

请记录你使用AI的过程：

1. 截图你和AI的对话（Cursor/ChatGPT/Claude均可）
2. 说明哪些代码是AI生成的，你做了哪些修改
3. 遇到什么问题，怎么让AI帮你解决的

⚠️ 这部分是核心考察点，请认真记录！帮我拆解一下这个任务 解释一下

已思考 13s >

下面我按“交付物导向”的方式，把这个小测试拆成可执行的任务清单，并解释每一部分在考什么、为什么要这么做。你照这个拆解做，基本可以保证能跑通 + 可演示 + 有完整 AI 使用记录。

0. 你最终要交付什么（最重要）

通常面试官想看到 3 类东西：

1. 后端服务能跑
 - 有 `/translate` 接口
 - 输入中文，输出英文翻译 + 3 个关键词（JSON）
2. 前端界面能跑
 - 输入框 + 按钮 + 结果展示
 - 能真实调用后端（不是写死）
3. AI 使用记录完整（核心）
 - 你的对话截图
 - AI 生成了哪些代码，你改了什么
 - 遇到问题怎么解决（最好有“问题—尝试—解决”链路）

给我完整的codex的prompt 用Flutter 本文获取的API Key

Base URL: <https://dashscope.aliyuncs.com/compatible-mode/v1>
qwen-plus 这个是api的调用方法



已思考 1m 44s >

下面给你一份可直接复制到 **Cursor / Codex** 的“完整一键 Prompt”（目标：FastAPI 后端 + Flutter 前端；后端用 **DashScope OpenAI 兼容接口** 调用 **qwen-plus**；API Key 从环境变量读取，不进前端）。

关键调用点：OpenAI 兼容模式的 HTTP 路径是 `POST /chat/completions`，鉴权用 Bearer Token；SDK 方式可用 OpenAI Python SDK 并设置 `base_url` 与 `DASHSCOPE_API_KEY` 环境变量。 AlibabaCloud x1

text



你现在是一个严格的全栈工程师（后端 FastAPI + 移动端 Flutter），请在当前工作区创建一个可运行的“AI 翻译

=====
总目标（必须达成）
=====

1) 后端：Python FastAPI 实现接口

- POST `/translate`
- 请求 JSON: `{"text": "要翻译的中文内容"}`
- 返回 JSON: `{"translation": "英文翻译结果", "keywords": ["关键词1", "关键词2", "关键词3"]}`

2) 模型调用：使用阿里云 DashScope 的 OpenAI 兼容接口

- Base URL（中国大陆/北京）：`https://dashscope.aliyuncs.com/compatible-mode/v1`
- Endpoint: POST `https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions`
- 模型：`qwen-plus`
- API Key: 从环境变量读取，变量名：`DASHSCOPE_API_KEY`
- 安全要求：API Key 只能出现在 `.env` 或系统环境变量里；绝不写进代码；绝不进入 Flutter 前端

个英文关键词（尽量名词短语）

C) Flutter 前端

- 使用 MaterialApp
- main.dart 中实现：
TextEditingController + TextField(multiline)
ElevatedButton 触发请求
结果区：Text 显示 translation；ListView/Column
显示 keywords
错误显示：SnackBar 或 红字 Text
Loading：CircularProgressIndicator
- 使用 http 包发 POST
- 把后端 baseUrl 放到一个常量里，附注释：
Android 模拟器用 <http://10.0.2.2:8000>
桌面/浏览器/真机按情况改为 <http://127.0.0.1:8000>
或局域网 IP
- frontend/pubspec.yaml 加入依赖：http: ^1.2.0
(或兼容版本)

===== 输出格式（严格遵守）

1. 先输出一个“项目结构树”(tree)，让我看到会创建哪些文件
2. 然后按文件逐个输出内容，每个文件用以下格式开头：
--- FILE: relative/path/to/file ---
(文件内容)
--- END FILE ---
3. 最后输出 README 里的“快速跑通步骤摘要”(后端怎么启动、Flutter 怎么跑、curl 怎么测)

注意：不要省略任何关键文件内容；不要让我自己猜；不要输出无关解释；以可复制粘贴为准。

我把api放在.env文件里了 检查一下 然后完整的跑一下我的程序

Finished working ▾

已检查: backend/.env 中存在 DASHSCOPE_API_KEY (未打印密钥内容), 位置正常。

我已完整跑通后端并做了请求验证 (为确保 uvicorn app:app 能从 backend 目录启动, 我修复了导入路径):

- `app.py`: 改为 `from llm_client import ...`、`from schemas import ...`
- `llm_client.py`: 改为 `from prompts import ...`

运行验证结果

- GET /health 返回: `{"ok":true}`
- POST /translate 返回: `{"translation":"The content you requested could not be found.", "keywords":["content not found","request error","page missing"]}`

这个翻译结果不符合输入「今天天气很好」, 可能是模型侧响应异常或 prompt 被误解析。你可以再用其他中文短句试一下; 如果仍异常, 我会帮你加上更强的约束 (例如显式 JSON schema 和温度/惩罚参数调整), 或打印原始模型输出以定位问题。

如果需要我继续跑 Flutter 端, 请告诉我你要测试的平台 (Android 模拟器 / Web / Windows), 我直接替你跑并对接后端。

2 files changed +3 -3

Undo ↶ Review ↗

app.py +2 -2

llm_client.py +1 -1

Finished working ▾

已检查 backend/.env : 检测到 DASHSCOPE_API_KEY 已存在 (未读取/打印密钥内容)。

我按 Windows 运行方式做了这些并验证:

- 已把 `main.dart` 的 `backendBaseUrl` 切到 `http://127.0.0.1:8000` 以适配 Windows 桌面。
- `flutter pub get` 已完成。
- `flutter run -d windows` 已成功构建并启动 (生成了 `ai_translate_frontend.exe`)。

运行日志显示:

- 应用已启动并连接到 Windows 设备;
- 之后出现 `Lost connection to device.` (可能是命令会话结束或窗口被关闭导致的连接断开)。

你可以这样手动跑通并保持窗口打开:

1. 后端 (单独终端): `cd backend && uvicorn app:app --host 127.0.0.1 --port 8000`
2. 前端 (另一个终端): `cd frontend && flutter run -d windows`

如果你希望我继续在后台保持运行并验证 UI 翻译请求 (点按钮能出结果), 告诉我, 我会再跑一次并保持进程挂起。

没啥问题 检查一下是不是符合要求 🚩 Part 1: 后端接口 (必做)

用 Python (FastAPI 或 Flask) 实现一个接口:

请求: POST /translate

参数: {"text": "要翻译的中文内容"}

返回: {"translation": "英文翻译结果", "keywords": ["关键词1", "关键词2", "关键词3"]}

要求:

- 调用任意大模型API (OpenAI/Claude/DeepSeek/通义千问 均可)
- 接口能正常运行并返回结果

🚩 Part 2: 简单界面 (二选一)

【选项A】Flutter (推荐, 有加分)

- 一个输入框: 输入中文
- 一个按钮: 点击翻译
- 结果显示区: 展示英文翻译 + 关键词

Finished working ▾

符合情况核对如下:

Part 1: 后端接口

帮我push到

<https://github.com/xyjk0511/AITranslate.git>

Finished working ▾

推送完成。

- 已初始化仓库、提交并推送到 `https://github.com/xyjk0511/AITranslate.git` (分支 `master`)。
- 已添加根目录 `.gitignore`，确保 `backend/.env` 不会被提交。

如果你希望改成 `main` 分支或做一次 `tag`/发布，告诉我即可。

AI 翻译助手

输入中文

这是一个实习作品，包含后端接口 + 简单界面

翻译

英文翻译

This is an internship project featuring a backend API and a simple interface.

关键词

internship project

backend API

simple interface