Web Fundamentals

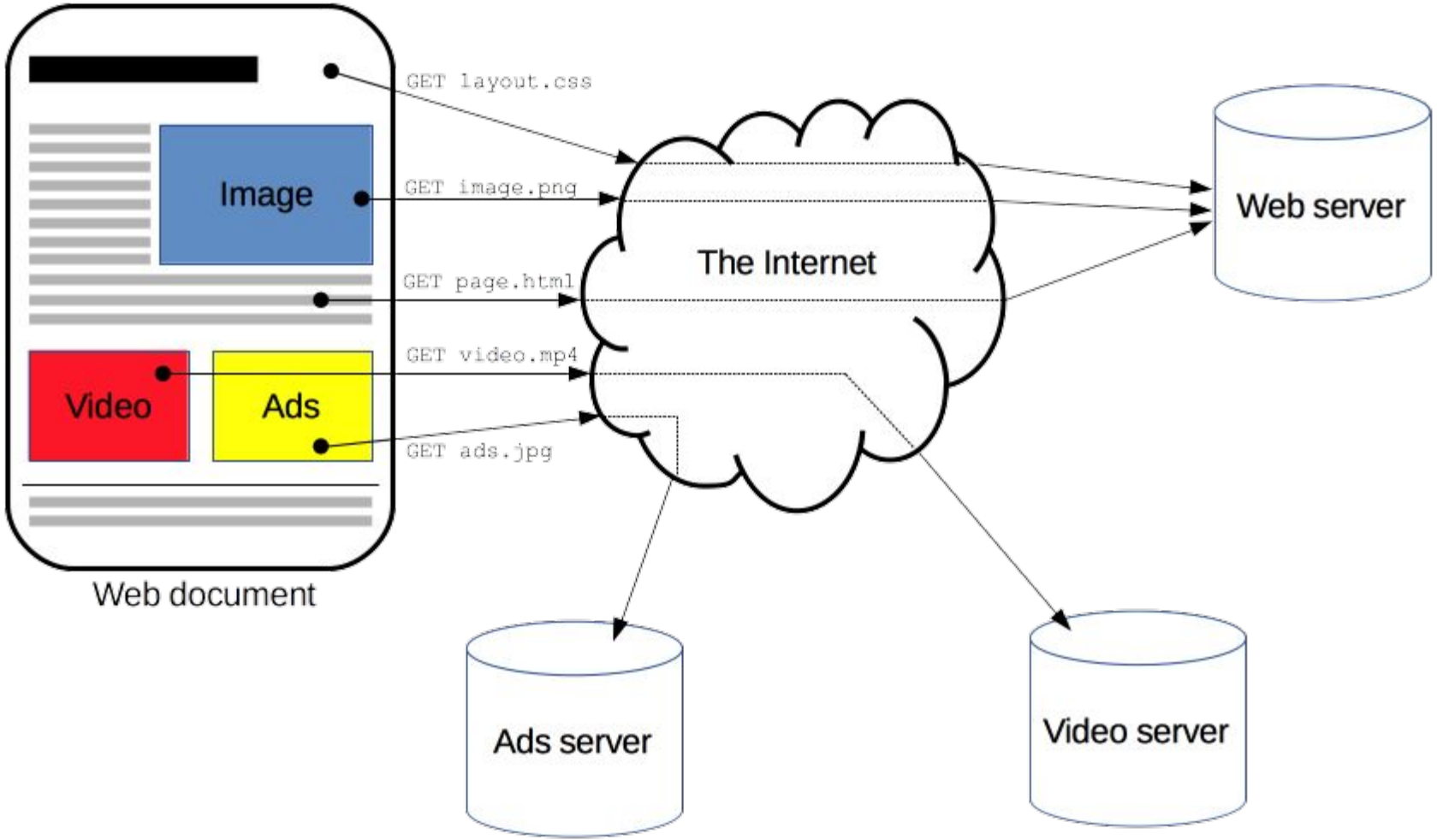
Prepared by
Mohamed Abdelmeged
Senior Software Engineer - Fixed Solutions
Freelancer Nodejs Instructor - ITI & AMIT
www.linkedin.com/in/mabdelmeged/

AGENDA

1. Introduction to Web.
2. HTTP
    1. Definition
    2. HTTP request response cycle
    3. HTTP guidelines.

Introduction to WEB

- The web has, from its inception, been structured around the idea of **resources**.
- Generally a platform for sharing simple text/HTML based files, documents, images etc. (**Resource-oriented**)
- Most web applications are built around a **client-server model.**
- **Client** could be something as simple as a web browser, crawlers, programs, scripts.
- **Server** presents only as a single machine virtually
- The web has evolved into a **more complex network** of interconnected applications.
- Between the Web browser and the server, numerous computers and machines relay theses messages.


GET layout.css
Image
GET image.png
GET page.html
Video
Ads
GET video.mp4
GET ads.jpg
Web document
The Internet
Web server
Ads server
Video server

But How these messages are transferred ....?

1. clients and servers had to agree upon a set of conventions—**a protocol**—that dictates all communication between them.
2. This protocol allows a web server to:
   a. **Receive** the information—**requests**—sent by any client.
   b. **Process** these requests..
   c. **Send** the appropriate **responses.**
3. There are lots of protocols that can transfer data between client and a server.
   a. Ex: HTTP, HTTPS, FTP, SMTP
   b. Each of them has its **purposes** and **functionalities.**

2. Hypertext Transfer Protocol “HTTP”

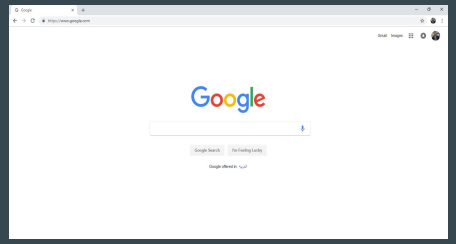
•••

What is it all about ?

2.1. What is HTTP ?

1. HTTP stands for Hypertext Transfer Protocol.
2. It is the **standard protocol** for transferring **web pages** (and their content) across the Internet.
3. HTTP provides a structured format for exchanging data over the web.
4. HTTP is a **stateless** protocol. Server doesn't keep any data between 2 requests.
5. HTTP has the guidelines needed to describe the following:
    a. **Client request.**
    b. **Server response.**
6. Let's check the whole request/response cycle.....

Client Agent



DNS server

(1) DNS look up **www.google.com**

(2) DNS resolves name to **IP address** (172.217.171.228)

2.2 HTTP Request Response Cycle

(3) HTTP Request

(8) HTTP Response

Web Server (nginx / Apache)

(7) HTTP Response

(4) Redirect request to application server

Application Server (express)

(6) Fetched data

(5) Fetch data

Database server

2.2 HTTP Guidelines

2.2.1 HTTP General guidelines

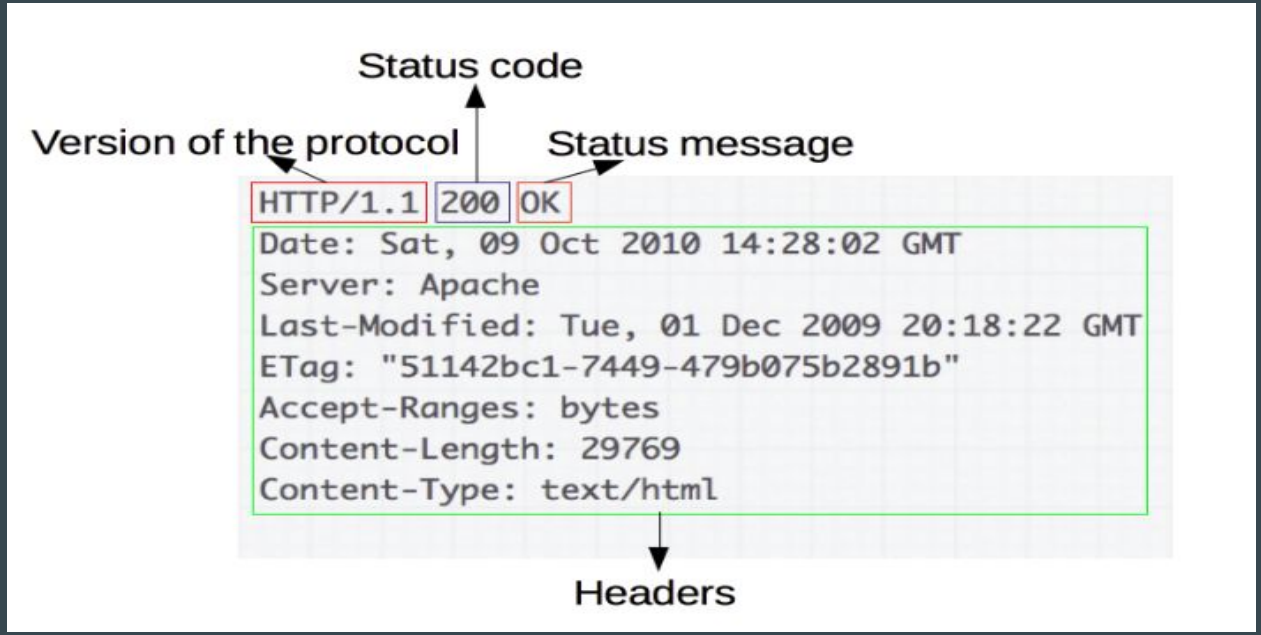
1. HTTP request/response usually consists of request/response header and request/response body according to http method.
2. headers hold metadata about the request/response.(http method, accepted format)
3. body usually consists of actual data sent from/to client/server.

2.2.2 URL

1. Stands for Uniform Resource Locator
2. URL refers to the physical location of the resource to be fetched. “/home.html”
3. URI (uniform Resource Identifier): refers to any resource that needs to be fetched.

   It may be a physical resource or not. (“/users” , “/users/user1/posts”)

2.2.5 HTTP RESPONSE



2.2.3 HTTP METHODS

1. It is a way used by client to describe the type of request(action) being made.
2. client may request to **create**, **delete**, **update** or simply **read** from a resource.
3. this corresponds to making **POST**, **DELETE**, **PUT** or **GET** requests respectively.
4. There are other HTTP methods like HEAD, OPTIONS ...etc.

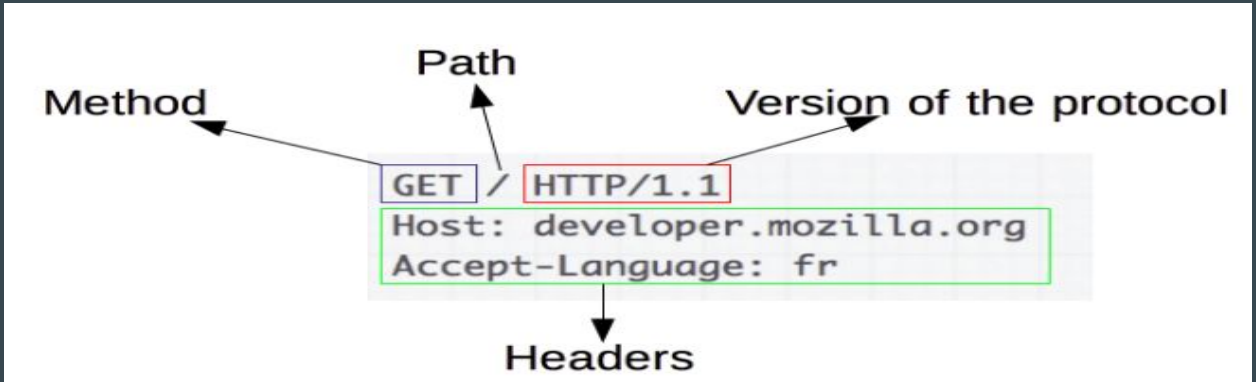
2.2.4 HTTP STATUS CODE

1. It is a useful way to provide information to client about the status of a request.
2. HTTP defines several status codes, each pertaining to a specific scenario.
    1. **2xx**: imply that the request completed successfully. (200)
    2. **3xx**: A code in the 3xx series implies redirection. (301)
    3. **4xx**: A 4xx errors are used when there is an error. (400, 401, 422, 404)
    4. **5xx**: Lastly, a 5xx response is used when there is an error on the server side. (500)
3. A Full list of HTTP status code: **https://httpstatuses.com/**

2.2.4 HTTP HEADERS

1. They serve to provide **additional information** for handling requests and responses.
2. They appear as **key value pairs** and provide a host of information such as:
    1. The cache policy for the response.
    2. the acceptable response types enforced by the client.
    3. the preferred language of response.
    4. Encoding.
    5. Credentials for **authentication** and **authorization**—such as **access tokens**.

2.2.5 HTTP REQUEST



Stay Tuned for Next part.....

Introduction to Nodejs