ПМ3 Разработка модулей ПО.

РО 3.2 Разрабатывать модули с применением DOM API, Regexp, HTTP

Тема 1. Протокол HTTP.

Лекция 1. Основы HTTP: HTTP/1.0 и HTTP/1.1.

Цель занятия:

Сформировать представление о принципах работы протокола HTTP, научиться различать версии HTTP/1.0 и HTTP/1.1, понимать структуру HTTP-запросов и ответов, применять знания на практике при работе с простыми запросами.

Учебные вопросы:

1. Назначение и роль протокола HTTP в работе сети Интернет.

2. Основные характеристики HTTP: клиент–серверная архитектура, stateless-природа.

1. Назначение и роль протокола HTTP в работе сети Интернет.

HTTP (HyperText Transfer Protocol) — протокол прикладного уровня, изначально созданный для передачи гипертекста (HTML-документов) в сети Интернет.

Сегодня HTTP используется для передачи любых данных: веб-страниц, изображений, видео, API-запросов и даже файловых потоков.

Роль HTTP в сети Интернет

HTTP - основной протокол Всемирной паутины (WWW):

- Без него невозможна работа браузеров, веб-сайтов и сервисов.
- Обеспечивает взаимодействие клиент–сервер:
- клиент (обычно браузер или приложение) формирует запрос;
- сервер обрабатывает запрос и возвращает ответ.

Простота и универсальность:

- передаёт данные в текстовом формате;
- легко анализируется человеком (можно читать запросы/ответы в «сыром» виде).

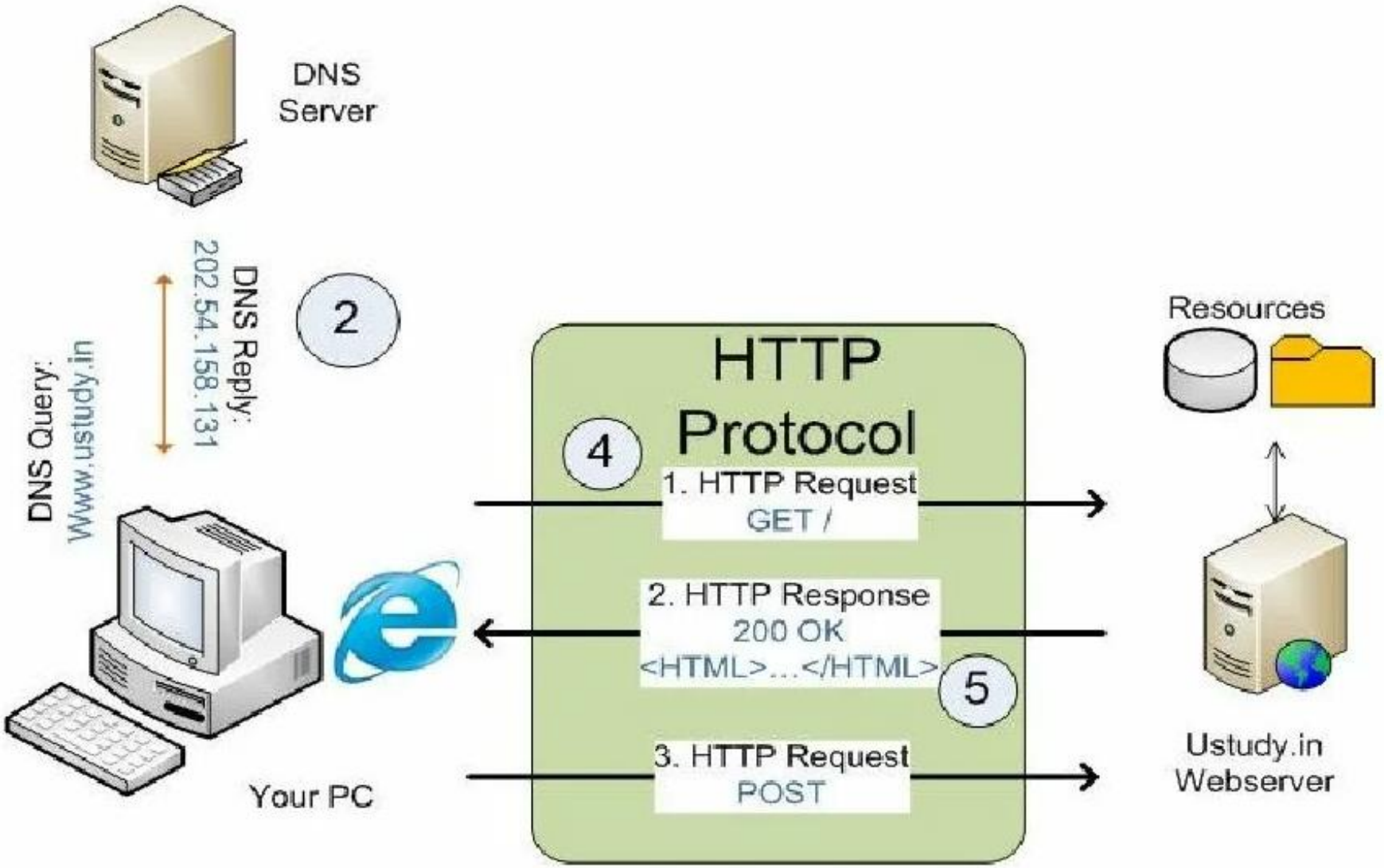
Расширяемость:

- новые возможности добавляются через заголовки и версии протокола;

Ключевые особенности:

- Клиент–серверная модель: инициатор взаимодействия всегда клиент.
- Stateless (без состояния): каждый запрос не зависит от предыдущего (сессии реализуются через дополнительные механизмы: cookies, токены).
- Универсальность передачи: HTTP не привязан к конкретному типу данных, а лишь задаёт правила взаимодействия.

Протокол HTTP



Значение для развития Интернета

- Сделал возможной работу Всемирной паутины и развитие веб-технологий.
- Лёгкость реализации и расширяемость привели к его широкому распространению.
- На основе HTTP работают современные протоколы: HTTPS (HTTP + TLS для защиты данных), а также API-сервисы (REST, GraphQL).

📌 **Краткий вывод:**

HTTP — это фундаментальный протокол Интернета, обеспечивающий обмен данными между клиентами и серверами.

Он стал основой для создания и развития Всемирной паутины, а благодаря своей простоте и универсальности используется повсеместно — от просмотра сайтов до взаимодействия приложений через API.

2. Основные характеристики HTTP.

Клиент–серверная архитектура.

HTTP построен на модели «клиент–сервер»:

- Клиент (браузер, мобильное приложение, скрипт) инициирует соединение, отправляя запрос.
- Сервер (веб-сервер: Apache, Nginx, IIS и др.) принимает запрос, обрабатывает его и возвращает ответ.

Роли строго разделены:

- клиент никогда не ждёт инициативы от сервера;
- сервер отвечает только в ответ на запрос клиента.

Такая модель обеспечивает:

- простоту взаимодействия;
- масштабируемость (один сервер обслуживает множество клиентов).

Природа «stateless» (без сохранения состояния):

- HTTP не хранит состояние соединения.
- Каждый запрос обрабатывается как новый, сервер не «помнит» предыдущие взаимодействия.
- Например: если пользователь открывает две страницы подряд, сервер рассматривает их как два независимых запроса.

Преимущества stateless-модели:

- простота реализации протокола;
- лёгкость масштабирования (серверы не обязаны хранить сессии пользователей).

Недостатки:

- необходимость дополнительных механизмов для «запоминания» пользователя (cookies, URL-параметры, сессионные токены).
- усложнение разработки интерактивных веб-приложений.

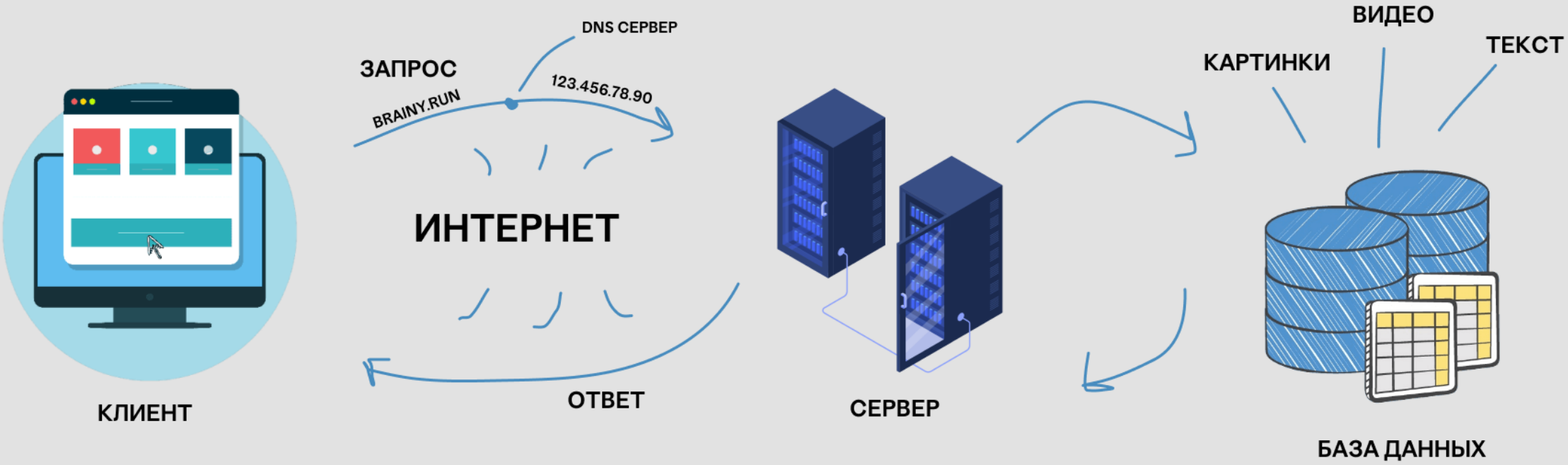
Пример:

- Клиент → запрашивает GET /index.html
- Сервер → возвращает HTML-документ.
- Если клиент отправляет второй запрос GET /about.html, сервер не «знает», что это тот же пользователь.

Плюсы и минусы stateless-модели HTTP

Плюсы	Минусы
Простая реализация протокола (каждый запрос автономен)	Нет «памяти» у сервера, все запросы обрабатываются независимо
Лёгкость масштабирования (серверы не обязаны хранить сессии, можно балансировать нагрузку между разными серверами)	Требуются дополнительные механизмы для хранения состояния (cookies, токены, сессии)
Снижение нагрузки на сервер (не хранит историю взаимодействия с клиентами)	Усложнение разработки интерактивных веб-приложений
Универсальность — любой клиент может обращаться к серверу без специальных условий	Дополнительный сетевой трафик (например, передача cookies при каждом запросе)

КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА



Пример с cookies, чтобы показать, как HTTP решает проблему «stateless».

Первый запрос клиента (без cookies)

GET /profile HTTP/1.1

Host: example.com

Ответ сервера:

HTTP/1.1 200 OK

Set-Cookie: session_id=abc123; Path=/; HttpOnly

Content-Type: text/html

<html>Добро пожаловать!</html>

👉 Здесь сервер создаёт уникальный идентификатор сессии (**session_id=abc123**) и отправляет его клиенту через заголовок Set-Cookie.

Последующий запрос клиента (с cookie)

GET /profile HTTP/1.1

Host: example.com

Cookie: session_id=abc123

Ответ сервера:

HTTP/1.1 200 OK

Content-Type: text/html

<html>Ваш профиль: Иван Иванов</html>

👉 Теперь клиент передаёт cookie обратно, и сервер «узнаёт» пользователя по session_id.

Выводы:

Архитектура HTTP упрощает взаимодействие между клиентами и серверами и позволила масштабировать Интернет.

Stateless-природа делает протокол простым и гибким, но требует дополнительных решений для реализации «сессий» и персонализации.

Контрольные вопросы:

- Какова основная роль протокола HTTP в работе Интернета?
- Что означает клиент–серверная модель в контексте HTTP?
- Почему HTTP называют «протоколом без состояния» (stateless)?
- Какую проблему решают cookies в HTTP?

Домашнее задание:

1. https://ru.hexlet.io/courses/http_protocol