# 13 HTTP verbs

## GET

Retrieves data from the server

## POST

**Sends data to server to create a resource.**

## PUT

Sends data to server to update a resource.

## PATCH

Sends data to server to update a resource partially.

## DELETE

Deletes a resource from server.

## PURGE

Invalidates a cached resource.

## TRACE

Returns the full HTTP request received by the server, for debugging purpose.

## OPTIONS

Returns HTTP methods supported by the server for the requested URL.

## CONNECT

Converts the request connection to a transparent TCP/IP tunnel for secure communication.

## LOCK

Locks the resource for the exclusive use by the client.

## UNLOCK

Unlocks the resource which is previously locked by the client.

## MKCOL

Creates a new collection resource.

## Need to work.

See what it means to create a new collection resource.

## COPY

Copies the resource identified by the request URI to the destination URI.

# HTTP request

An HTTP request is **a protocol used for sending and receiving information over the internet**

A standard HTTP request is **a message that a client (usually a web browser) sends to a server**, **requesting action or data**.

The basic structure of an HTTP request consists of a **request line, a set of headers, and an optional message body**.

The **request line** **contains**

* **the method** (e.g. GET, POST, PUT, DELETE), **the resource being requested** (e.g. a URL), and the **HTTP version being used**.

The headers provide additional information about the request,

* such as the user-agent,
* content type, and
* authorization credentials.

The message body,

* if present, contains data that is sent along with the request.

Here's an example of a basic HTTP request:

GET /example HTTP/1.1

Host: www.example.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

In this example, the client is requesting the resource located at "/example" on the server "www.example.com" using the GET method and the HTTP/1.1 protocol. The headers provide additional information about the request, such as the user-agent, which identifies the client software being used, and the Accept header, which specifies the types of data that the client can accept in response.

## What has a http request to do with an API

An api request is a tailored http request to request action/data from an API

## Post request

request.getlist(

request.GET.get

this is used to fetch query parameters

ex: in this url <http://localhost:8000/api/scenario_planner/selection?create_from_date=2021-11-11&create_to_date=2021-12-29> create\_from\_date and create\_to\_date can be fetched using the above method

ex1: request.GET.get(‘create\_from\_date’,None) will fetch create\_from\_date from the request URL, the None will be useful when you don’t have any values passed in the url, see ex2

ex2: for the same above URL if you try a=request.GET.get(‘promo\_id’,None) then a will have no value assigned because promo\_id is not passed in URL, it will assign nothing to a

request.data

request.object\_list

## Get request

# What is JWT and its need

■JSON Web Token is a self-contained way to securely

transmit data and information between two parties using a

JSON Object.

■JSON Web Tokens can be trusted because each JWT can

be digitally signed, which in return allows the server to

know if the JWT has been changed at all

■JWT should be used when dealing with authorization

■JWT is a great way for information to be exchanged

between the server and client

## JSON web token and structure

A JSON Web Token is created of three separate parts separated

by dots ( . ) which include:

■Header : (aaaaaaaa)

■Payload : (bbbbbbbb)

■Signature : (cccccccc)

A black rectangle with purple letters

Description automatically generated

# Status codes and their need

• An HTTP Status Code is used to help the Client (the user or system

submitting data to the server) to understand what happened on the

server side application.

• Status Codes are international standards on how a Client/Server should

handle the result of a request.

• It allows everyone who sends a request to know if their submission was

successful or not

### JWT HEADER

A JWT header usually consist of two

parts:

■ (alg) The algorithm for signing

■ “typ” The specific type of token

■The JWT header is then encoded using

Base64 to create the first part of the JWT (a)

A black rectangular object with colorful text

Description automatically generated

### JWT Payload

■A JWT Payload consists of the data.

The Payloads data contains claims, and

there are three different types of claims.

■ Registered

■ Public

■ Private

■The JWT Payload is then encoded using

Base64 to create the second part of the JWT

(b)

A computer screen shot of a code

Description automatically generated

### JWT SIGNATURE

■A JWT Signature is created by using the

algorithm in the header to hash out the

encoded header, encoded payload with a

secret.

■The secret can be anything, but is saved

somewhere on the server that the client does

not have access to

■The signature is the third and final part of a

JWT (c)

A black background with blue text

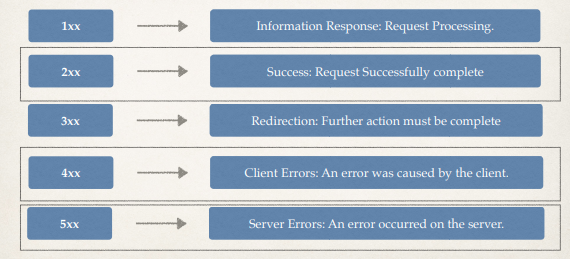
Description automatically generated

### JWT Example

A screenshot of a computer program

Description automatically generated

## Major classification of status codes



### 2xx : Successful status codes

A diagram of a diagram

Description automatically generated

### 4xx Client error status codes

A diagram of a process

Description automatically generated

### 5xx Server error status codes

A blue and white rectangular sign

Description automatically generated