**API Research**

1. **Research about How API Works**
2. **Client Sends a Request**:
3. A program (like a mobile app or website) makes a request to an API.
4. This request usually includes:
   1. A **URL** (what you're asking for)
   2. A **method** (e.g., GET, POST)
   3. Optional **data** (e.g., login info, new data to save)
   4. Optional **headers** (e.g., authentication keys)
5. **API Server Processes the Request**:
6. The API receives the request and decides what to do with it.
7. It may talk to a **database** or **another service** behind the scenes.
8. **Server Sends a Response**:
9. The API returns a **response**, typically in **JSON** or **XML** format.
10. This response includes:
    1. The **status** (e.g., success or error)
    2. Any **data** requested (e.g., user info, product list)
11. **What happen when you type google.com in browser and how you see the google page and every time you time gogole.com in the browser then always same process happen?**

Ans.**1. You Type google.com and Hit Enter**

The browser begins the process to locate and load the website.

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

* Your browser doesn't understand google.com; it needs an **IP address** (like 142.250.190.78).
* So it asks a **DNS server**: "What’s the IP for google.com?"
* If the answer isn't cached (in your browser, OS, or router), it reaches out to the internet's DNS hierarchy.
* Result: DNS server replies with Google’s IP address.

**3. Browser Connects to Google's Server (via TCP/IP)**

* Now your browser knows where to go.
* It opens a **TCP connection** to that IP address on **port 443** (since it's HTTPS).
* Then it performs a **TLS/SSL handshake** to secure the connection (this is what makes it HTTPS).

**4. HTTP Request is Sent**

* Your browser sends an **HTTP GET request** to Google’s server:

vbnet

CopyEdit

GET / HTTP/1.1

Host: www.google.com

**5. Google Server Processes the Request**

* Google’s servers receive the request.
* They generate the homepage, which includes:
  + HTML (structure)
  + CSS (styling)
  + JavaScript (behavior)
  + Images, logos, etc.

**6. Server Sends Back a Response**

* The response includes:
  + A status code (like 200 OK)
  + The requested HTML and other assets

**7. Browser Renders the Page**

* The browser parses the HTML and starts loading:
  + CSS (to style it)
  + JavaScript (to make it interactive)
  + Fonts, icons, etc.
* It builds the **DOM (Document Object Model)** and displays the page.

**8. You See Google’s Homepage**

You can now use the search box and interact with the page.

1. **How many Protocols are there for API/Network**.

Ans. **1. Web/API Communication Protocols**

These are the most common for APIs and web apps.

| **Protocol** | **Purpose** |
| --- | --- |
| **HTTP** (HyperText Transfer Protocol) | Foundation of web communication. Used for REST APIs. |
| **HTTPS** (HTTP Secure) | Encrypted version of HTTP (uses SSL/TLS). |
| **WebSocket** | For real-time, two-way communication (chat, games). |
| **GraphQL** | Query-based API protocol (alternative to REST). |
| **gRPC** | High-performance API communication using Protocol Buffers. |
| **SOAP** | XML-based protocol for older enterprise APIs. |

**🔷 2. Data Transfer Protocols**

Used to send files or data across the internet.

| **Protocol** | **Purpose** |
| --- | --- |
| **FTP** (File Transfer Protocol) | Transfer files between systems. |
| **SFTP** (Secure FTP) | Encrypted file transfers. |
| **TFTP** (Trivial FTP) | Very simple, no authentication, used in network booting. |

**🔷 3. Network Communication Protocols**

These are the "low-level" protocols that power the internet.

| **Protocol** | **Purpose** |
| --- | --- |
| **IP** (Internet Protocol) | Sends packets across networks. |
| **TCP** (Transmission Control Protocol) | Reliable connection (used in HTTP). |
| **UDP** (User Datagram Protocol) | Faster, no guarantee of delivery (used in streaming). |
| **ICMP** | Used by tools like ping (for diagnostics). |

**🔷 4. Email Protocols (For APIs that send emails)**

| **Protocol** | **Purpose** |
| --- | --- |
| **SMTP** | Sends emails. |
| **IMAP** | Reads emails from a server. |
| **POP3** | Downloads emails to your device. |

**🔷 5. Authentication & Security Protocols**

| **Protocol** | **Purpose** |
| --- | --- |
| **OAuth 2.0** | Secure user login for APIs. |
| **JWT (JSON Web Token)** | Used to send user identity and permissions. |
| **TLS/SSL** | Encrypts data (used in HTTPS). |

1. **What is the full form of API?**

Ans. Application Programming Interface

**5. Full form of URL.**

Ans. Uniform Resource Locator

1. **What is the max length of the URL?**

Ans. **Maximum URL Length (In Practice)**

| **Environment** | **Max Length** | **Notes** |
| --- | --- | --- |
| **Internet Explorer** | **2,083 characters** | This is a commonly cited practical limit. |
| **Chrome, Firefox, Safari** | **~32,000+ characters** | Can handle much longer URLs, but not always recommended. |
| **Apache Server (Default)** | ~8,192 bytes | Configurable via LimitRequestLine. |
| **NGINX Server (Default)** | ~4,096 bytes | Can be changed via large\_client\_header\_buffers. |
| **RFC (Standard)** | No strict limit | The HTTP standard (RFC 2616) doesn’t define a max URL length. |

**⚠️ Best Practice**

Stay under **2,000 characters** to ensure compatibility with all browsers and servers.

**7. How or when the IP address update of our mobile or laptop device (optional extra point)?**

Ans. Your mobile or laptop usually gets an **IP address** automatically from a network (like Wi-Fi or mobile data) using **DHCP (Dynamic Host Configuration Protocol)**.

**✅ IP Address Can Change When...**

| **Situation** | **What Happens** |
| --- | --- |
| **You switch Wi-Fi networks** | New network = new local IP address (e.g., home → café Wi-Fi). |
| **Your router restarts** | It may give you a new IP via DHCP. |
| **Lease time expires** | Routers assign IPs for a limited time (called a *DHCP lease*). When that time ends, you might get a new one. |
| **You toggle airplane mode or restart your phone** | This resets the connection, often triggering a new IP. |
| **Your ISP changes your public IP (for home Wi-Fi)** | ISPs assign dynamic public IPs that can change after reboot or over time. |
| **You switch from Wi-Fi to mobile data** | You'll get a new IP from your mobile network provider. |