

DNS Routing Manager - Краткое руководство пользователя

Что это такое

DNS Routing Manager автоматически направляет интернет-трафик через разные сетевые интерфейсы на основе доменных имен. Например, российские сайты через локальную сеть, международные через VPN.

Быстрый старт

1. Установка

```
bash

# Клонировать проект
git clone <repository-url>
cd dns-routing-manager

# Создайте виртуальное окружение
python3 -m venv venv
source venv/bin/activate

# Установите зависимости
pip install -r requirements.txt
```

2. Определите ваши сетевые интерфейсы

```
bash

# Посмотрите доступные интерфейсы
ifconfig | grep -E "(en|utun)"

# Найдите ваш основной gateway
netstat -rn | grep "default"
```

3. Настройте конфигурацию

Отредактируйте `config/settings.yaml`:

```
yaml
```

network:

local:

interface: "en0" *# Ваш основной интерфейс (en0, en1, en7...)*

gateway: "192.168.1.1" *# Ваш роутер*

is_tunnel: false

vpn:

interface: "utun0" *# Ваш VPN интерфейс (utun0, utun4...)*

gateway: null

is_tunnel: true

4. Проверьте статус

bash

python3 run.py status

Основные команды

Просмотр информации

bash

Статус системы

python3 run.py status

Статистика DNS кэша

python3 run.py dns cache

Тестирование доменов

bash

Резолвить домен

python3 run.py dns resolve yandex.ru

python3 run.py dns resolve google.com

С wildcard поддоменами

python3 run.py dns resolve github.com --type wildcard

Управление маршрутами

bash

```
# Добавить маршрут
python3 run.py routes add 8.8.8.8 --via vpn
python3 run.py routes add 192.168.1.0/24 --via local

# Проверить маршрут
python3 run.py routes check 8.8.8.8

# Удалить маршрут
python3 run.py routes remove 8.8.8.8
```

Массовая обработка

```
bash

# Сначала протестируйте (безопасно)
python3 run.py process --dry-run

# Обработать российские домены
python3 run.py process --ru-only

# Обработать международные домены
python3 run.py process --com-only

# Обработать все домены
python3 run.py process
```

Настройка доменов

Российские домены (`config/domains_ru.txt`)

```
# Поисковики
yandex.ru
mail.ru

# Банки
sberbank.ru
tinkoff.ru

# Wildcard домены (все поддомены)
*.gov.ru
*.edu.ru

# Deep wildcard (много поддоменов)
**.gosuslugi.ru
```

Международные домены ((config/domains_com.txt))

```
# Разработка
github.com
stackoverflow.com

# Облачные сервисы
*.amazonaws.com
*.googleapis.com
*.cloudflare.com

# Социальные сети
facebook.com
twitter.com
```

IP адреса и подсети

Локальные IP ((config/ips_local.txt)):

```
# Локальные подсети
192.168.0.0/16
10.0.0.0/8

# Конкретные серверы
192.168.1.100
```

VPN IP ((config/ips_vpn.txt)):

```
# DNS серверы
8.8.8.8
1.1.1.1

# Конкретные сервисы
52.84.0.0/15
```

Типичные сценарии использования

Сценарий 1: Российские сайты через локальную сеть

```
bash
```

```
# 1. Настройте российские домены в config/domains_ru.txt
```

```
# 2. Обработайте их
```

```
python3 run.py process --ru-only
```

```
# 3. Проверьте результат
```

```
python3 run.py routes check 77.88.55.88 # Yandex IP
```

Сценарий 2: Обход блокировок через VPN

```
bash
```

```
# 1. Добавьте заблокированные домены в config/domains_com.txt
```

```
# 2. Обработайте их
```

```
python3 run.py process --com-only
```

```
# 3. Проверьте что трафик идет через VPN
```

```
python3 run.py routes check 8.8.8.8
```

Сценарий 3: Корпоративная сеть + VPN

```
bash
```

```
# 1. Добавьте корпоративные подсети в config/ips_local.txt:
```

```
10.0.0.0/8
```

```
192.168.0.0/16
```

```
# 2. Добавьте корпоративные домены в config/domains_ru.txt:
```

```
*.company.com
```

```
intranet.company.com
```

```
# 3. Примените настройки
```

```
python3 run.py process --ru-only
```

Безопасность и восстановление

Перед изменениями

```
bash
```

```
# Всегда тестируйте сначала
```

```
python3 run.py process --dry-run
```

```
# Запомните текущие маршруты
```

```
netstat -rn > backup_routes.txt
```

Откат изменений

```
bash

# Удалить все добавленные маршруты
python3 run.py routes clear

# Очистить кэш
python3 run.py dns clear
```

Проверка работы

```
bash

# Проверить конкретный маршрут
route -n get yandex.ru
route -n get google.com

# Сравнить с нашими данными
python3 run.py routes check <IP>
```

Решение проблем

Ошибка "Permission denied"

```
bash

# Убедитесь что у вас есть права sudo
sudo -v

# Создайте правильные права на файлы
chmod 755 data data/cache
```

Ошибка "Interface not found"

```
bash

# Проверьте доступные интерфейсы
ifconfig | grep -E "(en|utun)"

# Обновите config/settings.yaml с правильными именами
```

VPN не работает

```
bash
```

```
# Проверьте что VPN интерфейс активен
ifconfig utun0 # или ваш интерфейс
```

```
# Проверьте default routes
netstat -rn | grep "default"
```

DNS не резолвится

```
bash

# Проверьте что dig работает
dig google.com

# Очистите DNS кэш
python3 run.py dns clear

# Проверьте DNS серверы в config/settings.yaml
```

Мониторинг

Ежедневная проверка

```
bash

# Статус системы
python3 run.py status

# Количество активных маршрутов
netstat -rn | grep -E "(en7|utun4)" | wc -l
```

Логи

Логи сохраняются в `logs/dns_routing.log` (если включено логирование в конфигурации).

Автоматизация

Создание скрипта автозапуска

```
bash

#!/bin/bash
cd /path/to/dns-routing-manager
source venv/bin/activate
python3 run.py process --ru-only
```

Cron задача (ежедневное обновление)

```
bash
```

```
# Добавьте в crontab -e
```

```
0 9 * * * /path/to/update_routes.sh
```

Примеры конфигураций

Для домашнего использования

```
yaml
```

```
network:
```

```
  local:
```

```
    interface: "en0"      # WiFi
```

```
    gateway: "192.168.1.1" # Домашний роутер
```

```
vpn:
```

```
  interface: "utun0"      # Личный VPN
```

Для корпоративной сети

```
yaml
```

```
network:
```

```
  local:
```

```
    interface: "en1"      # Ethernet
```

```
    gateway: "10.0.0.1"   # Корпоративный gateway
```

```
vpn:
```

```
  interface: "utun2"      # Корпоративный VPN
```

Для разработчика

```
yaml
```

```
network:
```

```
  local:
```

```
    interface: "en7"      # Thunderbolt Ethernet
```

```
    gateway: "10.255.0.1" # Офисная сеть
```

```
vpn:
```

```
  interface: "utun4"      # WireGuard
```

Помните: Изменения маршрутов влияют на весь сетевой трафик. Всегда тестируйте в безопасном режиме перед применением в продуктивной среде.