# SafeTrack 应用技术栈说明（中文版）

## 一、📱 前端技术栈（移动端 Flutter）

• Flutter SDK：构建跨平台原生应用（Android/iOS） —— 快速开发、界面统一

• provider / riverpod：状态管理 —— 控制健康数据、警报状态

• sqflite：本地数据库 —— 存储健康历史、阈值等信息

• shared\_preferences：本地键值存储 —— 保存设置，如用户偏好或阈值

• http / dio：网络请求 —— 与 FastAPI 或第三方 API 通信

• fl\_chart：图表显示 —— 展示健康趋势（Should have）

• google\_fit：访问心率、步数等数据 —— Android 专用（Could have）

• flutter\_local\_notifications：本地通知提醒 —— 超阈值警报（Should have）

## 二、🔙 后端技术栈（Python + FastAPI）

• FastAPI：创建 RESTful API 服务 —— 响应健康数据、警报逻辑等

• Uvicorn：FastAPI 的高性能运行环境 —— 启动本地后端服务

• Pydantic：定义数据模型 —— 确保输入/输出数据结构安全

• CORS Middleware：接收 Flutter 请求 —— 允许跨源调用 API 接口

• JWT（可选）：用户身份验证 —— 若启用登录功能

## 三、🗂 数据库

• SQLite：本地轻量级数据库 —— 前后端均可用，适合原型开发

• sqflite（Flutter 插件）：操作 SQLite —— 存储历史记录、阈值

## 四、🌐 第三方 API 集成

• OpenWeather API：获取温度、湿度、空气质量 —— ✅ 应该具备 | 展示环境健康背景信息

• Google Fit API：获取真实健康数据（步数、心率） —— ⚠ 可以具备 | Android 专属，开发略复杂

• Firebase API：云端存储与账户管理 —— ❌ 暂不具备 | 当前不支持云服务

• NHS API (UK)：获取健康症状和疾病信息 —— ❌ 暂不具备 | 留作未来版本扩展

## 五、📁 推荐项目结构（Flutter）

/lib

┣ /screens # 主界面、输入界面、警报界面等

┣ /models # 健康数据、天气数据、阈值等模型

┣ /services # 本地数据库服务、API 请求服务

┣ /providers # 状态管理模块

┣ /widgets # 可复用的 UI 组件

┗ main.dart # 应用入口

## 六、⚙ 开发工具建议

• Postman：测试 API

• Git + GitHub：版本控制

• VS Code 或 Android Studio：Flutter 开发环境

• Render / Railway（可选）：部署 FastAPI 后端以供测试