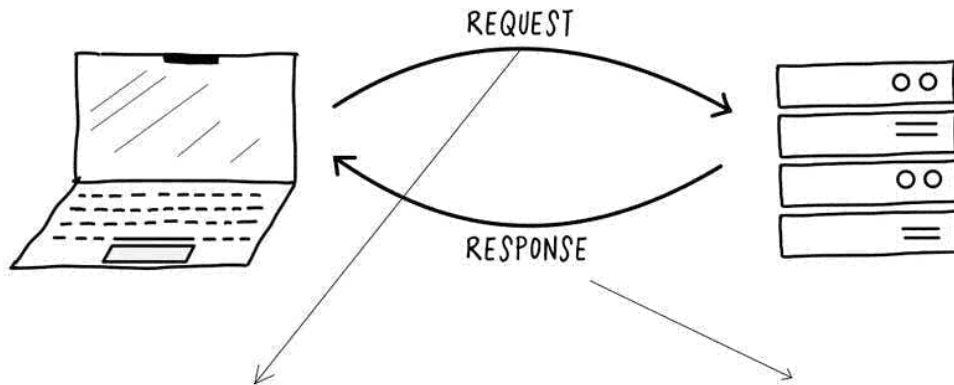


# REST API

REST STANDS FOR REPRESENTATIONAL STATE TRANSFER

REST APIS OPERATE ON A SIMPLE REQUEST/RESPONSE SYSTEM



CLIENT CAN MAKE A REQUEST USING HTTP METHODS

SERVER RETURNS A RESPONSE WITH AN HTTP STATUS CODE

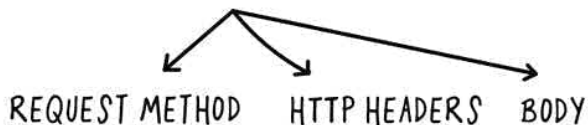
THESE METHODS ARE:

GET, POST, PUT, PATCH, DELETE, HEAD, TRACE, OPTIONS, CONNECT

POPULAR HTTP STATUS CODE:

EX, 200, 202, 403, 404, 500 ETC

HTTP REQUEST CONTAINS



HTTP RESPONSE CONTAINS



## ★ REST API CONSTRAINTS ★

### CLIENT-SERVER ARCHITECTURE

- NO THIRD PARTY INTERPRETATION

### UNIFORM INTERFACE

- FOLLOW A COMMON PROTOCOL

### LAYERING

- MULTIPLE INTERMEDIARIES BETWEEN CLIENT AND SERVER

### CACHEABILITY

- RESPONSE CAN BE CACHEABLE

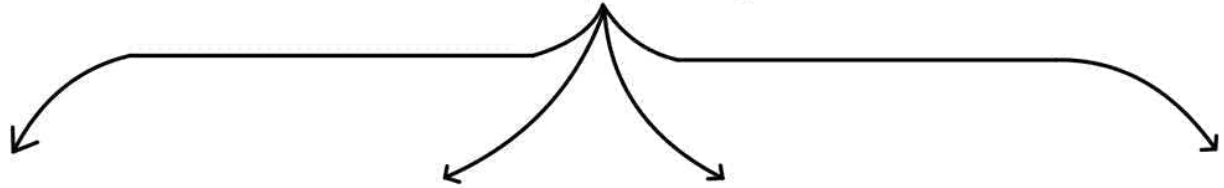
### STATELESSNESS

- THERE IS NO STATE. CLIENT AND SERVER ARE COMPLETELY SEPARATED

# HTTP HEADERS

CLIENT AND SERVER CAN PASS THE EXTRA BIT OF INFORMATION WITH THE REQUEST AND RESPONSE USING HTTP HEADERS

DIVIDED INTO FOUR PARTS



## REQUEST HEADERS

- CLIENT TO SERVER

## RESPONSE HEADERS

- SERVER TO CLIENT

## REPRESENTATION HEADERS

- INFORMATION ABOUT THE BODY OF THE RESOURCE

## PAYLOAD HEADERS

- INFORMATION ABOUT THE PAYLOAD DATA

## WIDELY USED HTTP HEADERS

### Accept

TYPE OF DATA CLIENT CAN UNDERSTAND

### Content-Type

SPECIFIES THE MEDIA TYPE OF THE RESOURCE

### Accept-Encoding

WHICH ENCODING METHOD CLIENT CAN UNDERSTAND

### Host

SPECIFIES THE DOMAIN NAME

### Authorization

USED TO PASS CREDENTIALS SO THAT SERVER CAN AUTHENTICATE

### Access-Control-Allow-Origin

WHICH ORIGIN IS ALLOWED TO ACCESS THE RESOURCES

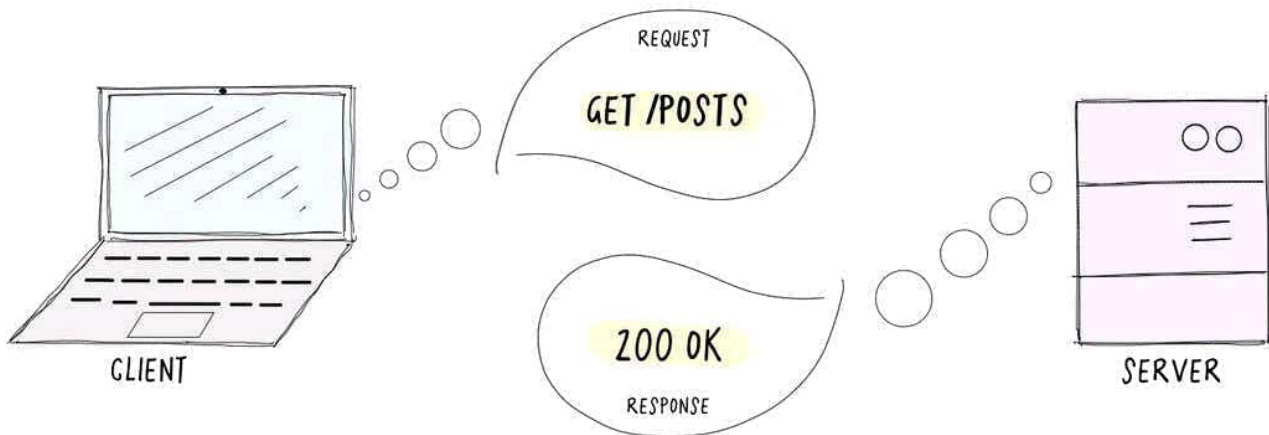
### Accept-Language

CLIENT IS EXPECTING THE RESPONSE IN THE MENTIONED LANGUAGE

### Access-Control-Allow-Methods

WHICH METHODS ARE ALLOWED TO ACCESS THE CROSS-ORIGIN RESOURCES

# HTTP STATUS CODES



★ SERVER ALWAYS RETURNS HTTP STATUS CODE WITH THE RESPONSE ★

## SUCCESSFUL RESPONSES

200 OK

EVERYTHING IS FINE

201 CREATED

NEW RESOURCE WAS CREATED

## REDIRECTION MESSAGES

301 MOVED PERMANENTLY

THE RESOURCE HAS BEEN MOVED PERMANENTLY  
TO THE NEW URL

## CLIENT ERROR

400 BAD REQUEST  
INVALID SYNTAX

401 UNAUTHORIZED  
CREDENTIALS ARE INCORRECT

403 FORBIDDEN  
YOU DON'T HAVE PERMISSION TO ACCESS THE RESOURCES

404 NOT FOUND  
INVALID URL

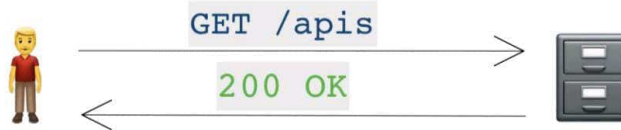
429 TOO MANY REQUESTS  
USER HAS SENT TOO MANY REQUESTS IN A  
GIVEN AMOUNT OF TIME

## SERVER ERROR

500 INTERNAL SERVER ERROR  
SERVER DOES NOT KNOW HOW TO HANDLE THE UNEXPECTED SITUATION



# HTTP Request Methods



## GET

The GET method is the most common of all these request methods.

It is used to fetch the desired resources from the server.

The POST method is used to submit the information to the server.

As we're submitting data, the POST method often changes the state of the server.

## POST

## PUT

The PUT method is used whenever you need to change the resource. The resource, which is already a part of resource collection.

The PATCH request method is used to modify only the necessary part of the data or response.

The PATCH method doesn't modify the entire response.

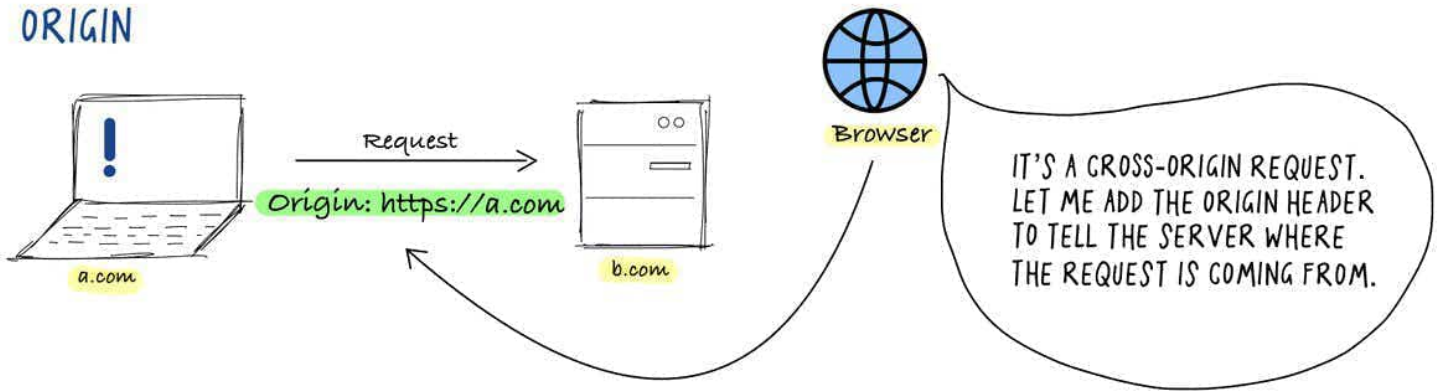
## PATCH

## DELETE

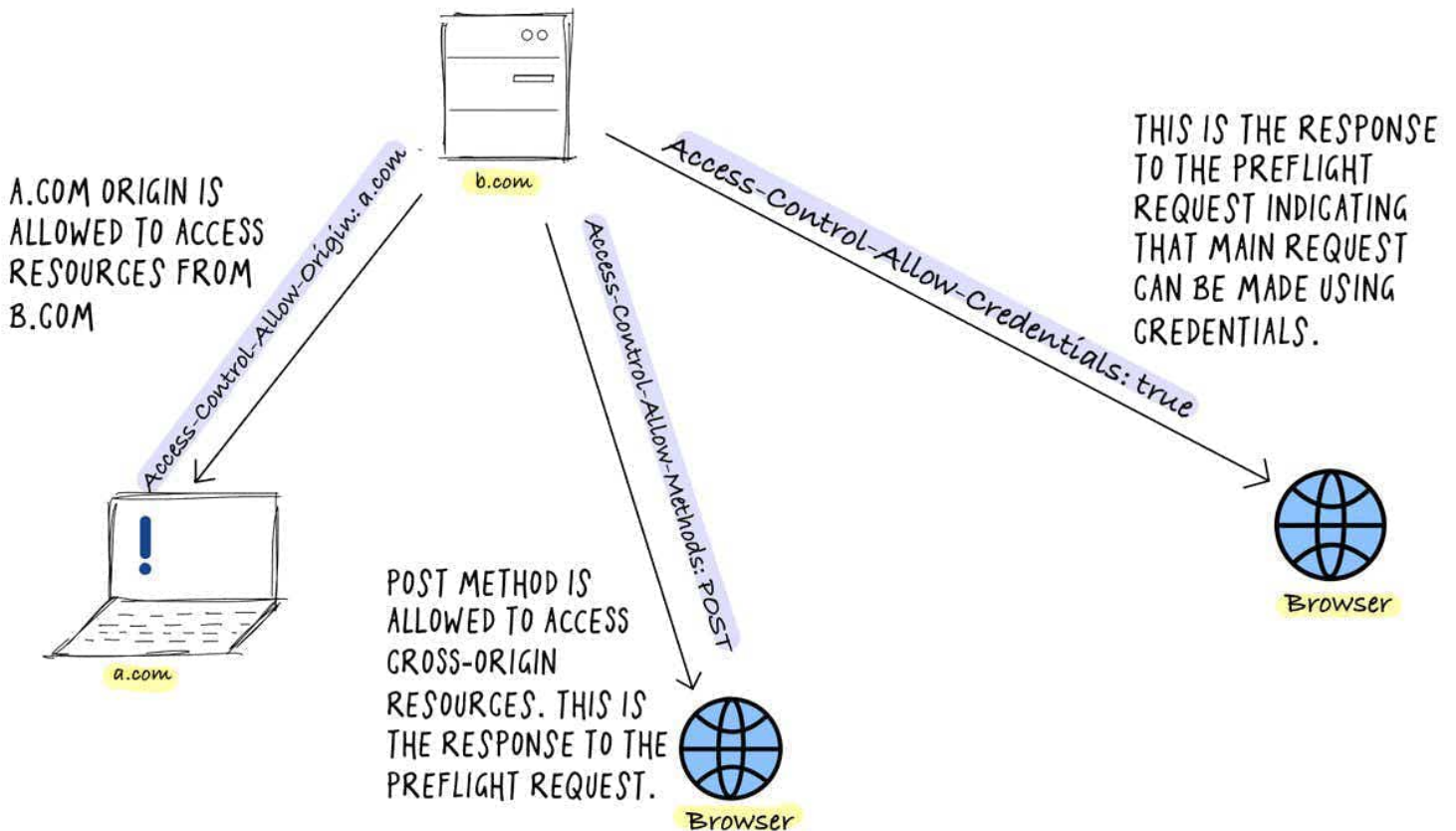
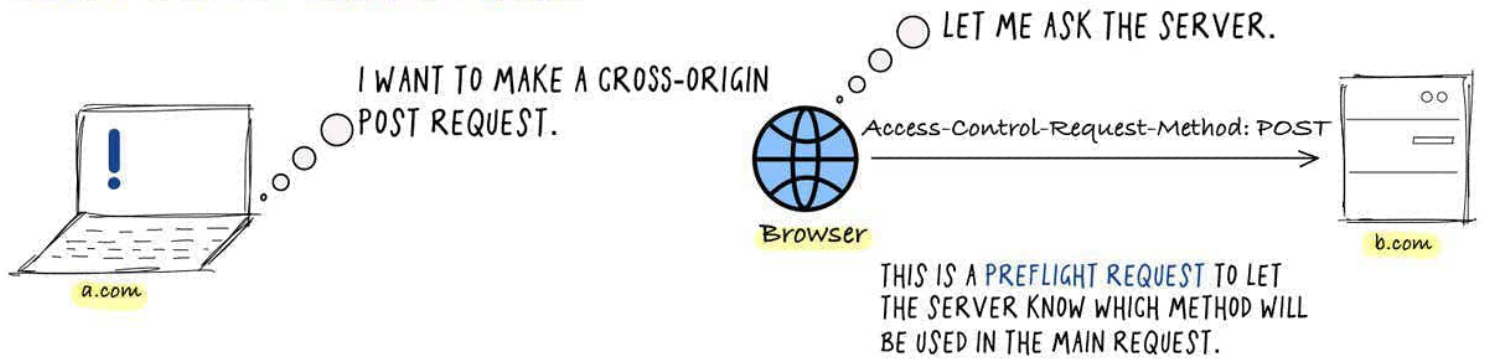
As the name says, the DELETE request method is used to delete the specified resource. It requests that the origin server delete the resource identified by the Request-URL.

# Access Control HTTP Headers

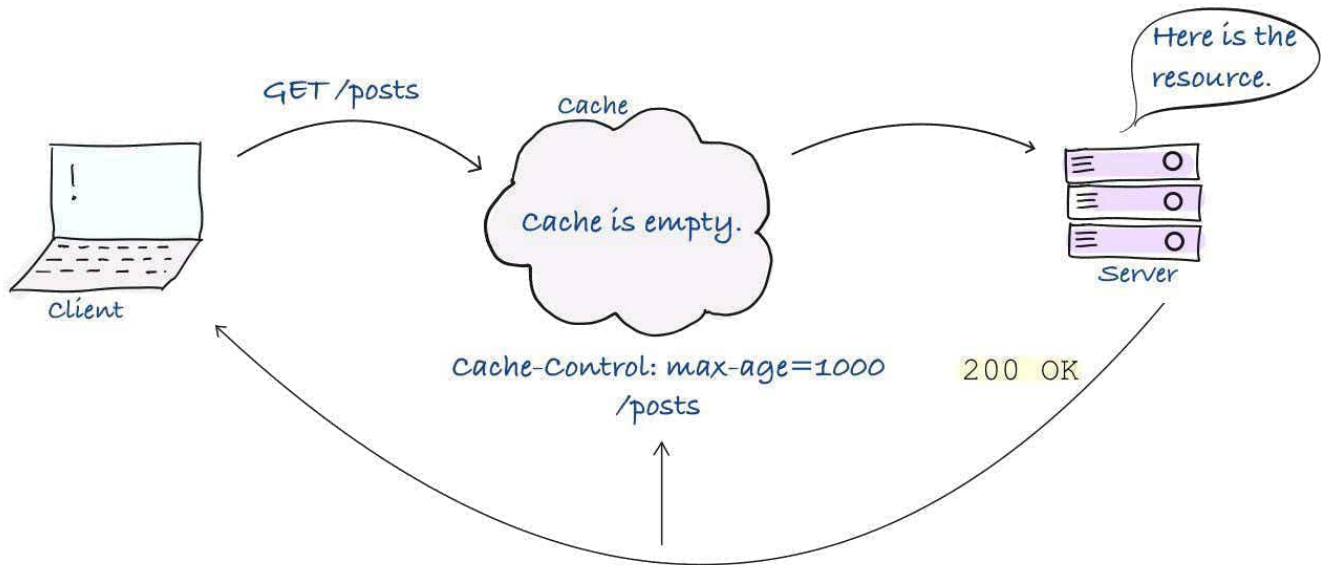
## ORIGIN



## ACCESS-CONTROL-REQUEST-METHOD



# Caching in API calls



After 200 Seconds

