

MMT-251 (Dự án WeApKrous)

Đây là dự án môn Mạng Máy Tính (MMT), hiện thực một hệ thống HTTP Server đầy đủ từ đầu (sử dụng socket và threading) bao gồm:

- Một **Proxy Server** đa luồng (cổng 8080).
- Một **Backend Pool** (các cổng 9000 , 9001 , 9002 ...).
- Một **Web Framework** tên là **WeApKrous** (chạy trên một backend, ví dụ 9001).

Kiến trúc này mô phỏng các hệ thống hiện đại (như NGINX + Gunicorn), trong đó Proxy (8080) đóng vai trò là "cổng vào" duy nhất và định tuyến request đến các backend phù hợp dựa trên tên miền (Host).

1. Cài đặt môi trường (BẮT BUỘC)

Bước 1.1: Tìm IP Mạng LAN của bạn

Vì Task 2 (Chat) yêu cầu các máy tính khác trong mạng LAN kết nối với bạn, bạn *không thể* dùng 127.0.0.1 . Bạn cần tìm IP thật của máy:

- **Windows:** Mở Command Prompt (cmd) và gõ ipconfig . Tìm địa chỉ IPv4 Address (ví dụ: 192.168.1.8).
- **macOS / Linux:** Mở Terminal và gõ ifconfig hoặc ip addr . Tìm địa chỉ inet (ví dụ: 192.168.1.8).

Trong toàn bộ hướng dẫn này, hãy thay YOUR_MACHINE_IP bằng IP bạn vừa tìm được (ví dụ: 192.168.1.8).

Bước 1.2: Cài đặt file hosts

Hệ thống này sử dụng Proxy định tuyến bằng **tên miền ảo** (Virtual Host) như app1.local . Bạn cần "dạy" máy tính của mình (và máy của bạn bè) hiểu các tên miền này trở về đâu.

1. Mở file hosts của máy bạn với quyền **Administrator**:
 - **Windows:** C:\Windows\System32\drivers\etc\hosts
 - **macOS / Linux:** /etc/hosts
2. Thêm các dòng sau vào cuối file và lưu lại. (Nhớ thay YOUR_MACHINE_IP):

```
# Ví dụ:  
# 192.168.1.8    app1.local  
# 192.168.1.8    app2.local  
  
YOUR_MACHINE_IP    app1.local  
YOUR_MACHINE_IP    app2.local
```

2. Cách chạy dự án

Bạn cần mở 3 Terminal (Command Prompt) riêng biệt và chạy song song 3 tiến trình sau.

(QUAN TRỌNG: Hãy dùng IP Mạng LAN của bạn, không dùng 0.0.0.0)

Terminal 1: Chạy Proxy (Cổng vào 8080)

Proxy sẽ lắng nghe ở YOUR_MACHINE_IP:8080 và đọc file config/proxy.conf .

```
# Ví dụ: python start_proxy.py --server-ip 192.168.1.8 --server-port 8080  
python start_proxy.py --server-ip YOUR_MACHINE_IP --server-port 8080
```

Terminal 2: Chạy Backend "thô" (Port 9000)

Đây là backend mặc định (theo config/proxy.conf), ứng với host là IP.

```
# Ví dụ: python start_backend.py --server-ip 192.168.1.8 --server-port 9000  
python start_backend.py --server-ip YOUR_MACHINE_IP --server-port 9000
```

Terminal 3: Chạy WeApKrous App (Port 9001)

Đây là ứng dụng WebApp chính (xử lý Login, Chat API), ứng với host app1.local .

```
# Ví dụ: python start_sampleapp.py --server-ip 192.168.1.8 --server-port 9001  
python start_sampleapp.py --server-ip YOUR_MACHINE_IP --server-port 9001
```

3. Cách kiểm tra & Hiện thực

Sau khi cả 3 Terminal đều chạy, bạn có thể bắt đầu kiểm tra:

Task 1: Kiểm tra Đăng nhập (Trên trình duyệt)

1. Mở trình duyệt (nên dùng tab **Ẩn danh** để tránh lỗi cache).
2. **ĐỊA CHỈ ĐÚNG (Để vào App 9001):** Gõ vào thanh địa chỉ: <http://app1.local:8080/login>
 - **Luồng:** Request → Proxy (8080) → Đọc Host: app1.local → Chuyển đến Backend (9001)
→ Bạn thấy trang Login.
 - Đăng nhập với tài khoản trong file db/users.json .
 - Sau khi login thành công, bạn sẽ được chuyển đến /index.html và cookie auth=true được set.
3. **ĐỊA CHỈ SAI (Để kiểm tra Proxy):** Gõ vào thanh địa chỉ: http://YOUR_MACHINE_IP:8080/login
(ví dụ: <http://192.168.1.8:8080/login>)

- **Luồng:** Request → Proxy (8080) → Đọc Host: YOUR_MACHINE_IP:8080 → Chuyển đến Backend (9000) → Backend "thô" (9000) không có route /login → Bạn thấy lỗi 404 Not Found.
- Điều này **chứng minh** Proxy đã định tuyến đúng!

Task 2: Hiện thực Chat API (Máy chủ Tracker)

Phần Task 2 (Chat hybrid) yêu cầu một "máy chủ theo dõi" (Tracker) để các peer tìm thấy nhau.

- **Nơi code:** Các API này cần được thêm vào file `apps/sampleApp.py` (bên trong hàm `create_sampleapp()`).
- **API cần thêm (ví dụ):**
 - GET `/get-list` : Để peer lấy danh sách các peer khác đang online.
 - POST `/submit-info` : Để peer đăng ký IP và port chat của mình với máy chủ.
- **Cách test:** Dùng Postman để gọi các API trên (ví dụ: `http://app1.local:8080/get-list`) sau khi đã đăng nhập (Task 1) để lấy cookie.