

# AIops 第三周

—

Luo Leng

# 上周回顾

1. Github创建及使用
2. Flask额外的操作

# 本周内容

1. 使用Streamlit搭建前端
2. 重构代码结构
3. 前后端连接并访问
4. Docker讲解
5. Web服务进行容器化

# 使用Streamlit搭建前端



# Streamlit

Web 应用的 Python 库

1. 简单直接
2. 组件化设计
3. 自动热重载



# 简单直接

- 用纯 Python 代码就能创建 Web 应用
- 不需要了解 HTML、CSS 或 JavaScript
- 一个 Python 文件就是一个完整的应用

```
with right:
    # Quick Python Examples
    st.subheader("Quick Python Examples")
    with st.expander("List Operations", expanded=False):
        st.code("""
        numbers = [1, 2, 3, 4, 5]
        numbers.append(6)      # Add to end
        numbers.pop()          # Remove last
        numbers.insert(0, 0)   # Insert at position
        """)
    st.write("---")
    with st.expander("Dictionary Operations", expanded=True):
        st.code("""
        person = {'name': 'Alice', 'age': 25}
        person['city'] = 'Beijing' # Add new key
        del person['age']           # Remove key
        """)
```

# 组件化设计

可以快速创建页面元素

- 例如标题文字
- 子标题
- 输入框
- 按钮等

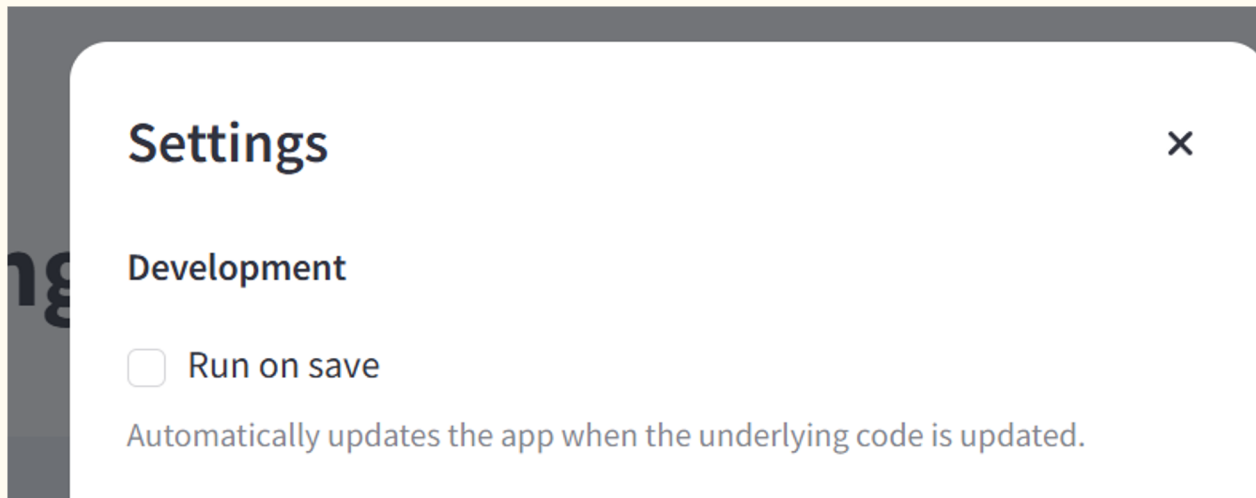
```
# 用户输入区域
input_col, space, button_col = st.columns([8,1,1])

with input_col:
    user_input = st.text_input("Ask something:")

with button_col:
    send_clicked = st.button("Send")
```

# 自动热重载

- 开发效率高 - 改代码立即看到效果
- 调试方便 - 可以快速测试和修改
- 用户体验好 - 实时响应,无需等待












# 重建代码结构





# 重建代码结构



main   [Go to file](#) [+](#) [Code](#)





 **Leng-Luo** Add .gitignore file 097033d · 2 weeks ago 

 .gitignore	Add .gitignore file	2 weeks ago
 README.md	Initial commit	2 weeks ago
 app.py	app.py, gitignore added	2 weeks ago

week3   [Go to file](#) [+](#) [Code](#)

This branch is 3 commits ahead of main. [Contribute](#)

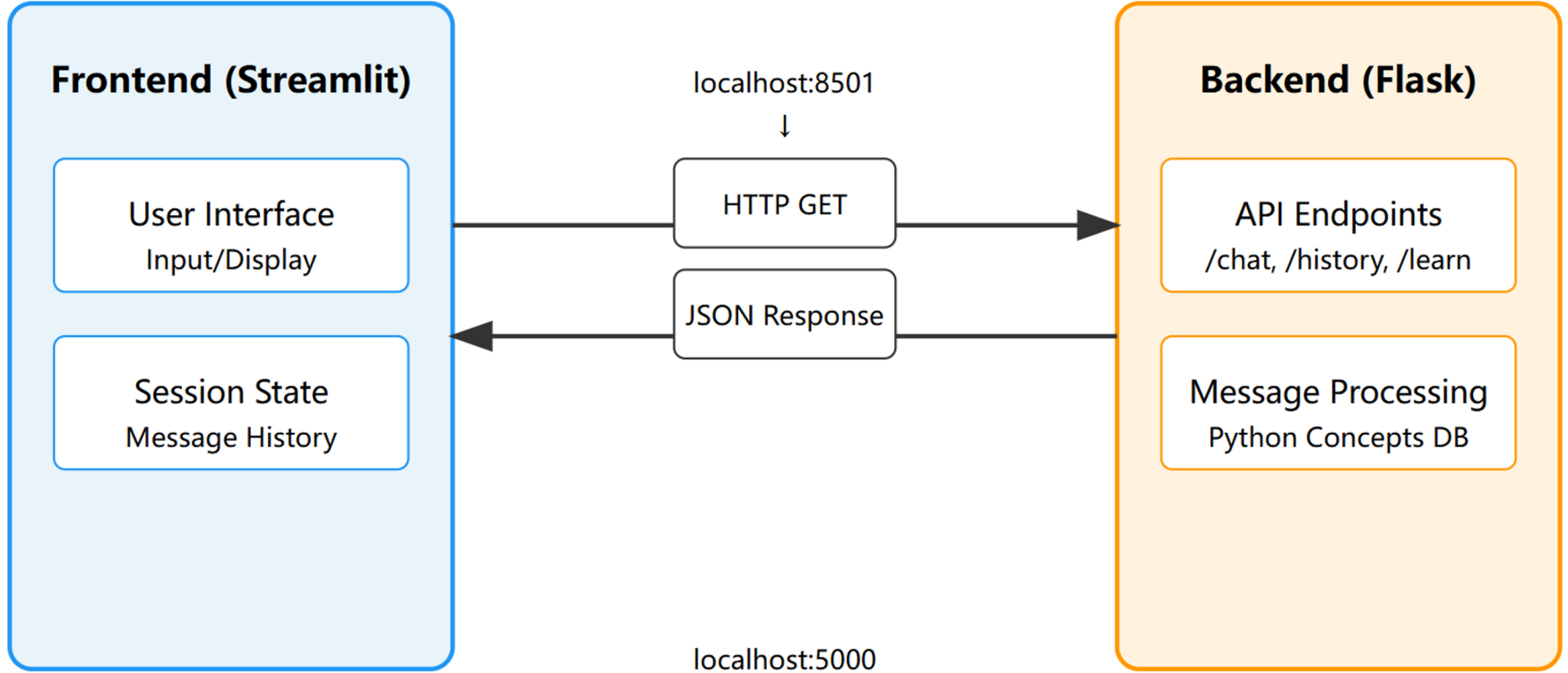
 **Leng-Luo** Update README.md ee4c7c3 · 1 hour ago 

 backend	week 3 example	1 hour ago
 frontend	week 3 example	1 hour ago
 .gitignore	week 3 example	1 hour ago
 README.md	Update README.md	1 hour ago

```
├─ backend/
│   └─ requirements.txt
│       └─ app.py
└─ frontend/
    └─ requirements.txt
        └─ app.py
```

# 前后端代码连接





# Docker讲解



# Docker

1. 镜像 image

2. 容器 container

3. 仓库 repo

---

# 镜像

一个只读的模板，包含了运行应用程序所需的所有内容

# 查看本地镜像

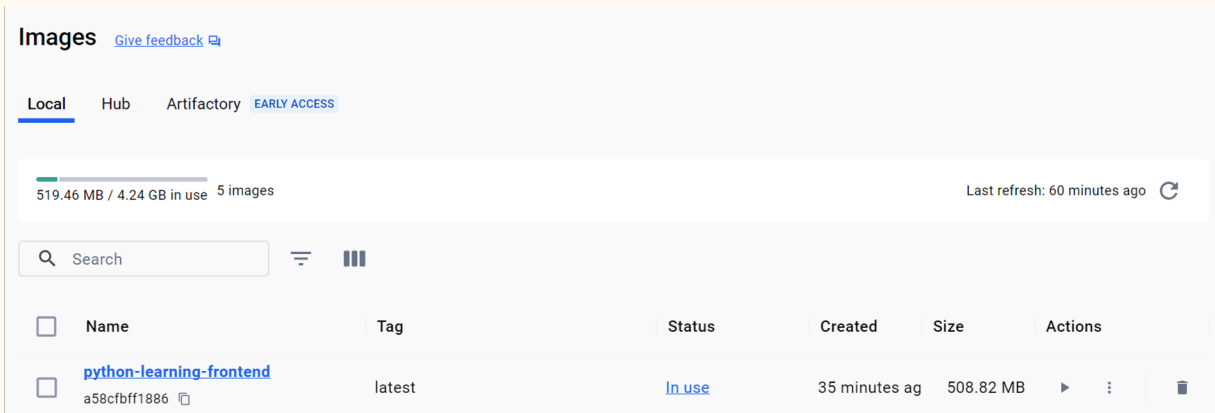
```
docker images
```

# 拉取镜像

```
docker pull python:3.9-slim
```

# 构建镜像

```
docker build -t myapp:1.0 .
```



# 容器

容器是镜像的运行实例，可以被启动、停止、删除

## # 创建并运行容器

```
docker run -d -p 5000:5000 --name myapp myapp:1.0
```

## # 容器操作

```
docker start container_id # 启动
```

```
docker stop container_id # 停止
```

```
docker restart container_id # 重启
```

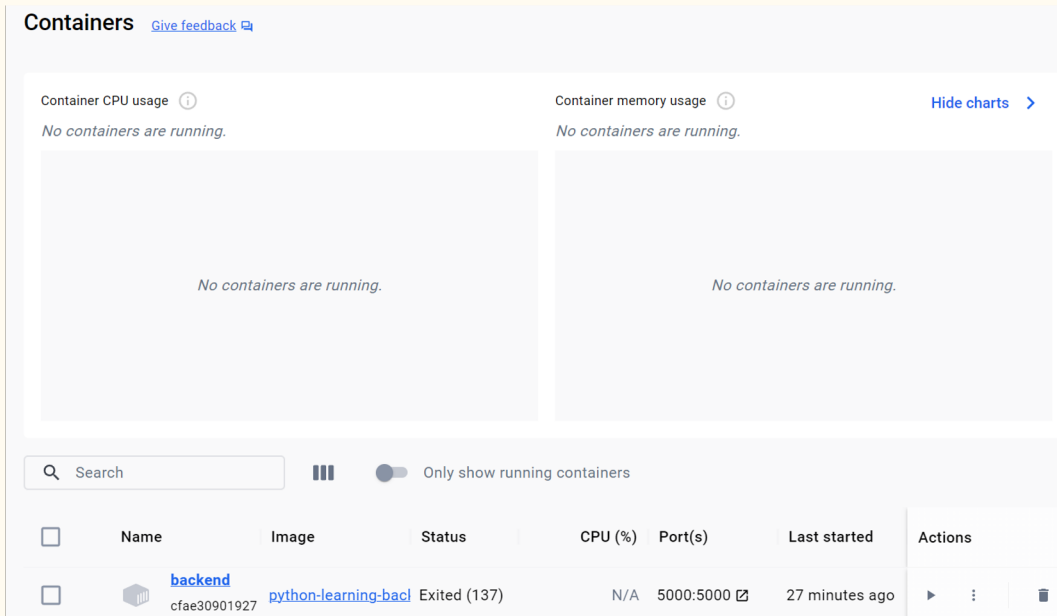
```
docker rm container_id # 删除
```

## # 容器状态

```
docker ps # 运行中的容器
```

## # 进入容器

```
docker exec -it container_id bash
```





# 仓库

用于存储和分发 Docker 镜像

# 登录到 Docker Hub

`docker login`

# 推送镜像到仓库

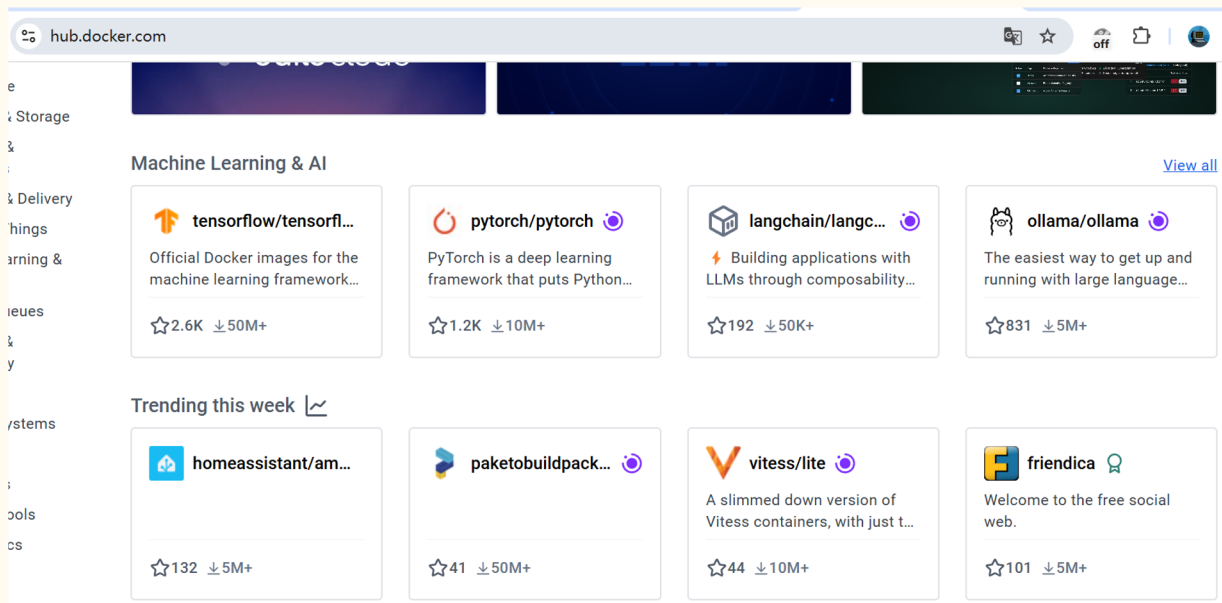
`docker push username/myapp:1.0`

# 从仓库拉取镜像

`docker pull username/myapp:1.0`

# 给镜像打标签

`docker tag myapp: 1.0 username/myapp:1.0`



# Web服务进行容器化



# Dockerfile

Dockerfile 是一个文本文件，包含了一系列指令和参数，用于自动化构建 Docker 镜像

FROM: 指定基础镜像

WORKDIR: 设置工作目录

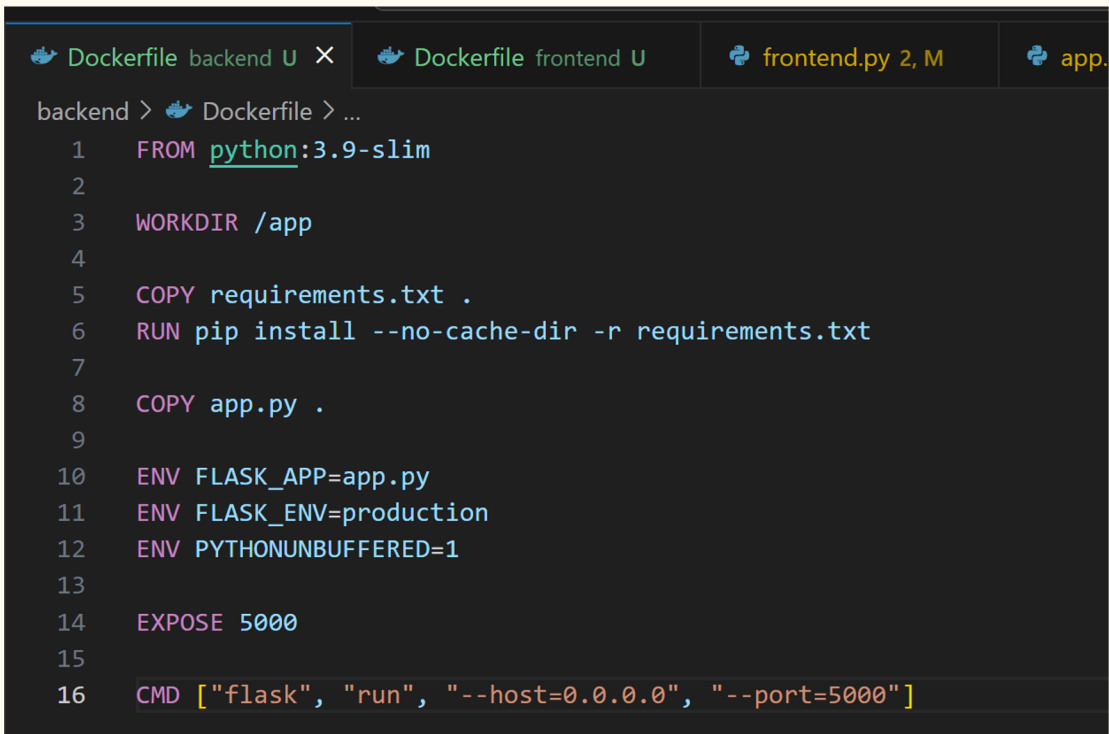
COPY/ADD: 复制文件到容器

RUN: 执行命令并创建新层

ENV: 设置环境变量

EXPOSE: 声明端口

CMD/ENTRYPOINT: 容器启动命令



```
backend > Dockerfile > ...
1 FROM python:3.9-slim
2
3 WORKDIR /app
4
5 COPY requirements.txt .
6 RUN pip install --no-cache-dir -r requirements.txt
7
8 COPY app.py .
9
10 ENV FLASK_APP=app.py
11 ENV FLASK_ENV=production
12 ENV PYTHONUNBUFFERED=1
13
14 EXPOSE 5000
15
16 CMD ["flask", "run", "--host=0.0.0.0", "--port=5000"]
```

# 指令

# Create network

```
docker network create python_learning_net
```

# Build images

```
docker build -t python-learning-backend .
```

```
docker build -t python-learning-frontend .
```

# Run containers

```
docker run -d --name backend --network python_learning_net -p 5000:5000 python-learning-backend
```

```
docker run -d --name frontend --network python_learning_net -p 8501:8501 python-learning-frontend
```

# 课后

<https://www.docker.com/> 安装docker应用，尝试把web应用容器化