，但"能跑"和"能上线"之间还有几步要走。AI 生成的代码可能有安全漏洞、性能问题、硬编码的测试数据。

安全审查

v0 生成的代码经常在这些地方留坑，逐项检查：

```
// ✗ v0 经常生成的问题代码
const res = await fetch(`/api/user/${userId}`) // 没有权限校验

// ✓ 加上认证检查
import { auth } from '@/lib/auth'

export async function GET(req: Request, { params }: { params: { id: string } }) {
  const session = await auth()
  if (!session) return Response.json({ error: 'Unauthorized' }, { status: 401 })
  if (session.user.id !== params.id) return Response.json({ error: 'Forbidden' }, { status: 403 })
  // ...
}
```

项目	检查内容
API 路由	每个路由都有认证和权限检查
环境变量	敏感 key 不在客户端代码里暴露（不带 NEXT_PUBLIC_ 前缀）
SQL 注入	用 Supabase SDK 或 Prisma，不拼接 SQL 字符串
XSS	dangerouslySetInnerHTML 的内容经过消毒处理
CORS	API 路由只允许你的域名

性能优化

```

// 1. 图片优化—用 next/image 替换 <img>
import Image from 'next/image'
// <Image src="/hero.png" width={1200} height={600} alt="Hero" priority />

// 2. 动态导入—大组件按需加载
import dynamic from 'next/dynamic'
const Chart = dynamic(() => import('@components/chart'), { ssr: false })

// 3. 数据缓存—频繁请求的数据加 revalidate
export const revalidate = 3600 // 1 小时缓存

// 4. Metadata—SEO 必备
export const metadata = {
  title: '我的应用',
  description: '用 v0 构建的全栈应用',
  openGraph: { images: ['/og.png'] }
}
```

```

# 本地构建后检查 bundle 大小
npm run build
# 关注 First Load JS 列，单个路由不要超过 200KB
```

自定义域名

v0 部署后默认给 `xxx.vercel.app` 域名。绑定自己的域名：

1. 在 v0 界面点 Deploy → Settings → Domains
2. 添加你的域名，比如 `app.yourcompany.com`
3. 在域名注册商（Cloudflare、阿里云等）添加 CNAME 记录指向 `cname.vercel-dns.com`
4. Vercel 自动申请 SSL 证书，几分钟后生效

```

# 验证 DNS 解析
dig app.yourcompany.com CNAME
# 应该返回 cname.vercel-dns.com

# 验证 SSL
curl -I https://app.yourcompany.com
# 应该返回 HTTP/2 200
```

监控与日志

```

// next.config.mjs - 开启 Vercel Analytics
/** @type {import('next').NextConfig} */
const nextConfig = {
  logging: { fetches: { fullUrl: true } }
}
export default nextConfig

```

```

# Vercel CLI 查看生产日志
npx vercel logs --follow

# 查看最近的部署状态
npx vercel ls

```

在 Vercel Dashboard → Logs 可以实时看到 Serverless Function 的请求日志和错误。建议接入 Sentry 或 LogRocket 做错误追踪——v0 生成的代码不会自带这些。

上线 Checklist

- ☐ 所有 API 路由有认证检查
- ☐ 环境变量不暴露到客户端
- ☐ 图片用 next/image 优化
- ☐ Lighthouse Performance > 85
- ☐ 自定义域名 + SSL 生效
- ☐ 错误监控已接入
- ☐ 备份数据库 (Supabase Dashboard → Backups)
- ☐ 去掉 v0 生成的 placeholder 数据和 Lorem ipsum



用 v0 做出来的东西，怎么写进简历？

这本手册的在线版在 jiangren.com.au/wiki/v0-guide，随官网更新。匠人学院还有 20+ 本同款 AI 工具指南（Claude Code、Cursor、Dify、CrewAI、n8n...）和 AI Engineer 方向的项目制课程——想知道 v0 生成的前端怎么接真实业务、做成找工作的作品，来聊聊。



Angela

IT Career Consultant

ESFJ型，10年澳洲留学工作经验，熟悉中国及澳洲就业市场情况，擅长为毕业生求职做规划，熟悉澳洲IT各方向就业情况。

—— 我们不只是指引，更是帮你创造机会

毕业生求职规划

擅长帮助零基础转码

就业市场咨询

帮助数千余人成功就业

免费简历诊断

专业求职建议

1000+

已帮助数千余人成功就业

200+

已帮助数百余人成功转码

300+

已帮助数百余人简历诊断

 <https://www.linkedin.com/in/angela-han-7212892b4/>



扫码进社群，v0 + AI 工具答疑

把这本 PDF 转给还在手写前端样板代码的同事。



jiangren.com.au