LU11a - CRUD and HTTP Methods

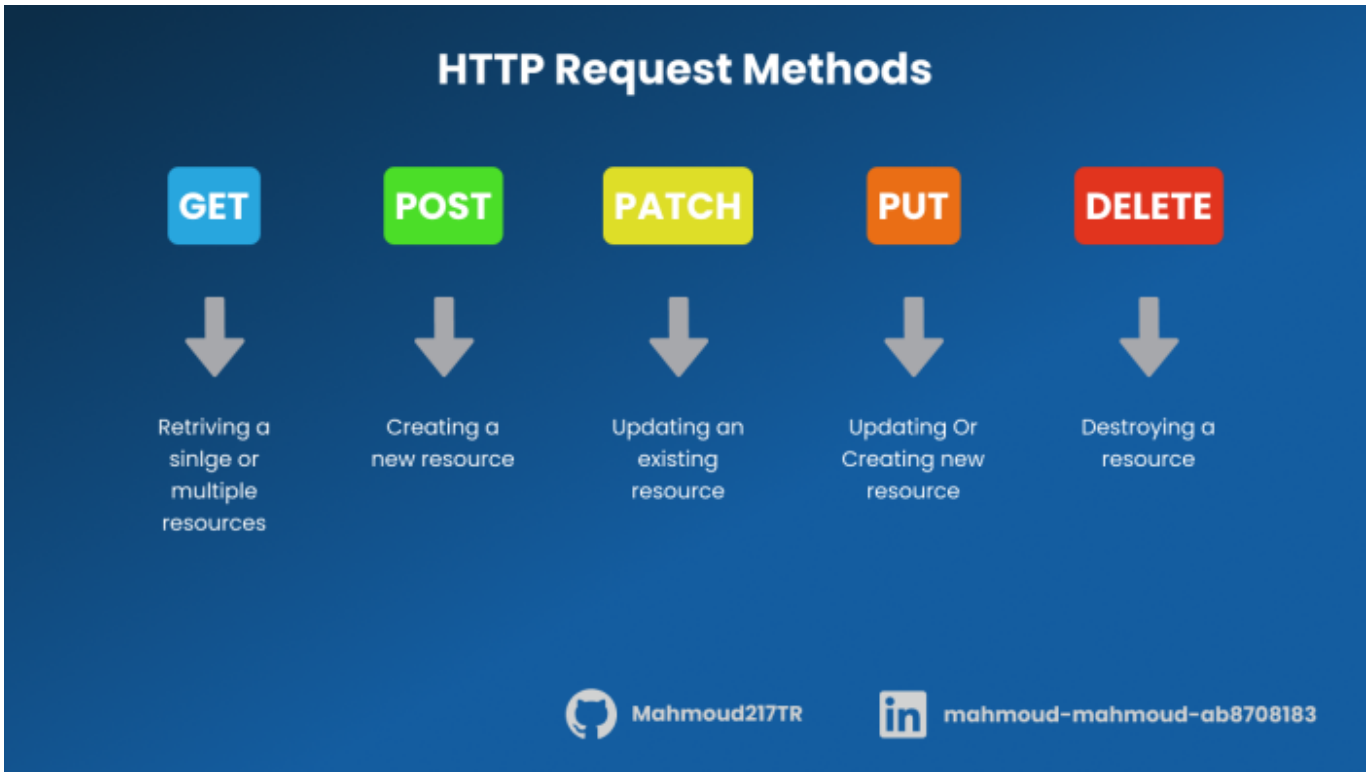
Source: W3Schools | HTTP Methods

Learning Objectives

1. Describe the four module relevant HTTP Methods
2. Explain at least three differnences between the POST and GET method
3. Asign each of the CRUD operations to one of the four main http request methods

Introcuction

HTTP (Hypertext Transfer Protocol) methods are fundamental to the communication between clients (like web browsers) and servers in web applications. They allow the client to perform various operations on the server, such as retrieving, creating, updating, or deleting resources. The most commonly used HTTP methods are POST, GET, PUT, and DELETE. Each has a specific purpose and behavior in how it interacts with the server.



During this module (290) we are going to use 4 of them. Let’s dive into each method and see practical examples of how they are used.

1. GET - Retrieving Data

The GET method is used to retrieve data from the server. It requests a resource without altering the

data on the server. Since it does not modify data, it is considered a „safe“ and „idempotent“ operation (multiple identical GET requests will produce the same result).

Practical Example:

Imagine a website displaying a list of products. To retrieve the list of products from a server, the client would use a GET request.

Request:

```
http
GET /products
```

Example URL:

```
https://api.example.com/products
```

This would fetch the list of products from the server and display them to the user.

2. POST - Creating Data

The POST method is used to submit data to the server to create a new resource. Unlike GET, which just retrieves data, POST modifies the server’s data by adding something new. It’s often used in forms where a user submits information to create a new record.

Practical Example:

Let’s say a user wants to add a new product to a database. They would submit a POST request containing the product data.

Request:

```
http
POST /products
Content-Type: application/json
{
  "name": "New Product",
  "price": 29.99
}
```

Example URL:

```
https://api.example.com/products
```

The server would process this request and create the new product in the system.

3. PUT - Updating Data

The PUT method is used to update an existing resource on the server. It is idempotent, meaning if you send the same PUT request multiple times, the result will be the same each time. PUT typically requires sending the entire entity, even if only a small part of it is being updated.

Practical Example:

If a user wants to update the price of an existing product, they would send a PUT request with the updated information.

Request:

```
http
PUT /products/123
Content-Type: application/json
{
  "name": "Updated Product",
  "price": 24.99
}
```

Example URL:

```
https://api.example.com/products/123
```

Here, the product with ID 123 is updated with the new name and price.

4. DELETE - Removing Data

The DELETE method is used to delete a specified resource from the server. Like PUT, it is idempotent, meaning if you send the same DELETE request multiple times, the result (the resource being deleted) will be the same.

Practical Example:

If a user wants to remove a product from the database, they would issue a DELETE request specifying the resource to be deleted.

Request:

```
http
DELETE /products/123
```

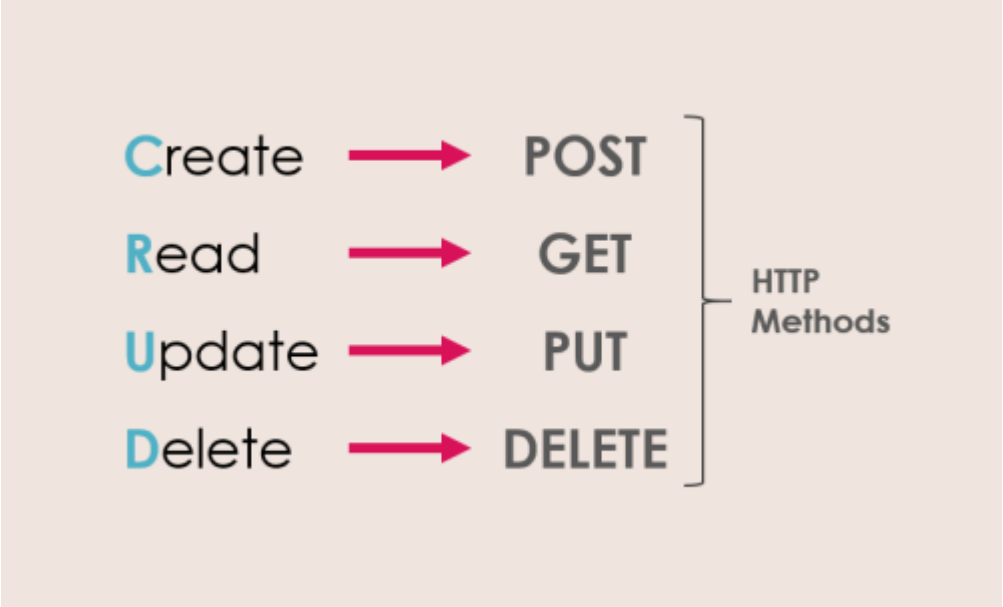
Example URL:

```
https://api.example.com/products/123
```

This would delete the product with ID 123 from the server.

Correlation Between CRUD and HTTP Methods

CRUD (Create, Read, Update, Delete) refers to the basic operations that can be performed on data in most applications, particularly databases. These operations align with common actions in managing resources. HTTP methods—such as POST, GET, PUT, and DELETE—are used in web development to interact with a server to perform these CRUD operations on web-based resources.



Vocabulary

English	Deutsch
to fetch	abrufen, herholen
to submit	zustellen, etwas vorlegen
to perform	ausführen