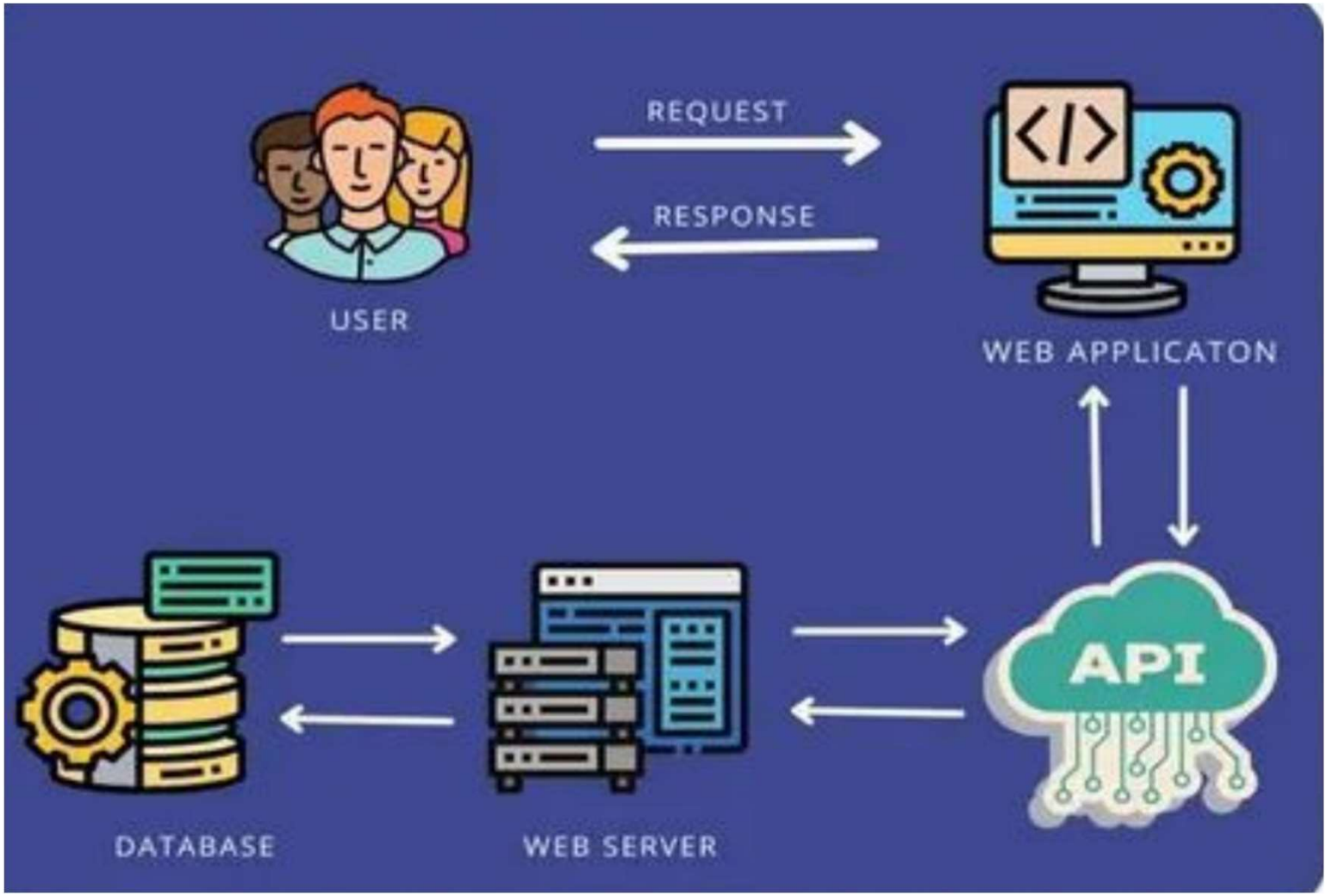
API

APPLICATION PROGRAMMING INTERFACE

API

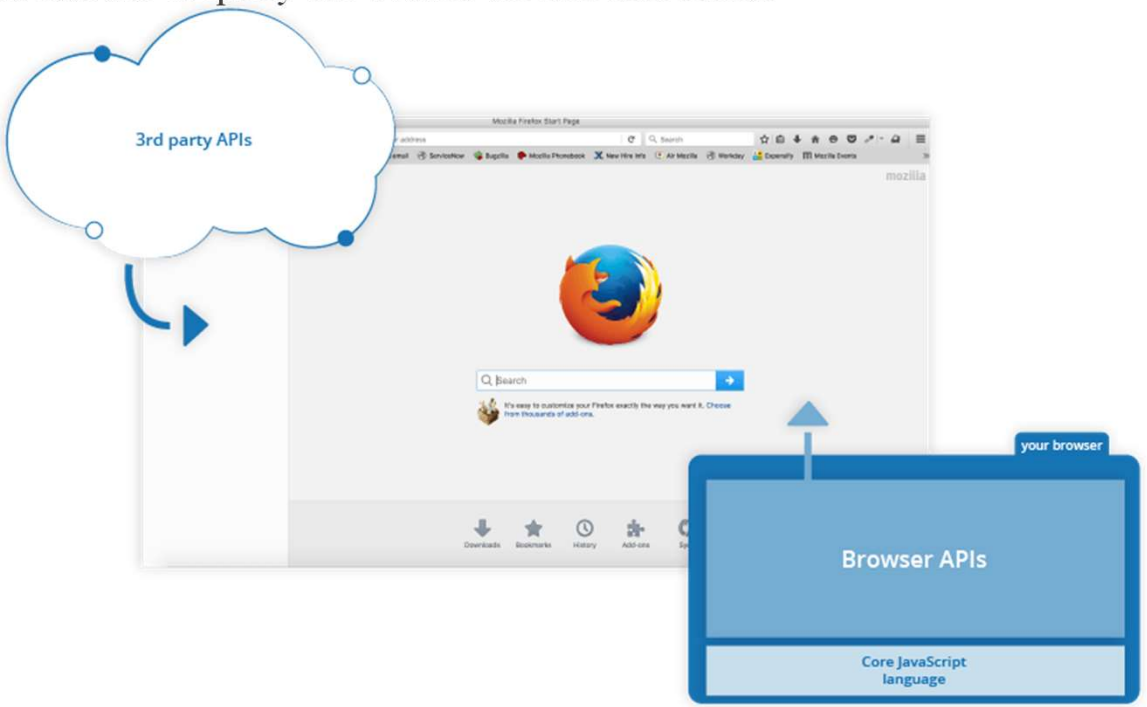
- Application Programming Interfaces (APIs) are constructs made available in programming languages to allow developers to create complex functionality more easily.
- They abstract more complex code away from you, providing some easier syntax to use in its place.
- Eg: As a real-world example, think about the electricity supply in your house, apartment, or other dwellings. If you want to use an appliance in your house, you plug it into a plug socket and it works. You don't try to wire it directly into the power supply — to do so would be really inefficient and, if you are not an electrician, difficult and dangerous to attempt.


REQUEST
RESPONSE
USER
WEB APPLICATON
API
DATABASE
WEB SERVER

APIs in client-side JavaScript

- **Browser APIs** are built into your web browser and are able to expose data from the browser and surrounding computer environment and do useful complex things with it.
- For example, the Web Audio API provides JavaScript constructs for manipulating audio in the browser — taking an audio track, altering its volume, applying effects to it, etc. In the background, the browser is actually using some complex lower-level code (e.g. C++ or Rust) to do the actual audio processing. But again, this complexity is abstracted away from you by the API.

- **Third-party APIs** are not built into the browser by default, and you generally have to retrieve their code and information from somewhere on the Web.
- For example, the Twitter API allows you to do things like displaying your latest tweets on your website.
- It provides a special set of constructs you can use to query the Twitter service and return specific information.



Relationship between JavaScript, APIs, and other JavaScript tools

- client-side JavaScript APIs are, and how they relate to the JavaScript language
- **JavaScript** — A high-level scripting language built into browsers that allows you to implement functionality on web pages/apps. Note that JavaScript is also available in other programming environments, such as Node.
- **Browser APIs** — constructs built into the browser that sits on top of the JavaScript language and allows you to implement functionality more easily.
- **Third-party APIs** — constructs built into third-party platforms (e.g. Twitter, Facebook) that allow you to use some of those platform's functionality in your own web pages (for example, display your latest Tweets on your web page).

- **JavaScript libraries** — Usually one or more JavaScript files containing custom functions that you can attach to your web page to speed up or enable writing common functionality. Examples include jQuery, Mootools and React.
- **JavaScript frameworks** — The next step up from libraries, JavaScript frameworks (e.g. Angular and Ember) tend to be packages of HTML, CSS, JavaScript, and other technologies that you install and then use to write an entire web application from scratch. The key difference between a library and a framework is "Inversion of Control". When calling a method from a library, the developer is in control. With a framework, the control is inverted: the framework calls the developer's code.

common categories of browser API

- APIs for manipulating documents loaded into the browser.
- APIs that fetch data from the server to update small sections of a webpage on their own are very commonly used
- APIs for drawing and manipulating graphics are widely supported in browsers
- Audio and Video APIs like HTMLMediaElement, the Web Audio API, and WebRTC
- Device APIs enable you to interact with device hardware: for example, accessing the device GPS to find the user's position using the Geolocation API.
- Client-side storage APIs enable you to store data on the client-side

Common third-party APIs

- The Twitter API, which allows you to do things like displaying your latest tweets on your website
- Map APIs, like Mapquest and the Google Maps API, which allow you to do all sorts of things with maps on your web pages.

- The Facebook suite of APIs, which enables you to use various parts of the Facebook ecosystem to benefit your app, such as by providing app login using Facebook login, accepting in-app payments, rolling out targeted ad campaigns, etc.
- The YouTube API, which allows you to embed YouTube videos on your site, search YouTube, build playlists, and more.

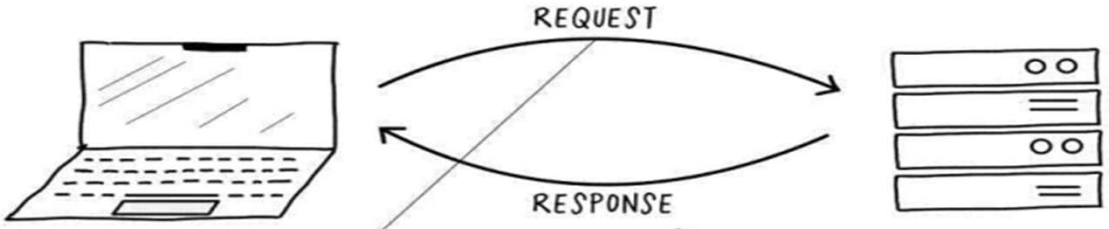
Benefits Of Using API

- APIs needed to make the connection between application functions to handle the data from predefined processes
- Easy of Integration: APIs can be embedded with any type of software application, so it can be easily integrate with the applications and functions between different websites

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

THESE METHODS ARE:
GET, POST, PUT, PATCH, DELETE, HEAD, TRACE, OPTIONS, CONNECT

HTTP REQUEST CONTAINS

- REQUEST METHOD
- HTTP HEADERS
- BODY

SERVER RETURNS A RESPONSE WITH AN HTTP STATUS CODE

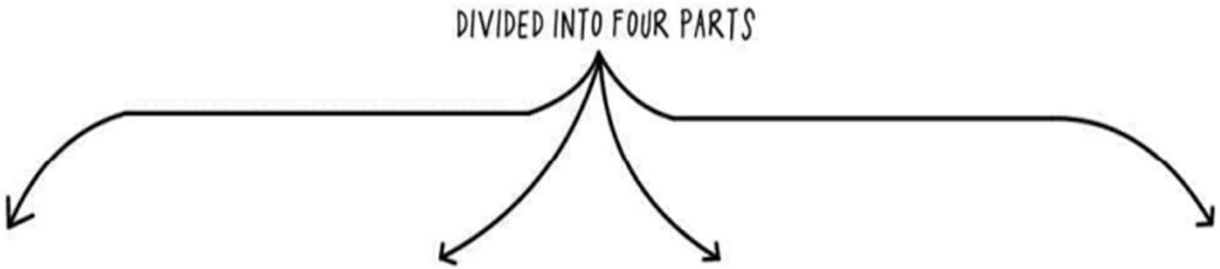
POPULAR HTTP STATUS CODE:
EX, 200, 202, 403, 404, 500 ETC

HTTP RESPONSE CONTAINS

- STATUS CODE
- HTTP HEADERS
- RESPONSE BODY

HTTP HEADERS

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



REQUEST HEADERS

- CLIENT TO SERVER

RESPONSE HEADERS

- SERVER TO CLIENT

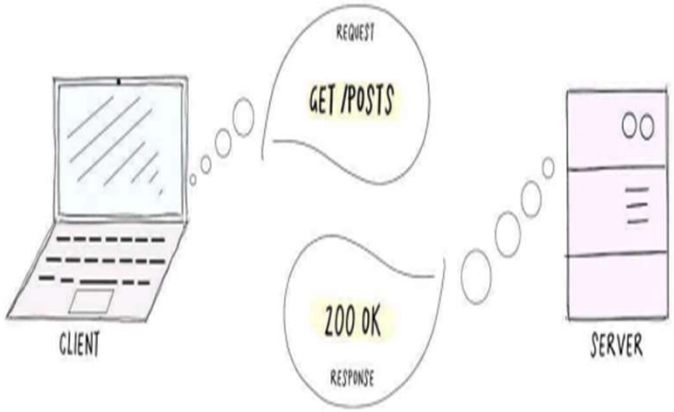
REPRESENTATION HEADERS

- INFORMATION ABOUT THE BODY OF THE RESOURCE

PAYLOAD HEADERS

- INFORMATION ABOUT THE PAYLOAD DATA

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

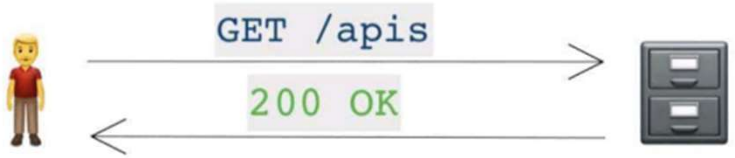
404 NOT FOUND
INVALID URL

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.

POST

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.

PUT

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

PATCH

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.

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.

REST API and RESTful API

- Difference between REST API and RESTful API
- Find a Api