Express Framework.
with Node js

04/10/2023

Express Framework:-

⇒ A framework on top of Node.

⇒ Node js is a Runtime framework.

⇒ Express - framework allows to make backend to our website.

⇒ Lesscode, Middleware. +.....

An express server:-

client - side -> front-end

server - side -> back-end.

creating a Express server.

-> Create directory

-> create index.js file

-> Initialize npm. [npm install express]

-> Install the Express package.

-> Write server application in index.js.

-> Start server. [node index.js]

refer document:- Express.js.com

code:-

```
import express from "express";
const app = express();
                  -> Port
app.listen(3000, () => {
    console.log("server running on port 3000");
});                                 ↑
                                call back.
```

What is Local host ?

→ Making our own computer as server when there is no server

→ we can give many ports on the Local host

we can check which port on our computer is currently listening for interactions from outside by typing

```
netstat -ano | findstr "LISTENING"" → windows
sudo lsof -i -P -n | grep LISTEN" → Mac/Linux
```

HTTP request :-

Learn the language to make request to your server

→ HTTP → Hyper Text Transfer Protocol - used to Communicate b/w Computers via internet.

Request vocab :-

① GET → Request a resource from server. [index.html]

② POST → sending a resource [email sign in etc...,]

③ PUT
④ PATCH

both stands for update
PUT → replace resource
Patch → Patch up a resource.

⑤ DELETE → Delete resource.

code :-

```
import express from "express";
const app = express:
app.get("/",(req, res) => {
    res.send("Hello world");
});
app.listen(3000,() => {
    console.log("....");
});
```

install nodemon
for auto refresh
npm i -g nodemon

Making Requests:-

-> Postman and making other HTTP requests.

-> HTTP is to communicate to servers.

HTTP Response Status Code:-

↳ indicates whether a specific HTTP request has been successfully completed. Responses are grouped in 5 classes.

1. informational response (100 - 199)
2. Successful responses (200 - 299)
3. Redirection responses (300 - 399)
4. client - error responses (400 - 499)
5. Server - error responses (500 - 599)

www.postman.com/downloads/

Express Middleware:-

-> request between the server and the Route [Get, PUT, POST...] is called middleware.

-> Middleware - Preprocess the request, authentication

body-parser. used to parse the body

Eg:- forms.html -> Node.js.

sending a page as response : refer course resource.

some addons

```
import {dirname} from "path";
import {fileURLToPath} from "url";
const __dirname = dirname(fileURLToPath(import.meta.url));

app.get("/", (req, res) => {
    res.sendFile(__dirname + "<path>");
});
```

To post something:-

```
import bodyParser from "body-parser";

~~app.post~~
app.use(bodyParser.urlencoded({extended : true}));
app.post("/submit", (req, res)=> {
    console.log (req.body);
});
app.listen(port.........);
```

ⓧ morgan => logging:-

```
 import morgan from "morgan";
 app.use(morgan("tiny"));
```

Building own middleware.

```
app.use((req, res, next)=> {
   console.log("Request method:", req.method);
   next();
});
```

Note :-

use "req.body["<for="?" in label"]"
to get the value.