



資料庫檢視器 - 完整部署檢查清單



一鍵部署指令

Windows 用戶

batch

1. 創建項目目錄

`mkdir database-viewer && cd database-viewer`

2. 執行 Windows 啟動腳本

`start.bat`

Linux/Mac 用戶

bash

1. 創建項目目錄

`mkdir database-viewer && cd database-viewer`

2. 執行啟動腳本

`chmod +x start.sh && ./start.sh`



必需文件清單



核心文件 (必須)

- ☐ `main.py` - FastAPI 後端主程序
- ☐ `requirements.txt` - Python 依賴項列表
- ☐ `static/index.html` - 前端主頁面
- ☐ `static/css/style.css` - CSS 樣式文件
- ☐ `static/js/app.js` - JavaScript 應用邏輯

✅ 配置文件 (推薦)

- ☐ `.env` - 環境變量配置
- ☐ `config.py` - 配置管理模組
- ☐ `start.py` - Python 啟動腳本

✅ 啟動腳本 (便利工具)

- ☐ `start.bat` - Windows 批處理啟動腳本
- ☐ `start.sh` - Linux/Mac Shell 啟動腳本

✅ 文檔文件 (可選)

- ☐ `README.md` - 項目說明文檔
- ☐ `api/models.py` - API 數據模型 (模組化版本)

✅ 目錄結構 (自動創建)

- ☐ `static/` - 靜態文件目錄
- ☐ `uploads/` - 上傳文件存儲
- ☐ `temp/` - 臨時文件目錄
- ☐ `logs/` - 日誌文件目錄

🔧 系統環境檢查

Python 環境

```
bash
```

```
# 檢查 Python 版本 (需要 3.8+)
```

```
python --version
```

```
python3 --version
```

```
# 檢查 pip
```

```
pip --version
```

```
pip3 --version
```

虛擬環境設置

```
bash

# 創建虛擬環境
python -m venv venv

# 激活虛擬環境
source venv/bin/activate # Linux/Mac
venv\Scripts\activate   # Windows

# 確認虛擬環境已激活 (應顯示 (venv))
which python # Linux/Mac
where python # Windows
```

依賴項安裝驗證

核心依賴項檢查

```
bash

# 安裝所有依賴項
pip install -r requirements.txt

# 逐一驗證關鍵套件
python -c "import fastapi; print('✅ FastAPI:', fastapi.__version__)"
python -c "import uvicorn; print('✅ Uvicorn: OK!)"
python -c "import duckdb; print('✅ DuckDB:', duckdb.__version__)"
python -c "import pymilvus; print('✅ PyMilvus:', pymilvus.__version__)"
python -c "import pandas; print('✅ Pandas:', pandas.__version__)"
```

依賴項問題解決

```
bash
```

如果安裝失敗，嘗試以下解決方案：

1. 升級 pip

python -m pip **install** --upgrade pip

2. 清除緩存

pip cache purge

3. 使用國內鏡像 (中國用戶)

pip **install** -r requirements.txt -i <https://pypi.tuna.tsinghua.edu.cn/simple/>

4. 逐個安裝關鍵套件

pip **install** fastapi uvicorn[standard]

pip **install** duckdb pandas numpy

pip **install** pymilvus python-multipart

啟動流程檢查

第一次啟動

bash

1. 直接運行主程序

python main.py

2. 使用啟動腳本 (推薦)

python start.py

3. 使用 uvicorn

uvicorn main:app --host **0.0.0.0** --port **8000** --reload

啟動成功標誌

看到以下輸出表示啟動成功：

INFO: Started server process [12345]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

連接測試

bash

1. 健康檢查

`curl http://localhost:8000/health`

2. 瀏覽器訪問

- 主界面: `http://localhost:8000`

- API 文檔: `http://localhost:8000/docs`

功能測試清單

前端界面測試

- ☐ 主頁面正常載入
- ☐ 左側功能選擇面板顯示
- ☐ 可以切換 Milvus 和 DuckDB 視圖
- ☐ 響應式設計在不同螢幕尺寸下工作正常

Milvus 功能測試

- ☐ 可以輸入連接信息
- ☐ 連接狀態正確顯示
- ☐ 如有 Milvus 服務器可連接測試
- ☐ 錯誤處理正常工作

DuckDB 功能測試

- ☐ 文件上傳區域正常顯示

- ☐ 支持拖拽上傳
- ☐ 可以選擇 .db/.duckdb 文件
- ☐ SQL 查詢區域功能正常

API 端點測試

```
bash
```

```
# 測試所有主要端點
```

```
curl http://localhost:8000/health
```

```
curl http://localhost:8000/milvus/collections
```

```
curl http://localhost:8000/duckdb/tables
```

```
# 使用瀏覽器訪問 API 文檔
```

```
http://localhost:8000/docs
```

常見問題快速解決

問題 1: 端口被占用

```
bash
```

```
# 錯誤: [Errno 98] Address already in use
```

```
# 解決方案:
```

```
# 1. 查找占用端口的進程
```

```
netstat -tulpn | grep 8000 # Linux/Mac
```

```
netstat -ano | findstr 8000 # Windows
```

```
# 2. 終止進程或更改端口
```

```
kill -9 <PID> # Linux/Mac
```

```
# 或修改 .env 中的 PORT=8001
```

問題 2: Python 套件安裝失敗

```
bash
```

錯誤: *ERROR: Could not install packages*

解決方案:

1. 檢查 Python 版本

python --version # 需要 3.8+

2. 升級工具

python -m pip install --upgrade pip setuptools wheel

3. 重新安裝

pip install -r requirements.txt --no-cache-dir

問題 3: 權限問題

bash

Linux/Mac 權限錯誤

sudo chown -R \$USER:\$USER database-viewer/

chmod -R 755 database-viewer/

Windows: 以管理員身分運行命令提示字元

問題 4: 虛擬環境問題

bash

重新創建虛擬環境

deactivate # 如果已激活

rm -rf venv # 刪除舊環境

python -m venv venv # 重新創建

source venv/bin/activate # 激活

pip install -r requirements.txt # 重新安裝

進階配置選項

自定義端口和主機

```
env

# .env 文件
HOST=127.0.0.1 # 只允許本地訪問
PORT=9000      # 使用不同端口
DEBUG=False    # 生產模式
```

日誌配置

```
env

LOG_LEVEL=DEBUG # 詳細日誌
LOG_FILE=./logs/app.log
```

文件上傳限制

```
env

MAX_FILE_SIZE=209715200 # 200MB
UPLOAD_DIR=./data/uploads
```

監控和維護

日誌監控

```
bash
```


實時查看日誌

```
tail -f logs/app.log
```

查看錯誤日誌

```
grep ERROR logs/app.log
```

日誌輪轉 (當文件過大)

```
mv logs/app.log logs/app.log.old
```

```
touch logs/app.log
```

系統資源監控

bash

檢查進程

```
ps aux | grep python
```

```
ps aux | grep uvicorn
```

檢查端口使用

```
netstat -tulpn | grep 8000
```

檢查磁碟空間

```
df -h # Linux/Mac
```

```
dir # Windows
```

定期維護

bash

清理臨時文件

```
rm -rf temp/*
```

```
rm -rf uploads/*
```

更新依賴項 (謹慎)

```
pip list --outdated
```

```
pip install --upgrade package_name
```

安全檢查清單

基本安全

- ☐ 更改默認密碼和金鑰
- ☐ 限制文件上傳大小
- ☐ 驗證上傳文件類型
- ☐ 設置適當的 CORS 來源

生產部署

- ☐ 使用 HTTPS
- ☐ 設置防火牆規則
- ☐ 配置反向代理
- ☐ 定期備份重要數據
- ☐ 設置日誌輪轉
- ☐ 監控系統資源使用

獲取幫助

自助診斷

1. 檢查 `logs/app.log` 文件
2. 訪問 <http://localhost:8000/health>
3. 查看瀏覽器控制台錯誤
4. 確認所有文件存在且權限正確

除錯模式

```
bash
```

```
# 啟用詳細日誌
```

```
export DEBUG=True # Linux/Mac
```

```
set DEBUG=True # Windows
```

```
# 重新啟動應用程序
```

```
python start.py
```

重置到初始狀態

```
bash
```

```
# 完全重置 (將刪除所有數據)
```

```
rm -rf venv uploads temp logs
```

```
python -m venv venv
```

```
source venv/bin/activate
```

```
pip install -r requirements.txt
```

```
python start.py
```

✓ 最終檢查清單

部署成功的最終確認：

- ☐ 應用程序成功啟動 (無錯誤訊息)
- ☐ 可以訪問 <http://localhost:8000>
- ☐ 前端界面正常載入和操作
- ☐ API 文檔可訪問 <http://localhost:8000/docs>
- ☐ 健康檢查返回正常狀態
- ☐ 日誌文件正常生成
- ☐ 能夠上傳和測試基本功能

🎉 恭喜！您已成功部署資料庫檢視器！

📱 快速命令參考

bash

一鍵啟動 (在項目目錄中)

python start.py

檢查狀態

curl http://localhost:8000/health

查看日誌

tail -f logs/app.log

停止服務器

Ctrl+C

重新啟動

python start.py