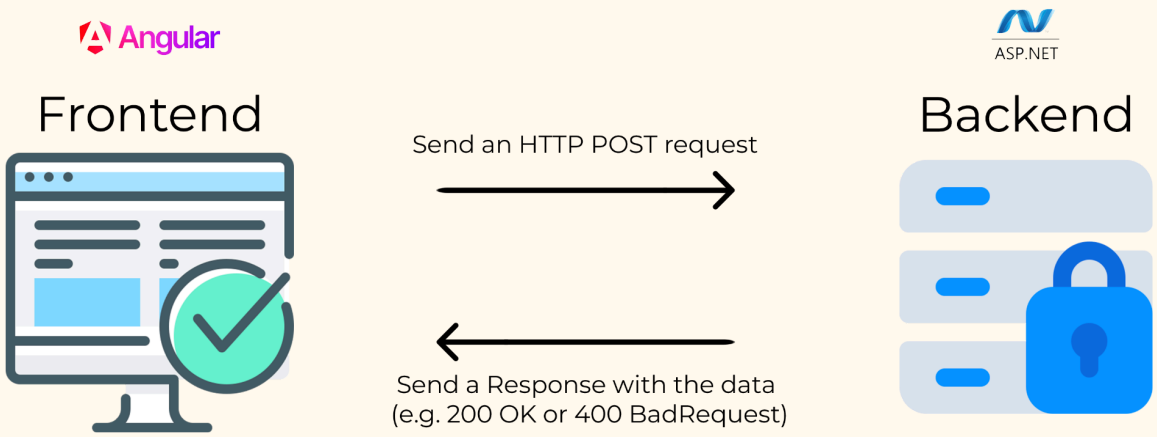
This cheat sheet is for the course [Learn C# Full Stack Development with Angular and ASP.NET] by Jannick Leismann.

HTTP REQUEST AND RESPONSE

HTTP stands for Hypertext Transfer Protocol

Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**



Communication between clients and servers is done by requests and responses:

1. A client (a browser) sends an HTTP request to the web
2. A web server receives the request
3. The server runs an application to process the request
4. The server returns an HTTP response (output) to the browser
5. The client (the browser) receives the response

HTTP Request

Is sent by a client to a server to request resources such as HTML files, images, or data from a server. It consists of several components:

1. **Request Line**

   Method: Specifies the action to be performed (e.g., **GET**, **POST**, **PUT**, **DELETE**).

URL: The resource being requested.

HTTP Version: The version of the HTTP protocol being used (e.g., HTTP/1.1)

2. **Headers**

   Information about the request, including the kind of material being transmitted or requested, the client submitting the request, and the appropriate course of action.

   Common Headers:

   Host: The domain name of the server (e.g., Host: www.example.com).

   User-Agent: Information about the client (e.g., User-Agent: Mozilla/5.0).

   Accept: Types of content the client can handle (e.g., Accept: text/html).

   Content-Type: The type of content being sent (for POST or PUT requests).

3. **Body (Optional)**

   Contains data being sent to the server, typically used with POST and PUT requests.

   JSON Payload Example:

```
{
  "username": "user",
  "password": "pass"
}
```

HTTP Response

A server replies to a client over HTTP with the resources the client requested or details about the request's status.

1. **Status Line:**

   HTTP Version: The version of the HTTP protocol being used.

   Status Code: Indicates the result of the request (e.g., 200 OK, 404 Not Found).

   Status Message: A short description of the status code.

```
HTTP/1.1 200 OK
```

2. **Headers**

Metadata about the response, such as content type, length, and how to cache the response.

Common Headers:

Content-Type: The type of content being returned (e.g., Content-Type: text/html).

Content-Length: The length of the content being returned (e.g., Content-Length: 1234).

Set-Cookie: Cookies to be stored by the client.

3. **Body**

   Contains the actual content being returned, such as an HTML page, JSON data, or an image.

Example Request

```
GET /api/users HTTP/1.1

Host: www.example.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

Accept: application/json
```

Example Response

```
HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 85

{

  "users": [

    {"id": 1, "name": "John Doe"},

    {"id": 2, "name": "Jane Smith"}

  ]

}
```

HTTP Methods

GET

Requests data from a specified resource.

Use Case: Retrieving data without modifying it.

POST

Sends data to the server to create a new resource.

Use Case: Submitting form data or uploading a file.

PUT:

Updates an existing resource with the provided data.

Use Case: Updating user information.

DELETE:

Deletes the specified resource.

Use Case: Removing a user account.

HEAD

Similar to GET, but it only requests the headers, not the body.

Use Case: Checking if a resource exists or getting metadata.

OPTIONS

Returns the HTTP methods that the server supports for the specified URL.

Use Case: Checking allowed operations on a resource.

PATCH

Applies partial modifications to a resource.

Use Case: Updating a part of a user's profile.

Status Codes

2xx Success

200 OK: The request was successful.

201 Created: The request was successful and a new resource was created.

3xx Redirection

301 Moved Permanently: The resource has been permanently moved to a new URL.

302 Found: The resource has temporarily moved to a different URL.

4xx Client Errors

400 Bad Request: The request could not be understood or was missing required parameters.

401 Unauthorized: Authentication is required and has failed or not been provided.

403 Forbidden: The server understood the request but refuses to authorize it.

404 Not Found: The requested resource could not be found.

5xx Server Errors

500 Internal Server Error: An error occurred on the server.

503 Service Unavailable: The server is not ready to handle the request, usually due to maintenance or overload.