When you enter an HTTP URL into your browser and hit enter, a series of steps take place behind the scenes to retrieve the requested webpage. Here’s an overview of the process:

**1. Entering the URL**

You type an HTTP URL (e.g., http://example.com) into your browser's address bar and press enter. The browser parses this URL into its components:

* **Protocol**: http (or https for secure communication).
* **Domain name**: example.com.
* **Path** (optional): /path/to/resource.

**2. Domain Name Resolution (DNS Lookup)**

The browser needs to convert the human-readable domain name (example.com) into an IP address. This involves:

1. Checking the browser's cache for the IP address.
2. Querying the operating system's DNS cache.
3. Contacting a DNS server to resolve the domain name into an IP address (e.g., 93.184.216.34).

**3. Establishing a Connection**

Once the IP address is resolved:

* The browser establishes a connection to the server using the IP address.
* For HTTP:
  + A **TCP connection** is initiated on port 80.
  + For HTTPS (secure):
    - A **TCP connection** is initiated on port 443.
    - A **TLS/SSL handshake** occurs to encrypt communication.

**4. Sending an HTTP Request**

The browser sends an HTTP request to the server. The request includes:

* **HTTP method**: Typically GET, to fetch resources.
* **Headers**: Metadata about the request (e.g., user-agent, cookies, accepted content types).
* **Path**: Specifies the resource being requested (e.g., /path/to/resource).

Example of an HTTP request:

GET /path/to/resource HTTP/1.1

Host: example.com

User-Agent: Mozilla/5.0

**5. Server Processes the Request**

The server:

1. Receives the request.
2. Identifies the requested resource.
3. Prepares a response, often by:
   * Retrieving a static file.
   * Executing backend code (e.g., a Python or PHP script) to generate dynamic content.

**6. Sending an HTTP Response**

The server sends an HTTP response back to the browser, which includes:

* **Status Code**: Indicates success (200 OK), redirection (301 Moved Permanently), or errors (404 Not Found).
* **Headers**: Information about the response (e.g., Content-Type: text/html).
* **Body**: The actual content, such as HTML, CSS, JavaScript, or images.

Example of an HTTP response:

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 1234

<html>

<body>Hello, world!</body>

</html>

**7. Browser Renders the Page**

The browser processes the response:

1. Parses the HTML to build a **Document Object Model (DOM)**.
2. Sends additional requests for resources (e.g., CSS, JavaScript, images).
3. Applies CSS rules to style the page.
4. Executes JavaScript to add interactivity.
5. Displays the final webpage.

**8. User Interaction and Follow-up Requests**

* The browser remains connected to the server (using **keep-alive** for HTTP/1.1) for subsequent requests.
* Additional interactions (e.g., clicking a link) repeat this process.

**Visual Summary:**

* **URL Input → DNS Resolution → TCP Connection → HTTP Request → Server Processing → HTTP Response → Page Rendering**



