

SkyTrip 部署说明 (前端 + 后端 + 数据库)

本指南用于在新的主机上部署 SkyTrip 的完整系统，包括：

- 前端 (Vite + React)
- 后端 (FastAPI + Uvicorn)
- 数据库 (MySQL，使用提供的 `skytrip.sql` 数据和结构)

1. 环境准备

- 操作系统：Windows 10/11 或 Linux (Ubuntu 20.04+)
- 数据库：MySQL 8.0 (推荐使用 `utf8mb4`)
- 后端运行环境：
 - Python 3.10+ (建议 3.11)
 - pip、virtualenv/venv
- 前端运行环境：
 - Node.js 18+ (Vite 5 要求)
 - npm 9+ 或 pnpm/yarn (任选其一)

2. 代码结构

- `backend/`：后端 API (FastAPI)
- `fronted/wing-quest-suite/`：前端 (Vite + React + shadcn-ui)
- `skytrip.sql`：完整数据库结构与数据 (SQL Dump)
 - 如未在根目录看到该文件，可使用 `backend/skytrip.sql`

3. 数据库部署 (迁移 `skytrip.sql`)

目标：在新主机的 MySQL 8.0 中恢复 SkyTrip 的数据库。

3.1 在 MySQL 中创建数据库与用户

- Windows PowerShell 或 Linux Shell：

```
mysql -u root -p -e "CREATE DATABASE IF NOT EXISTS skytrip CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;"  
mysql -u root -p -e "CREATE USER IF NOT EXISTS 'skytrip'@'%' IDENTIFIED BY 'your_password';"  
mysql -u root -p -e "GRANT ALL PRIVILEGES ON skytrip.* TO 'skytrip'@'%'; FLUSH PRIVILEGES;"
```

- 如果目标环境不是 MySQL 8.0，且不支持 `utf8mb4_0900_ai_ci`，可使用 `utf8mb4_general_ci`：

```
CREATE DATABASE skytrip CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

3.2 导入 SQL 数据

- Windows (示例路径) : `c:\users\Lenovo\Desktop\skytrip\skytrip.sql`

```
mysql -u root -p skytrip < "c:\users\Lenovo\Desktop\skytrip\skytrip.sql"
```

- Linux (示例路径) : `/opt/skytrip/skytrip.sql`

```
mysql -u root -p skytrip < /opt/skytrip/skytrip.sql
```

- 使用 MySQL Workbench 导入:

- 打开 Workbench → 选择目标连接 → 打开 `Server > Data Import`
 - 选择“Import from Self-Contained File”，指向 `skytrip.sql`
 - 目标数据库选择 `skytrip` → 点击 Start Import

3.3 验证导入

- 登录 MySQL，检查核心表是否存在 (如 `airlines`、`airports`、`routes`、`flights`、`orders`、`order_items`) :

```
mysql -u root -p -e "USE skytrip; SHOW TABLES;"
```

- 该 SQL 中包含 `alembic_version` 表，用于记录迁移版本。如需确保结构与当前代码一致，可在后端执行一次迁移同步 (见后端章节)。

4. 后端部署 (FastAPI)

后端默认使用 MySQL，连接字符串由 `.env` 或 `app/core/config.py` 提供。

4.1 安装依赖

```
cd backend
python -m venv .venv
.venv\scripts\activate          # Windows
# 或: source .venv/bin/activate # Linux/Mac
pip install -r requirements.txt
```

4.2 配置数据库连接

- 推荐在 `backend/.env` 中设置连接 (后端已支持从 `.env` 读取) :

```
SQLALCHEMY_DATABASE_URI=mysql+pymysql://skytrip:your_password@<DB_HOST>:3306/skytrip
SECRET_KEY=replace_with_a_strong_key
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=11520
AI_BASE_URL=
AI_API_KEY=
AI_MODEL=
AI_TIMEOUT_SECONDS=20
```

- 默认代码位置: `backend/app/core/config.py:9` (如未提供 `.env`, 将使用此处的默认连接字符串)

4.3 数据库迁移 (可选)

- 导入了 `skytrip.sql` 后, 一般无需再次初始化。但若代码模型更新, 需要同步到数据库:

```
alembic upgrade head
```

4.4 启动后端

```
uvicorn app.main:app --host 0.0.0.0 --port 8000
```

- 访问健康页面: `http://<SERVER_IP>:8000/`
- API 文档: `http://<SERVER_IP>:8000/docs`

5. 前端部署 (Vite + React)

前端默认在开发模式下访问 `http://localhost:8000` 的后端接口。如后端不在本机或端口不同, 请调整前端 API 基址。

5.1 安装依赖与运行

```
cd fronted/wing-quest-suite
npm i
npm run dev      # 开发模式 (http://localhost:8080/)
npm run build    # 生产构建 (生成 dist/)
npm run preview  # 预览生产构建 (默认 4173, 可改为 8080)
```

5.2 配置后端 API 地址

- 当前代码直接在 `fronted/wing-quest-suite/src/lib/api.ts` 使用 `http://localhost:8000`。跨机/跨端口部署时, 请修改为你的后端地址:
 - 文件位置: `fronted/wing-quest-suite/src/lib/api.ts` (多处使用)
 - 示例替换: `http://<SERVER_IP>:8000`
- 生产环境推荐通过反向代理 (如 Nginx) 将前端与后端置于同域不同路径 (例如 `/api`) , 以避免跨域问题。

5.3 Mapbox Token (酒店详情地图)

- 如果你使用 酒店详情 页面 (包含地图)，请替换为自己的 Mapbox 访问令牌：
 - 文件: `fronted/wing-quest-suite/src/pages/HotelDetail.tsx:234`
 - 将 `mapboxgl.accessToken = '...'` 替换为你的 Token (建议移至环境变量并读取)

6. 端口与进程

- 后端默认端口: `8000`
- 前端默认端口: `8080` (开发模式) 或 `4173` (preview)
- 如端口被占用，请更改启动端口或停止占用进程：
 - Windows (管理员 PowerShell) :

```
netstat -ano | findstr :8000
taskkill /F /PID <PID>
```

7. 快速验证

- 前端主页: `http://<SERVER_IP>:8080/home`
- 客服中心: `http://<SERVER_IP>:8080/customer-service`
- 价格日历: `http://<SERVER_IP>:8080/price-calendar`
- 退改签: `http://<SERVER_IP>:8080/refund-change`
- 我的订单: `http://<SERVER_IP>:8080/my-orders`
- 后端 API 文档: `http://<SERVER_IP>:8000/docs`

8. 部署建议 (生产)

- 反向代理与静态部署：
 - 使用 Nginx/Apache 将 `fronted/wing-quest-suite/dist` 作为静态目录部署
 - 将后端 FastAPI 通过 Uvicorn/Gunicorn 运行，监听内网端口，再由反向代理映射至公网
- 进程守护：
 - Windows: `nssm` / `pm2` (Node) 等工具
 - Linux: `systemd` 服务管理 (创建 unit 文件守护 Uvicorn 与前端静态服务)
- 容器化 (可选)：
 - 后端已提供 Docker 指引 (见 `backend/README.md`)
 - 可自定义 docker-compose 将 MySQL、后端、前端同时编排运行

9. 常见问题

- 数据库字符集/排序规则不兼容：
 - MySQL < 8.0 无法使用 `utf8mb4_0900_ai_ci`，请改为 `utf8mb4_general_ci`
- 跨域问题：

- 前后端不在同域时，需在后端启用 CORS 或使用反向代理同域部署
- 地图不可用：
 - 确认已替换 Mapbox Token (见 `fronted/wing-quest-suite/src/pages/HotelDetail.tsx:234`)
- 登录接口报错：
 - 检查后端 `.env` 中 `SECRET_KEY`、`SQLALCHEMY_DATABASE_URI` 是否正确

10. 目录与路径

- 根目录: `c:\Users\Lenovo\Desktop\skytrip\`
 - 数据库 SQL: `skytrip.sql` (若不存在，使用 `backend/skytrip.sql`)
 - 后端: `backend/`
 - 前端: `fronted/wing-quest-suite/`

如需将上述步骤写入一键脚本 (Windows PowerShell 或 Linux Shell)，我可以在根目录新增 `scripts/` 并提供自动化安装与部署脚本。请告知你的目标环境与端口规划。