How the Internet Works

A Developer's Perspective

Introduction

As developers, understanding how the internet works beyond writing code is crucial. It gives us a deeper appreciation of the systems we interact with daily. This presentation will break down the internet from a developer’s perspective, covering concepts like URLs, URIs, DNS, protocols, and the intricate parts of internet addresses.

1. The Internet: A Brief Overview

The internet is a global network of computers communicating with each other using standardized protocols. It connects devices worldwide, allowing them to share data and resources.

- **Client-Server Model**: Most internet interactions follow this model, where a client (e.g., browser) sends requests to a server, and the server responds with the requested data.
- **IP Address**: Every device on the internet has a unique identifier called an IP address, which is used to locate and communicate with other devices.

2. URL vs. URI

What is a URL?

A **URL (Uniform Resource Locator)** is a specific address that identifies a resource on the internet and provides the means to retrieve it.

What is a URI?

A **URI (Uniform Resource Identifier)** is a broader term that can refer to either a URL or a URN (Uniform Resource Name). All URLs are URIs, but not all URIs are URLs.

URI	URL	URN
Identifies a resource	Identifies the location	Identifies the name
Can be a URL or URN	Always includes a protocol	No retrieval information

3. DNS (Domain Name System)

The **DNS** translates human-readable domain names (e.g., www.example.com) into IP addresses that computers use to identify each other.

How DNS Works:

1.User types a URL: The browser sends a request to a DNS resolver.

2.DNS Resolver queries: The resolver checks if the IP address is cached; if not, it queries other DNS servers.

3.DNS Server responds: It returns the IP address corresponding to the domain name.

4.Browser connects: The browser uses the IP address to establish a connection with the server.

4. Protocols

HTTP and HTTPS

not secured

•**HTTP (HyperText Transfer Protocol)**: A protocol for transferring web pages and other resources over the internet.

secured

•**HTTPS (HTTP Secure)**: Adds encryption (via SSL/TLS) to secure the communication between the client and server.

Other common protocols:

•**FTP (File Transfer Protocol)**: For transferring files.

•**SMTP (Simple Mail Transfer Protocol)**: For sending emails.

•**IMAP/POP3**: For receiving emails.

5. Anatomy of a URL

Example URL:

https://www.example.com:8080/path/to/page.html?query=123#section1

Part	Description
Protocol	Specifies the communication protocol (e.g., https)
Domain	The human-readable name of the server (e.g., www.example.com)
Port	Optional. Specifies the port number (e.g., 8080)
Path	The location of a resource on the server (e.g., /path/to/page.html)
Query String	Optional. Contains key-value pairs for dynamic content (e.g., ?query=123)
Fragment (Anchor)	Optional. Refers to a specific section of a page (e.g., #section1)

Explanation of Each Part:

1.Protocol: Defines the set of rules for communication. Common protocols include http, https, ftp, and mailto.

2.Domain: The name registered with a domain registrar, which maps to an IP address.

3.Port: A number indicating the entry point for communication on the server. By default, http uses port 80, and https uses port 443.

4.Path: Indicates the specific file or resource requested on the server.

5.Query String: Used to pass additional data to the server in a key-value format.

6.Fragment: Refers to a specific section within the page, typically used for navigation.

6. Understanding Domain Names

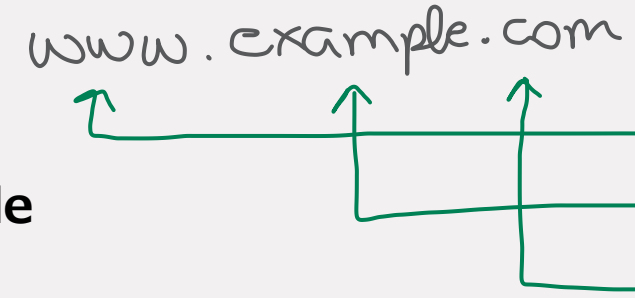
A domain name is divided into several parts:

Example: www.example.com

Common TLDs:

- **.com**: Commercial
- **.org**: Organization
- **.net**: Network
- **.edu**: Education
- **.gov**: Government

.fi — Finland
.es — Spain
.fr — France



Part	Description
www	Subdomain
example	Second-level domain (SLD)
com	Top-level domain (TLD)

7. Ports in Detail

Ports are numerical identifiers for specific processes on a server.

Common Ports:

Port Number	Service
80	HTTP
443	HTTPS
21	FTP
25	SMTP
3306	MySQL Database
8080	Alternative HTTP

8. Putting It All Together

When a user types a URL into their browser, this is what happens behind the scenes:

1. **The browser parses the URL** and identifies the protocol, domain, and other components.
2. **A DNS lookup is performed** to resolve the domain name to an IP address.
3. **The browser establishes a connection** with the server using the specified protocol and port.
4. **The server processes the request** and sends back the requested resource (e.g., HTML page).
5. **The browser renders the page**, processing HTML, CSS, and JavaScript to display it to the user.