

# 本地部署

## 第一步：准备环境 (Node.js)

1. 访问 [Node.js 官网](#)。
2. 下载 **LTS** 版本。
3. 按照安装向导完成安装。
4. **验证安装**：打开您的终端（Windows 下是 CMD 或 PowerShell，Mac/Linux 下是 Terminal），输入以下命令并回车：

```
node -v  
npm -v
```

如果您看到了版本号说明安装成功。

---

## 第二步：获取并解压代码

1. **下载代码**：从 AI Studio 下载了 `.zip` 压缩包。
2. **解压**：将压缩包解压到一个容易找到的文件夹中，例如 `Desktop/my-ai-app`。
3. **进入目录**：
  - 打开您的终端/命令行工具。
  - 使用 `cd` 命令进入该文件夹。例如（Windows）：

Bash

```
cd Desktop\my-ai-app
```

- 小技巧：在 Windows 上，您可以直接打开该文件夹，然后在地址栏输入 `cmd` 并回车，终端会自动在该路径下打开。

---

## 第三步：安装依赖项

在终端中，确保您位于项目文件夹内，输入以下命令。会自动读取项目中的 `package.json` 文件，并自动下载项目运行所需的所有第三方库。



```
npm install
```

## 第四步：配置 API 密钥

项目需要连接 Google 的服务器，所以需要 API Key。

### 1. 获取密钥：

- 访问 [Google AI Studio API Key 页面](#)。
- 点击 "Create API key"。
- 复制生成的以 `Alza` 开头的长字符串。

### 2. 创建环境变量文件：

- 回到项目文件夹。查看是否有 `.env.local` 文件。如果没有则需要创建一个。Win 右键新建文本文档，命名为 `.env.local`（**注意：** 确保没有 `.txt` 后缀。如果系统隐藏了后缀名，可能需要先开启“显示文件扩展名”）。或者，如果项目里有一个叫 `.env.example` 的文件，您可以将它复制一份并重命名为 `.env.local`。

### 3. 填入密钥：

- 用记事本打开 `.env.local`。输入以下内容（将 `您的_API_KEY_粘贴在这里` 替换为您刚才复制的 `Alza...` 那个字符串）：

```
GEMINI_API_KEY=您的_API_KEY_粘贴在这里
```

- 保存并关闭文件。

## 第五步：运行应用程序

一切准备就绪，在终端中输入：



```
npm run dev
```

- **注意：** 终端会显示一些启动信息，最后通常会提示：

```
□ Ready in Xms  
Listening on http://localhost:3000 (或其他端口号)
```

## 第六步：在浏览器中查看

1. 打开浏览器（Chrome, Edge 等）。
  2. 在地址栏输入：`http://localhost:3000`
  3. 现在应该能看到并使用 AI 应用程序了
- 

## 常见问题排查

- 报错 `'npm' 不是内部或外部命令`：说明 Node.js 没安装好，或者安装后没有重启终端。请重新安装并重启电脑尝试。
- 报错 `EADDRINUSE: address already in use`：说明端口（如 3000）被占用了。您可以尝试关闭其他正在运行的开发程序，或者查看 `package.json` 修改启动端口。
- 应用报错 `"API Key invalid"`：请检查 `.env.local` 文件名是否正确（不要叫 `.env.local.txt`），以及 Key 是否复制完整。

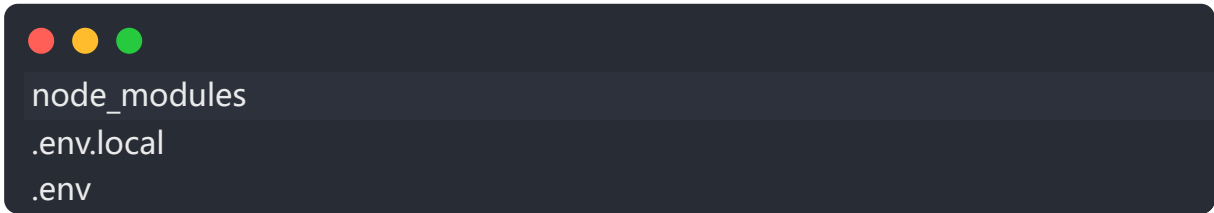
# Vercel 部署

## 第一步：准备与安全检查（至关重要）

在上传代码之前，必须确保**不要**把您的密钥文件传到公开的 GitHub 上。

### 1. 检查 `.gitignore` 文件：

- 在项目根目录下，找找看有没有一个叫 `.gitignore` 的文件（如果没有，请新建一个，名字就是 `.gitignore`）。
- 用记事本打开它，确保里面有以下两行内容（如果没有就加上）：



```
node_modules
.env.local
.env
```

- **解释：**这告诉 Git 工具：“请忽略这些文件”。这样您的 API Key (`.env.local`) 就不会被上传到 GitHub，只有自己电脑上和 Vercel 后台才有。

## 第二布：将代码上传到 GitHub

没经验的推荐Github Desktop，会的就命令行Git解决。

## 第三步：在 Vercel 上操作

### 1. 登录与导入：

- 打开 [Vercel 官网](#) 并登录，推荐直接用 GitHub 账号登录。
- 在控制台点击 **"Add New..." -> "Project"**。
- 在左侧列表里找到您刚才上传的 GitHub 仓库，点击 **"Import"**。

### 2. 配置项目（关键步骤）：

- Vercel 会自动识别这是 Next.js 项目，大部分设置保持默认即可。
- **重点来了：**找到 **Environment Variables** 这一栏，点击展开。
- 需要手动填入密钥，因为刚才 `.gitignore` 把它拦截了：
  - **Key** (名称): 填入 `GEMINI_API_KEY` (必须和代码里用的一模一样)。
  - **Value** (值): 填入您的 `Alza...` 开头的那个长字符串。
  - 点击 **Add** 按钮。

### 3. 开始部署：

- 点击底部的 "**Deploy**" 按钮。
- 屏幕上会出现构建日志，通常等待 1-2 分钟。

## 第四步：大功告成

- 当屏幕上出现全屏的彩带庆祝动画时，说明部署成功了！
  - 点击预览图或 "**Visit**" 按钮，您会进入一个类似 `https://your-project.vercel.app` 的网址。正常来说网页可以直接运行。
  - 同时网页支持直接分享到public。
- 

### 常见错误自查：

- **部署失败 (Build Failed)**：通常是因为代码里有 ESLint 错误。如果遇到，可以在 Vercel 项目设置 -> General -> Build & Development Settings 里，把 "Build Command" 改为 `next build --no-lint` 临时绕过。
- **AI 不回复 (500 Error)**：这通常是 API Key 没填对。去 Vercel 的 Settings -> Environment Variables 检查一下，确保名字是 `GEMINI_API_KEY` 且值没有多余空格。