

Веб-сервер, прокси-сервер

14 февраля 2026 г.

Что такое веб-сервер?

HTTP веб-сервер - программное обеспечение, которое принимает HTTP-запросы от **клиентов**, обрабатывает и возвращает HTTP-ответы

Принцип работы:

1. Клиент отправляет HTTP-запрос к **ресурсу**, указывая его **идентификатор**
2. Веб-сервер **обрабатывает** запрос и возвращает HTTP ответ

Примеры веб-серверов:

- Apache
- Nginx
- Python http.server

Примеры клиентов веб-серверов:

- Браузер
- Утилиты командной строки (curl, wget)
- Другой веб-сервер (reverse proxy)
- Серверное приложение (backend приложение)

Ресурс - абстрактная сущность, идентифицируемая URL, доступ к которой предоставляется через веб-сервер.

Примеры ресурсов:

- Файл (HTML файл готовый или обработанный, бинарные файлы)
- Динамически сгенерированные данные
- Записи в БД
- Состояние сервиса
- Проксируемые ресурсы другого сервиса

Uniform Resource Locator (URL)



https://developer.mozilla.org/en-US/docs/Learn_web_development...

<https://habr.com/ru/articles/190154>

Request

```
POST / HTTP/1.1
```

```
Host: developer.mozilla.org
```

```
User-Agent: curl/8.6.0
```

```
Accept: */*
```

```
Content-Type: application/json
```

```
Content-Length: 345
```

```
{  
  "data": "ABC123"  
}
```

Start line

Headers

Empty line

Body

Response

```
HTTP/1.1 403 Forbidden
```

```
Server: Apache
```

```
Date: Fri, 21 Jun 2024 12:52:39 GMT
```

```
Content-Length: 678
```

```
Content-Type: text/html
```

```
Cache-Control: no-store
```

```
<!DOCTYPE html>  
<html lang="en">  
(more data...)
```

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Messages>

Что посмотрим:

- Запуск статического и динамического веб-сервера
- Обращение к веб-серверу из браузера, developers tools
- Обращение к веб-серверу с помощью curl

Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

Прокси сервер — сервер-посредник между клиентом (кто запрашивает ресурс) и сервером (кто предоставляет ресурс)

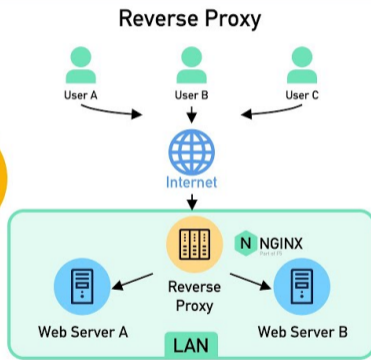
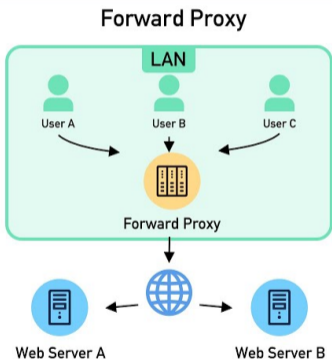
Forward proxy:

- Клиент → Прокси → Интернет
- Скрывает клиента от сервера
- Пример: корпоративный прокси

Reverse proxy:

- Интернет → Прокси → Сервер
- Скрывает сервер от клиента
- Пример: nginx перед приложением

Forward Proxy vs. Reverse Proxy



Что посмотрим:

- Маршрутизация по портам
- Маршрутизация по path
- Маршрутизация по заголовку Host

Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

- **HTTP** — основа веб-коммуникации (заголовки, статус коды)
- **Forward Proxy** — клиент → прокси → интернет
- **Reverse Proxy** — интернет → прокси → сервер