

Http protocol basics:

- HTTP = **Hypertext Transfer Protocol**

The HTTP is an application layer protocol that allows web-based applications to communicate and exchange data.

- the HTTP is the messenger of the web
- It is a **TCP/IP** based protocol
- It is used to deliver contents, for example, images, videos, audios, documents, etc.
- The computers that communicate via the HTTP must speak the http protocol.

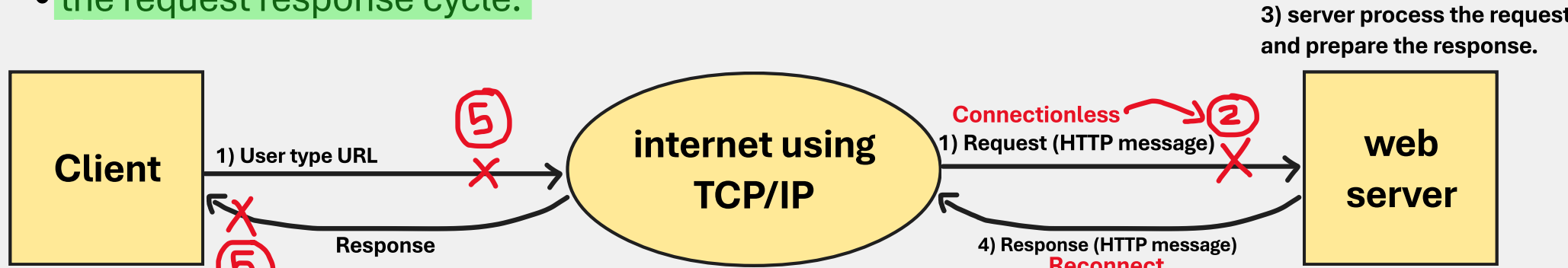
- **Three Important things about the HTTP:**

- 1 - **HTTP is connectionless:** after making the request, the client disconnect from the server, then when the response is ready the server re-establish the connection again and deliver the response.
- 2 - **The HTTP can deliver any sort of data**, as long as the two computers are able to read it.
- 3 - **The HTTP is a stateless:** The client and server know about each other just during the current request, if it closes, and the two computers want to connect again, they need to provide information to each other anew, and the connection is handled as the very first one.

- **Why The HTTP?**

- The http was designed mainly to fetch html documents and sends it to the client.
- It was designed in an exquisite way.
- It was being continually evolved and features were being added to it.
- It became the most convenient way to quickly and reliably move data on the web.

- **the request response cycle:**

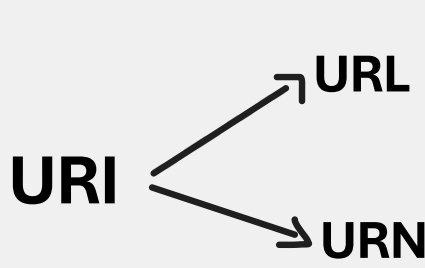


URI's Vs. URL's Vs. URN's

- **URI = Uniform Resource Identifier**

It is a string of characters used To identify a resource on the internet either by location or by name, or both.

- For Example:
Name: Mohammad
Address: Amman



- **URL = Uniform Resource Locator**

It is a string of characters but it refers to just the address. It is the most used way to locate resources on the web.

- For Example:
Address: Amman

- **URN = Uniform Resource Name**

- For Example:
Name: Mohammad

- **URLs in deep details**

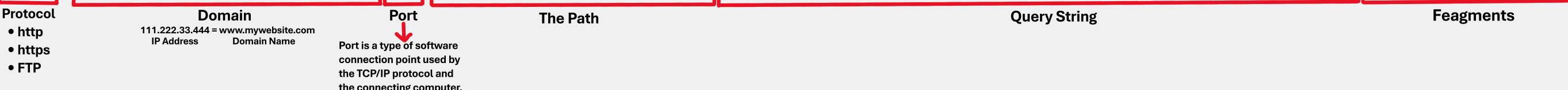
A URL consists of two required components:

- The Protocol
- The Domain

- Other Optional components:

- The Path
- The Port
- Others (e.g., query parameters, fragment)

http://www.mywebsite.com:80/product/product.php?first_Name:Mohammad&second_Name=Ismail#External_Links

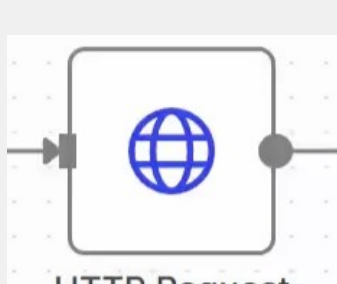
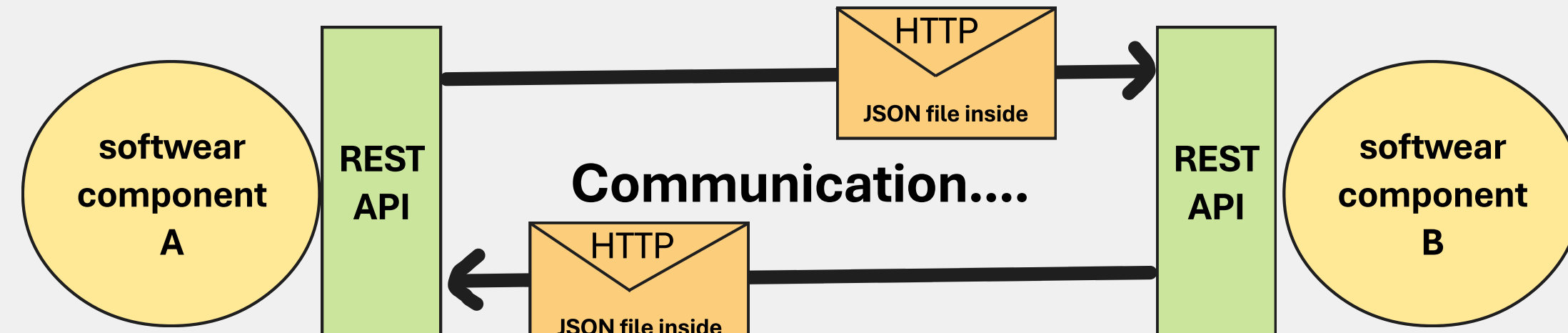


API: Application Programming interface

- An API (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate and exchange information.



- REST API (Representational State Transfer) is a specific, widely-used type of API that adheres to the REST architectural style for web services.
- It uses HTTP methods suitably (GET for getting data, POST for creating...)
- Scoping information (and other data) goes in the parameters part of the URI.
- It uses common data formats (most commonly JSON).
- Communication is stateless.



n8n Http Request node:

- in order to connect to an API using n8n we have 7 questions we must answer:

1) method: how API's work (GET, POST) -x

- 1.1) GET: GET data
- 1.2) POST: Create new data
- 1.3) PUT: Update entire data
- 1.4) PATCH: Update part of the data
- 1.5) DELETE: Delete data

2) End Point (URL): API body (Curl)

Ex:

- **https://api.example.com/v1/**

3) Path Resource Path Parameters

Ex:

- **https://api.example.com/v1/users/123/ord/456**
- users: the Path resource
- 123: the Path parameter
- ord: nested path resource
- 456: Another path parameter

4) Authentication

- 4.1) API Key (Passed in headers or query)
- 4.2) Bearer token (OAuth 2.0)
- 4.3) Basic Auth (username: password)

5) Query Parameters (variables, questions, filters)

Ex:

- **https://api.example.com/products?category=Shoes&sort=price_desc&page=2**
- After question mark we send Query Param's. For example:
- 1) category=Shoes means variable
- 2) sort=price_desc means variable
- 3) page=2 means variable

6) header Parameters (Used to pass metadata like): -H

6.1) Authentication

6.2) Content type

6.3) ... etc

- Common headers

- Authorization: Bearer token, API key, etc.
- Content-Type: Usually application/json
- Accept: Expected response format
- User-agent: Client information
- X-Request-ID: Tracking / logging ID

7) Body Parameters (Data, Payload) -d

- Used in POST, PUT, PATCH to send data mostly in JSON

```
{
  "name": "mohammad",
  "email": "mohammad@gmail.com",
  "password": "1234"
}
```

- EndPoint (URL) & CURL:

- **CURL:** Client URLIs you send Curl you can use it to make or write API automatically instead of writing it manually.

- Any HTTP request must have response:

- 1) Status code
- 2) Header's
- 3) Body (often JSON)

Status code:.....

Codes	meaning
200	OK
201	Created
204	No content
400	Bad request
401	Unauthorized
403	Forbidden
404	Not found
500	Server error