

# How The Web Works

## Part One: Solidify Terminology

1. HTTP stands for Hypertext Transfer Protocol. It is a protocol used for communication between web servers and web clients (browsers) on the Internet.
2. A URL (Uniform Resource Locator) is a web address that identifies a specific resource (such as a webpage, image, or file) on the Internet.
3. DNS (Domain Name System) is a system that translates domain names (like google.com) into IP addresses (like 216.58.194.174) that computers use to communicate with each other on the Internet.
4. A query string is a part of a URL that contains data that is sent to a web server as part of an HTTP request. It's usually indicated by a question mark followed by key-value pairs.
5. Two common HTTP verbs are GET and POST. GET is used to retrieve information from a server, while POST is used to submit data to a server.
6. An HTTP request is a message sent by a web client (such as a browser) to a web server, asking for a specific resource.
7. An HTTP response is a message sent by a web server to a web client in response to an HTTP request, containing the requested data or resources.
8. An HTTP header is a piece of metadata that provides additional information about an HTTP request or response. Examples of request headers may include "Authorization" which includes credentials (such as a username and password) needed to access a protected resource, and "Accept-Language" (specifying the preferred language for the response). Response headers may include "Date", which indicates the date and time when the response was sent, and "Server" which specifies the name and version of the web server software used to generate the response.

9. When a user types "http://somesite.com/some/page.html" into a browser, several processes occur:
- The browser parses the URL into its component parts (protocol, domain, and path).
  - The browser uses DNS to look up the IP address associated with the domain name "somesite.com".
  - The browser sends an HTTP GET request to the server at that IP address, requesting the resource at the specified path ("/some/page.html").
  - The server receives the request, processes it, and sends an HTTP response back to the browser containing the requested resource (the HTML file).
  - The browser receives the response, parses the HTML, and renders the page for the user to view.

## Part Two: Practice Tools

The top panel shows a terminal window with the following commands and output:

```
marissa@DESKTOP-AC4JTDP:~$ curl -H "Accept: application/json" https://icanhazdadjoke.com/search?term=pirate
{"current_page":1,"limit":20,"next_page":1,"previous_page":1,"results":[{"id":"SvzIBAQS0Dd","joke":"What did the pirate say on his 80th birthday? Aye Matey!";{"id":"QuscibaMCLb","joke":"What does a pirate pay for his corn? A buccaneer!";{"id":"2gii3LeN7Ed","joke":"Why couldn't the kid see the pirate movie? Because it was rated arrr!";{"id":"SnOf2gqjiqc","joke":"Why are pirates called pirates? Because they arrr!";{"id":"exXSctk0Ke","joke":"Why do pirates not know the alpha bet? They always get stuck at \"C\".\"}]],\"search_term\":\"pirate\",\"status\":200,\"total_jokes\":5,\"total_pages\":1}
marissa@DESKTOP-AC4JTDP:~$ dig icanhazdadjoke.com

; <<>> DiG 9.18.1-lubuntu1.2-Ubuntu <<>> icanhazdadjoke.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 20630
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;icanhazdadjoke.com.      IN      A

;; ANSWER SECTION:
icanhazdadjoke.com.      0      IN      A      104.21.66.15
icanhazdadjoke.com.      0      IN      A      172.67.198.173

;; Query time: 0 msec
;; SERVER: 172.28.208.1#53(172.28.208.1) (UDP)
;; WHEN: Sun Feb 26 00:32:47 EST 2023
;; MSG SIZE rcvd: 86

marissa@DESKTOP-AC4JTDP:~$
```

The bottom-left panel shows a web browser at `localhost:8000/index.html` displaying:

Welcome to my web page!

This is a silly simple web page created using Python's built-in HTTP server.

The bottom-right panel shows a Visual Studio Code editor with the following `index.html` file:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="
6   <meta name="viewport" content="width=device
7   <title>My First Server</title>
8 </head>
9 <body>
10  <h1>Welcome to my web page!</h1>
11  <p>This is a silly simple web page created
12 </body>
13 </html>
```

Below the editor, a terminal window shows the following commands and output:

```
marissa@DESKTOP-AC4JTDP:~$ cd my-first-server
marissa@DESKTOP-AC4JTDP:~/my-first-server$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [26/Feb/2023 00:43:28] "GET /index.html HTTP/1.1" 200 -
127.0.0.1 - - [26/Feb/2023 00:43:28] code 404, message File not found
127.0.0.1 - - [26/Feb/2023 00:43:28] "GET /Favicon.ico HTTP/1.1" 404 -
```

## Part Three: Explore Dev Tools

The image is a composite of four screenshots arranged in a 2x2 grid, illustrating web development tools and HTTP requests.

**Top Left: Browser View (Successful GET Request)**  
The browser window shows a page titled "Welcome to my web page!". The address bar displays the URL: `localhost:8000/index.html?name=Linus&email=linusizkool%40gmail.com`. The developer tools are open to the Network tab, showing a single request to `localhost:8000/index.html?name=Linus&email=linusizkool%40gmail.com`. The request method is GET, and the status code is 304 (Not Modified). The response headers show a date of Sun, 26 Feb 2023 06:08:17 GMT and a server of SimpleHTTP/0.6 Python/3.10.6.

**Top Right: Code Editor View (index.html)**  
The code editor shows the `index.html` file. The HTML structure includes a DOCTYPE declaration, a meta charset of UTF-8, a meta http-equiv of X-UA-Compatible, a meta viewport, a title "My First Server", and a body containing a welcome message and a form with name and email input fields.

**Bottom Left: Browser View (Error Response)**  
The browser window shows an "Error response" page. The error code is 501, and the message is "Unsupported method ('POST')". The error code explanation states: "HTTPStatus.NOT\_IMPLEMENTED - Server does not support this operation." The developer tools show a request to `localhost:8000/index.html` with a POST method, resulting in a 501 status code.

**Bottom Right: Code Editor View (index.html)**  
The code editor shows the `index.html` file. The HTML structure is identical to the top right, but the form method is set to "POST". The terminal window below the code editor shows the output of the `python3 -m http.server` command, indicating that the server is serving HTTP on port 8000 and showing the log of the POST request that resulted in a 501 error.

## Part Four: Explore the URL API

