**Node advance 2**

Streams:  
streams-->they are kind of array of data

streams are used if we have huge amount of data to process in code

if a file is requested then we can fulfill it using streams-->we should read the file as a stream and write the file as a stream

in this way our application will not consume size of file in memory

one chunk at a time is passed through stream

node http pkg is stream ready

we can create our own stream using the stream package

zip,crypto function uses streams

express is a flexible nodejs web app framework

It is used to make web apps faster and easier

**res.end:** comes from NodeJS core. In Express JS if you need to end request in a quick way and do not need to send any data then you can use this function.

**res.send:** Sends data and end the request.

**res.json:** Sends data in JSON format and ends the request.

what can server do?

routing,handling rquest,handling responses

we can write coding for server using http package without using express

express code too uses http behind the scene

express is a pkg available in npm

Template engine

it is used for using javascript inside html and it is also used for calculation

html just prints 2+3 but ejs can give result as 5

throw error searches for error handler in our code and error handler will show the error

and runs the next automatically

If no error-handling middleware is defined or if none of the error-handling middleware functions respond, Express sends a default error response to the client.

writing the code below of error handler after end of all routes

Defining error handler(optional)

function errorHandler (err, req, res, next) {

    if (res.headersSent) {

      return next(err)

    }

    // If headers haven't been sent, set status code and render error template

    res.status(500)

    res.render('error', { error: err })

  }

Creating and using express server

A computer screen with text

Description automatically generated

Get,post,put ,delete request:

A screen shot of a computer program

Description automatically generated

Eventemitter in js:

Event emitters are objects that can emit named events and allow functions (listeners) to be attached to these events.

The **server** object is an event emitter, and one of the events it emits is the **request** event. This event is emitted each time the server receives an HTTP request.

import EventEmitter from events module:

import not compatible with latest version

used require and for this command to work first i did npm i events

and then in package.json i added "type": "module" under dependencies

if we emit event before listening then it will not print anything on screen as here they work synchronously

first we have to give what to do when a event is emitted  using the on keyword and after that we use emit keyword

listen to event-->reacting to change something

if we start listening after it has been emitted-->this will not work at all

as it works synchronously

so if we have multiple events going on,we wil use eventemitter

Synchronous events:

const EventEmitter = require('events');

// Create an instance of EventEmitter

const myEmitter = new EventEmitter();

// Register two listeners for the 'myEvent' event

myEmitter.on('myEvent', () => {

console.log('Listener 1');

});

myEmitter.on('myEvent', () => {

console.log('Listener 2');

});

// Emit the 'myEvent' event

myEmitter.emit('myEvent');

console.log('Event emitted');

* We create an instance of **EventEmitter** named **myEmitter**.
* We attach two listeners to the event **'myEvent'**.
* When we emit the **'myEvent'** event using **myEmitter.emit('myEvent')**, both listeners are called synchronously, one after the other.
* Only after both listeners complete their execution, the program moves on to execute the next statement (**console.log('Event emitted')**).

However, it's important to note that while the execution of event listeners is synchronous, the event loop itself in Node.js is asynchronous.