node js codevolution

<https://www.youtube.com/watch?v=LAUi8pPlcUM&list=PLC3y8-rFHvwh8shCMHFA5kWxD9PaPwxaY>

<https://www.youtube.com/watch?v=CTxrpmY1At8&list=PLC3y8-rFHvwhco_O8PS1iS9xRrdVTvSIz>

<https://www.youtube.com/watch?v=vLvNr3Wa5YI&list=PLC3y8-rFHvwj200LLotCYum_9wmGeTJx9>

<https://www.youtube.com/watch?v=YYnf5vJRuUY&list=PLC3y8-rFHvwh11bWtwm3_qKvo46uDmaal>

<https://youtu.be/XQT6XiJt4DE?list=PLC3y8-rFHvwh8shCMHFA5kWxD9PaPwxaY>

github code step by step

[GitHub - anil-sidhu/node-js](https://github.com/anil-sidhu/node-js)

Node Js Tutorial – code step by step

<https://youtu.be/zaLfOjNEOaQ?list=PL8p2I9GklV456iofeMKReMTvWLr7Ki9At>

Node JS in Hindi #3 Install node , NPM

npm -v

node -v

node - enter

console.log(“hello world”);

jab bhi koi statement return nhi krti

undefined milta hai

kuch return nhi ho rha hai esliye undefined aata hai

var x = 10; var y = 20;

console.log(x + y );

console.warn(x+y);

make folder(node\_tut) : index.js : console.log(“hello world”);

Terminal cmd – node . \index.js

same javascript code we need to write for node js

node js me hum import, export use nhi kr skte : jaise hum javascript, react me use krte hai

You can use use like this !!!

Folder(node tut) : index.js(file)

App.js

files import export

app.js

module.exports = {

X : 10,

Y : 20,

Z :function(){

Return 10;

}

}

Index.js

const app = require(‘./app’);

Console.log(app);

Console.log(app.x);

Console.log(app.y);

Console.log(app.z());

Core Modules

module - by default feature har ek programming language ke hote hai

two types of module

Core Module 1.) Global 2.) Non Global

External module

Core Modules : har ek programming language ke undar ek basic feature phle se hi hote hai jaise ..this is called core module…data base se connect krna, files vagear create krne ke liye, code ko process krne ke liye, api call krne ke liye or.. core module.. kyuki ye basic se hi hamare programming language me bane hote hai.

Index.js - core module - console.log(“hello world”);

File system, buffer, http, api calls, database connections, these are core module.

Global, Non Global Module

Global aise module hote hai jinko import krne ki jarurat nhi…

Global and non Global

Glaobal Module

Global aise module jinhe import krne ki jarurat nhi hoti.

console.log(“hello world”); //Global module

Non Global Module

jo module hume import krne pdte hai non global module.

Index.js

const fs = require(‘fs’);

console.log(“code step by step”);

fs.writeFileSync(“hello.txt” , “hello world”); //write this code to create files.

Output :

Hamari file create ho jayegi : hello.txt

jo module hume import krne hote hai : non global module hote hai

jo import nhi krne padte wo hamare : Global module hote hai : console.log

To check directories

const fs = require(‘fs’);

console.log(“directories name”, \_\_dirname);

node .\index.js

C:\Users\HP\Desktop\node-tut

To check filename

const fs = require(‘fs’);

console.log(“directories name”, \_\_filename);

C:\Users\HP\Desktop\node-tut\index.js

or

const fs = require(‘fs’).writeFileSync;

fs(“abc.txt”, abc);

Create Server : make basic server output on browser

Make basic server output on browser

function as parameter in node

arrow function

get output on browser

index.js

const http = require(‘http’);

http.createServer((req, resp) => {

resp.write(“hello this is anil sidhu”);

resp.end();

}).listen(4500);

Output : node .\index.js

Hello this is anil sidhu

we can also write like this

resp.write(“<h1>hello this is anil sidhu</h1>”);

const http = require(‘http’);

http module : node js ke undar server ki request or response ko handle krta hai.

http.CreateServer() : pure ke pure function ko as a parameter lega.

http.createServer().listen(4500);

const http = require(‘http’);

http.createServer( (req, res) => {

}).listen(4500);

req , request kya karega lets see ki hmne form vagera submit kra hai 🡪 or urls se koi parameter vagera bheje hai : unko node js ke undar access krte hai

res – yha se jo send krna hota hai , wo hota hai response…yha se client ko kuch na kuch response de rhe hai

Index.js

const http = require(‘ http’);

http.createServer((req, res) => {

resp.write(“hello this is anil sidhu”);

resp.end();

}).listen(4500);

or we can also write like this : index.js

const http = require(‘ http’);

Function dataControl(req, res){

Res.write(“hello this is anil sidhu”);

Resp.end();

}

http.createServer(dataControl).listen(4500);

http : req or res ko handle krtsa hai

localhost:4500

create server

All About package.json

cmd : npm init

eske undar package manage hote hai : browser(npm colorful package) : npm I colors : folder create ho jayega node module ke name se

package.json : sare dependencies rhte hai

node module

package.json

package.lock.json

node js : Single threaded : ek time me ek hi command run krti hai

make basic server output on browser

make basic server

function as a parameterer in node

arrow function

get output on browser

index.js

const http = require(‘ http‘);

http.createServer() : yahi ek function hota hai jo ki pure function ko hi as a parameter lega.

htttp.createServer().listen(4500);

Test(a,b){

}

Test(10, 20);

Parameter number ki jagah function bhi pass kr skte hai.

Ye jo createServer() function hai, ye pura ka pure ek function as a parameter leta hai.

Bahut se function hai jo . function as a parameter leta hai.

const http = require(‘http’);

http.createServer((req, resp) =>{

resp.write(“ hello this is anil sidhu”)

resp.end();

}).listen(4500);

req

req krega humne form vagera submit kri : urls se parameter vagera bhejhte hai : Unko node js ke undar aceess krta hai : yha se jo send krna jo hota : wo hota hai response : yha se client ko kuch na kuch response de rhe hai

yha se directly response send krenge.  
abhi ke liye, yha se kuch send krna hota hai

wo hota hai response: yha se client ko kuch na kuch response de rhe hai : response send kr rhe : resp.write(“hello this is anil sidhu”);

response ko end bhi krna hota hai. otherwise node js ko pta nhi chlega mujhe stop kha pe krna hai.

resp.end();

simplify code

function dataControl(req, resp){

resp.write(“hello this is anil sidhu”);

resp.end()

}

http.createServer(dataControl).listen(4500);

convert. in arrow function

const http = require(‘http’);

const dataControl = (req, resp) =>{

resp.write(“hello this is anil sidhu”);

resp.end();

}

http.createServer(dataControl).listen(4500);

All about package.json

What is package file

Install external package

Package.json : ye hamre project ki detail rkhti hai : coding se related : jaise hamare projects ka version kya hai: uska name kya hai: uski git repository kya hai : uske undar koun koun si command use ho rhi hai : uske undar koun koun se package dal rhe hai : jab ek project bnate hai to usme bahut sare projects bna rkhe hai : validation ke liye hmne dal liya : kuch console logger wagera dal liya data base connection ke liye : mango db :

mysql : external package wo sare ke sare kaise manage hote hai : wo hote hai : package.json me manage hote hai

Kaise project build bnati hai

Kaise project run krni hai

Command run krni hai

Ye sare package.json me hoti hai

Package.json hamare projects ki details rakhti hai

coding se related details rkhti hai

Index.js

console.log(“package.json”);

npm init : enter : package.json file ban jayegi :

package.lock.json : package ki detail rkhera

node js single threaded hai ya multi threaded

node js single threaded hai : ek time me ek hi command run krti hai

multi threaded me ke sath : 4,5 command ko ek sath run kr skti hai

small challage for u

what happens if the node\_module folder is deleted

correct way to push data on git

index.js

const colors = require(‘colors’);

console.log(‘package’, Red);

output - node .\index.js

Package(red color me hoga);

node js chalk package - install - import use

node module galti se delete ho gya : npm install

node\_module ke bahar ek file bnani hai .gitignore github me push krne ke liye .gitignore

.gitignore file - /node\_modules

nodemon time saving module

es package ka use krke bahut sara time save kr skte hai : sare project me time save krne ke liye use kiya jata hai :

index.js

console.warn(“try nodemon with node js”);

command run krne ke bad ye band ho gyi

ab kuch bhi change kiya code me : ab humko esko phir se run krna pdega

har bar jitni code me change krenge : agar yha par node command use kr rhe hai : again and again run : esse hamara time waste hota hai

abhi to thik ek line ka code likha agar api bna rhe : usko test kr rhe : us time me complex api : database se connectivity ho chuki hai: bar bar aisa check krenge to bahut time hai ­: esliye time save krne ke liye nodemon use krte hai : nodemon node js ke project ko continuesly run krta hai : bar bar node js ko run krne ki jarurat nhi

installation nodemon : nodemon documentation : cmd : npm I nodemon -g

to run : folder directories: : nodemon .index.js

ab code me kuch bhi change krenge to apne app run hoga : time saving

interview question : node js : async language : node js single threaded

sync : lets see 3 4 syntax run kr di : to wo sync me chlenge : 1st, 2nd, 3rd, 4th

async : if second syntax is taking time for fun : will execute next one : then 2nd one will execute wait nhi kr skti kisi bhi script ka

make a simple api : simple and basic api

make a server

create header and api body

create an api with static data

put data in api file

index.js

const http= require('http');

const data = require('./data');

http.createServer((req,resp)=>{

resp.writeHead(500,{'Content-Type':'application\json'});

resp.write(JSON.stringify(data));

resp.end();

}).listen(5000);

Data.js

const data=[

    {name:'anil',mail:'anil@test.com'},

    {name:'sam',mail:'peter@test.com'},

    {name:'peter',mail:'peter@test.com'}

]

module.exports=data;

Output : nodemon ./index.js

<https://github.com/anil-sidhu/node-js>

index.js

whi hamari api bnane me help krta hai : req, resp ko yhin wala package handle krta hai

const http = require(‘http’) ;

http.createServer((req, resp) => {

}).listen(6000);

run : nodemon .\index.js

response send krenge uske liye resp.writeHead() : data send kr rhe hai

resp.writeHead(200, {‘Content-Type ’ : ‘application/json’});

Body ho jayegi : resp.write(JSON.stringyfy({‘name ’ : ‘anil sidhu’}, ‘email’ : ‘anil@tcs.com’}));

resp.end()

request leni hai user se , react,angular,vue se tab use krte hai : request

index.js

const http = require(‘http’);

http.createServer((req, resp) =>{

resp.writeHead(200,{‘Content-Type’ : ‘application\json’})

resp.write(JSON.strigyfy({name : “anil sidhu”, ‘email’ : ‘anilsidhu@tcs.comm’})));

resp.end();

} ).listen(5000);

Output : {‘name’ : ‘anil sidhu’, ‘email’ : ‘anilsidhu@tcs.com’}

api me postman attach postman

Data ko alag se file me rakhu : or kuch or data add kru : hotchpauch

Data.js

const data = {‘name’ : ‘Monija’, email : ‘monika@gamil.com’};

index.js

const http = require(‘http’);

const data = require(‘ ./data’);

http.createServer((req, resp) =>{

resp.writeHead(200, {‘Content-Type’ : ‘application/json’});

resp.write(JSON.stringyfy(data));

resp.end();

}).listen(4000);

server node js ke Sath me

header and api ki body bnanni hai

create an api with static data

put data in another file.

data dusri file me put kr denge. So alag file me dikhe.

or api ka code alag file me dikhe

create server

response ka header add krna hai

first parameter(response ka code)

second parameter : jis type ka hum data send krenge : content type /json

Uske bad body likhni hai wo likh skte hai

then response ko end krna hai

data.js

const data = {

‘name’ : ‘anil sidhu’,

‘email’ : ‘anil@tcs.com’

};

module exports = data;

lets see hume kuch or user add krne hai : multiple user to object ko array ke undar lena pdta hai

const data = [

{name : ‘anil’, email : ‘anilsidhu@tcs.com’, contact : ‘999999’},

{name : ‘anil’, email : ‘anilsidhu@tcs.com’, contact : ‘1234’},

{name : ‘anil’, email : ‘anilsidhu@tcs.com’, contact : ‘1234’}

]

200 : ok

201: created

404: page data nhi milta

500 : server me hamara error hai

Getting input from command line

set input from command line

create file with input

delete the file with input

index.js

console.log(“code step by step”);

run - node index.js

code step by step

ab maine node js ke sath me kuch extra parameter pass kr diya

node index.js 100 anil

ye jo input hai usko node js ke undar use kr paye

abhi kuch nhi hoga normal output hai whi chlegi 100 anil jo hai - esi ko hum get krenge

index.js

console.log(process);

process ek object hota hai jiske undar almost sab kuch hota hai node js ka : run krne me kitna sari chige print ho jayega.

console.log(process.argv)

ye yha pe 2 properties de di

two input

first node js ki hamari file save hai wo

'C:\\Program Files\\nodejs\\node.exe',

second - jaha se hum esko run kr rhe hai wo

'C:\\Users\\kamlesh\\Desktop\\4. node js\\3. node js practice\\index'

Terminal - node index.js hello

ab 3 input aa gya

'C:\\Program Files\\nodejs\\node.exe',

'C:\\Users\\kamlesh\\Desktop\\4. node js\\3. node js practice\\index'

Hello

Terminal - node index.js hello hi

ab 4 input aa gya

'C:\\Program Files\\nodejs\\node.exe',

'C:\\Users\\kamlesh\\Desktop\\4. node js\\3. node js practice\\index'

Hello

Hi

Ab ek ek krke kaise get kr skte hai

Console.log(process.argv[2]);

Output

Hello

Console.log(process.argv[3]);

Hi

first-properties, second properties, hello

Console.log(argument.argv[2]) - as an array

Scenario - add/remove - name – text

index.js

to create file - we need file system module- jo ki node js ke undar inbuild hota hai

const fs = require(‘fs’);

const input = process.argv;

fs.writeFileSync(input[2], input[3]);

1st : input[2] : file ka name

1st : input[3] : data jo kuch hum enter krna chahte hai

Terminal : node index.js apple.txt ‘this is a fruits’

Output : jaise hi enter kiya apple.txt file create ho gyi

and text : this is a fruit

const fs = require(‘fs’);

const input = process.argv;

fs.writeFileSync(input[2], input[3]);

scenorio : ab esko hum jahte hai : add bhi kr pau or remove bhi kr pau

const fs = require(‘fs’);

const input = process.argv;

fs.writeFileSync(input[2], input[3]);

If(input[2] = = ‘add’){

fs.writeFileSync(input[3], input[4])

} else if (input[2] == ‘remove’) {

fs.unlinkSync( input[3])

}

else{

console.log(“invalid output”);

}

run : node index.js add orange.txt “this is color and fruit”

orange.txt - name se file ban gyi - or text - this is color and fruit

lets see ab file ko remove krna chahte

hai

run : node index.js remove orange.txt - hamara orange.txt file remove ho jayega

Node JS in Hindi # 12 Getting input from command line

Display file list from folder

create file and display those file in console

show file list

make file in a folder

use path module

get filename and print

node-tut (folder)

node-module,

index.js,

package.json,

package.lock.json,

files

index.js ke sath files name ka folder create krte hai

Scenario - files : command line ke through create krte hai : file create krna : or read krna bhi sikh jaye

Index.js

const fs = require(‘fs’);

fs : file system : ye file system module hai : node js ka core : esko install krne ki jarurat nhi hoti : node js by default deta hai esko

const fs = require(‘fs’);

fs.writeFileSync(‘apple.txt’, ‘this is apple file’);

node .\index.js

apple.txt

file create ho gyi

but ye files : files folder ke undar create nhi hue

Loop me file create

ek sath 7, 8 file create krna chahte hai

Esko file module ke sath - path module hota hai uske sath use krenge or loop ke undar create krenge

Index.js

const fs = require(‘fs’);

const path = require(‘path’);

Now path get krte hai

const dirPath = path.join(\_\_dirname); - current directory ka path

console.warn(dirpath);

// C:\Users\kamlesh\Desktop\4. node js\3. node js practice

run : node .\index.js

output :   
directories ka path mil gya  
  
const dirPath = path.join(\_\_dirname, ‘files’);

but hume files ka path chahiye

ab hume files ke name se output mil gyi

path aa gya

ab hume files ko create krni hai

const fs = require(‘fs’);

const path = require(‘path’);

const dirPath = path.join(\_\_dirname, ‘files’);

C:\Users\kamlesh\Desktop\2. node js\2.Practice\files

path hamare pass aa gya  
ab hume files ko create krni hai

files ke undar 5 files create krni hai

const fs = require(‘fs’);

const path = require(‘fs’);

const dirPath = path.join(\_\_dirname, ‘files’);

for(i = 0; i<5; i+

+){

fs.writeFileSync(dirPath + ‘ /hello’ + i + ‘.txt’ , ‘a simple test file’);

}

node .\index.js

files

hello0.txt

hello1.txt

hello2.txt

hello3.txt

hello4.txt

ab es file ko read krke list krwani hai

file read krne me jab kam aati hai , ek file ke undar ka content chahiye hota hai

jab hume list dikhani rhti hai , to directories read krni rhti hai,

hum log files wali directories ko read krenge

taki eske undar jitni bhi content files (txt files ) dikh jaye

const fs = require(‘fs’);

const path = require(‘path’);

const dirPath = path.join(\_\_dirname, ‘files’);

fs.readdir(dirPath, (err, files) => {

console.log(files);

})

node .\index.js

output

hello0.txt

hello1.txt

hello2.txt

hello3.txt

hello4.txt

as an array mili : single single files chahiye to

const fs = require(‘fs’);

const path = require(‘path’);

const dirPath = path.join(\_\_dirname, ‘files’);

fs.readdir(dirPath, (err, files) => {

files.forEach((item) => {

console.log(item);

})

})

node .\index.js

hello0.txt

hello1.txt

hello2.txt

hello3.txt

hello4.txt

next lecture read : ek files ke undar ka data kis tarike se read kr skte hai

next video me data ko web page : api : ke undar bhi lekar jayenge

14 Asynchronous Programming Language

crud with file system

make file

read file

update file

rename file

delete file

const fs = require(“fs”);

fs.writeFileSynch(“apple.txt”, “this is simple text file”);

//ese run krengge to root me file create hogi

Particular files crud me create krni hai

const fs = require(“fs”);

const path = require(“path”);

const dirpath = path.join(\_\_dirname, “crud”);

const filepath = `${dirpath}/apple.txt`; //here sign is not a single quotes

fs.writeFileSynch(filepath, “this is simple text files”);

output

files – apple.txt (create ho chuki hai)

this is simple text files

ab es files ko read krni hai

const path = require(“path”);

const dirpath = path.join(\_\_dirname, “crud”);

const filepath = `${dirpath}/apple.txt`;

// that means comment

//fs.writeFileSynch(filepath, “this is simple text files”);

fs.readFile(filePath, (err, item) =>{

console.log(item);

});

<Buffer 63 72 65 61 74 69 6e 67 20 61 20 62 6e 61 6e 61 20 66 69 6c 65 73 73 73>

Buffer ????

fs.readFile(filePath, ‘’utf’, (err, item) =>{

console.log(item);

})

Output

This is a simple text file

Files update

const path = require(“path”);

const dirname = path.join(\_\_dirname, “files”);

const filepath = `${dirpath}/apple.txt `;

fs.appendFile(filePath, “and file name is apple.txt”, (err) =>{

if(!err) console.log(“file name is updated”);

});

Rename Files

const fs = require(“fs”);

const path = require(“path”);

Const dirname = path.join(\_\_dirname, “files”);

Const filepath = `${dirname}/apple.txt`;

fs.rename(filePath, `${filepath}/fruit.txt`, (err) =>{

if(!err) console.log()

});

Delete file

fs.unlinkSync(`${filePath}/fruit.txt`);

const fs = require(“fs”);

|  |
| --- |
| const path = require('path'); |
| const dirPath= path.join(\_\_dirname,'crud'); |
| const filePath = `${dirPath}/apple.txt`; |
| // fs.writeFileSync(filePath,'this is a simple file'); |
| // fs.readFile(filePath,'utf8',(err,item)=>{ |
| // console.log(item); |
| // }) |
| // fs.appendFile(filePath,' for fruits',(err)=>{ |
| // if(!err) console.log("file is updated") |
| // }) |
| // fs.rename(filePath, `${dirPath}/fruit.txt`,(err)=>{ |
| // if(!err) console.log("file name is updated") |
| // }) |
|  |

|  |
| --- |
| fs.unlinkSync(`${dirPath}/fruit.txt`);  🡪Buffer  Ek temporary memory location , jab bhi hum file create krte hai , koi bhi operation perform krte hai file system ke uppar , node js ko thodi si memory chahiye rhti hai,  Jaise ki RAM chahiye rhta hai humare system ko koi bhi operation perform krne ke liye  Waise hi node ko thodi si memory , taki wo operation ko perform kr sake.  Buffer |

 15 Asynchronous Programming Language

synchronuous : synchronuous operation tasks are performed one at a time.

asynchronuous : in asynchronuous, second task do not wait to finish first .

example

console.log(“start execution…… ”);

setTimeOut(() =>{

console.log(“logic exe….”);

}, 2000);

console.log(“complete execution…… ”);

output : start execution……

complete execution……

logic exe….

Example :

let a = 10;

let b = 0;

setTimeOut(() =>{

b = 20;

}, 2000);

console.log(a+b);

Output 10: drawback

16.  Handle Asynchronous Data

Example :

let a = 10;

let b = 0;

setTimeOut(() =>{

b = 20;

}, 2000);

console.log(a+b);

Output 10: drawback

To handle asynchronous operation use promises

Using promises

Let a = 10;

Let b= 0;

Let waitingData = new Promise((resolve, reject) => {

setTimeOut(() => {

resolve(30);

}, 2000)

})

waitingData.then((data) =>{

b = data;

console.log(a+b)

});

Output –> 50

let a = 10;

let b = 20;

let waitingData = new Promise((resolve, reject) =>

{

setTimeout(() =>{

    resolve(30);

})

})

waitingData.then((datab) =>{

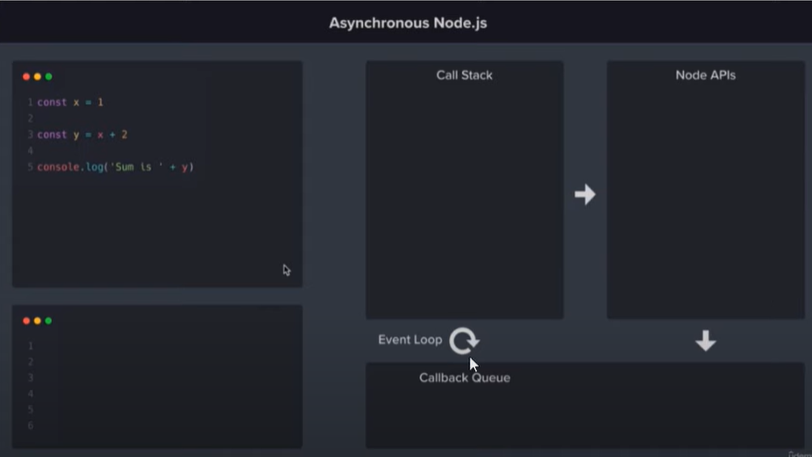
    b = datab;

    console.log(a+b);

})

17.   how node js works : architechture: callstack

🡪callstack 🡪 node api 🡪 callback queue 🡪



<https://youtu.be/mGVFltBxLKU?list=PL8p2I9GklV456iofeMKReMTvWLr7Ki9At>

example :

console.log(“starting up ”);

setTimeOut(() =>{

console.log(“2 second log”);

}, 2000) ;

setTimeOut(() =>{

console.log(“0 second log”);

}, 0) ;

console.log(“finishing up”);

Output : starting up

finishing up

0 second log

2 second log

Important Important Important Important

18. Express js Introduction

Important Important Important Important

Api

Route

Middle ware

Request handling : get, post, put, delete

Web pages

Css integrate

https://expressjs.com/

<https://www.npmjs.com/package/express>

npm i express

what is express js

install

make example with express js

index.js

cmd command : npm install express : ek package package.json ke undar install ho jayega : package.lock.json

index.js

//abhi about us page, contact us page , help, home page

const express = require(‘express’);

const app = express();

app.get(‘ ’, (req,res) => {

res.send(‘hello this is home page’);

})

app.get(‘ /about’, (req,res) => {

res.send(‘hello this is about page’);

})

App.listen(5000);

node .\index.js

Output: localhost:5000

Hello, this is home page

localhost:5000/about

Hello, this is about page

//get method routes provide krte hai : ye route get bhi ho skte hai: post bhi ho skte hai ye

//req : aa kya rha hai : ye kab use hota hai

//res jo hum send krenge yha se

Refer github pages

const express = require('express');

const app = express();

app.get("", (req, resp) => {

console.log(req.query.name)

resp.send("Welcome,"+req.query.name);

});

app.get("/about", (req, resp) => {

resp.send("Welcome, This is a About Page");

});

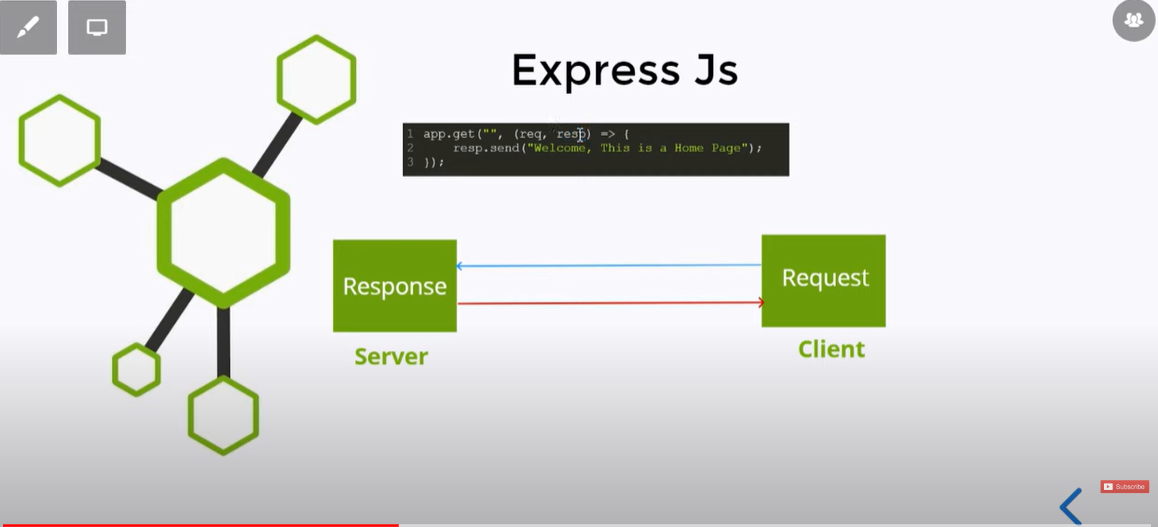
app.get("/help", (req, resp) => {

resp.send("Welcome, This is a Help Page");

});

app.listen(5000);

19.  19 Routing Params - Request and Response



client hamesha request krta hai

server hamesha response deta hai

node js hamara ek server hai

usne response diya

es example me hmne koi request kri nhi ,

parameter aa rha but ,

req parameter humne use nhi kiya

no need to use both parameter ,

kuch specific data hmare pass nhi aaya client side se ,

agar hume kuch data send krna hota ,

name, id kuch send kr rhe hote – to eska req ,

req parameter ka use kr rha hota ,

abhi req ke undar kpoi data nhi aaya hai ,

to esliye server by default samajta hai ,

ha mere pass bas ek request aa gyi ,

or jo bhi function likha hai usko as a response throw kr do ,

esme hmne kuch bhi deta send nhi kra ,

browser ka url dala ,

uske according yha par deta de diya ,

home ke liye home,

about ke liye about wala page ,

ye jo req or res hai ,

server or client ko interact kran eke liye basically use me aate hai

Jab bhi client se deta aayega wo req wale parameter uske undar aayega or

Jo bhi deta send krenge wo res ke undar

Example

es tarah se deta ko send kr rhe

Localhost:5000?name = anil

esko get krke dikao

ye req ko get krke dikao

const express = require(‘express’);

const app = express();

app.get(“ “, (req, res ) => {

console.log(“ data send by browser”, req.query);

//output: localhost : 5000?name= monika

//req.query//output

//data send by browser {name : anil}

//req.query.name

//output : data send by browser anil

//

res.send(“welcome this is jome page” + req.query.name );

});

app.get(“ /about“, (req, res ) => {

console.log(“ data send by browser”, req.query);//output

res.send(“welcome this is about page”);

}) ;

app.listen(5000);

const express = require('express');

const app = express();

app.get("", (req, resp) => {

console.log(req.query.name)

resp.send("Welcome,"+req.query.name); //monika

});

app.get("/about", (req, resp) => {

resp.send("Welcome, This is a About Page");

});

app.get("/help", (req, resp) => {

resp.send("Welcome, This is a Help Page");

});

app.listen(5000);

20. Render HTML and JSON

Render html and json

How html tags

Show json data

Link pages

HTML and JSON data display

Json data api ke liye important

Html data website ke liye important

Ek page se dusre page kis tarah se link kr skte hai

html data ko render

const express = require(‘express’);

const app = express();

app.get(“ ”, (req, resp) => {

resp.send(<h1> welcome to home page</h1> );

} )

app.get(‘ /about’, (req, res) => {

resp.send(`

< input type = “text” placeholder = “user name”/>

<button> click me </button>

`);

})

app.listen(5000);

json data ko render krna hai

example :

app.get(“/help”, (req, res) => {

resp.send(

{

name : “anil”,

email :”anil@gmail.com”

}

)

} )

Output

{“name” : “anil”, “email” : “anil@tcs.com”}

Ek se jyada user ka data

app.get(“/help”, (req, res) => {

resp.send(

[

{

name : “anil”,

email :”anil@gmail.com”

},

{

name : “monika”,

email :”monika@gmail.com”

},

]

)

} )

Output :

[

{

“Name1” : “anil”,

“email” :”anil@gmail.com”

},

{

“Name” : “monika”,

“email” :”monika@gmail.com”

}

]

If we want to go one page to another page

Home to about us page

app.get(“ ”, (req, resp) => {

resp.send(

`

<h1>welcome to home page </h1>

<a href = “ /about“ /> go to about page</a>

`

)

})

app.get(‘/about’, (req, resp) =>{

resp.send(`

< input type = “text” placeholder = “user name”/>

<button> click me </button>

`);

app.listen(3000);

output

Localhost:5000/about

Input text box : click button

interview question

parameter se value kis tarike se get kroge

localhost:5000/about?name = anil

example

app.get(“/about”,(req, res) => {

resp.send(`

< input type = “text” placeholder = “user name” value = “${req.query.name}” />

<button>click me<button/>

<a href = “/ ”>go to home page <a/>

`)

})

localhost:5000/about?name = sam

output

input field me : sam value aayega

es tarah se query parameter se deta ko le skte hai

const app = express();

app.get("", (req, resp) => {

resp.send(`

<h1>Welcome to Home page</h1>

<a href="/about">Go to About page</a>

`);

});

app.get("/about", (req, resp) => {

resp.send(`

<input type="text" placeholder="user name" value="${req.query.name}" />

<button>Click</button>

<a href="/">Go to Home page</a>

`);

});

app.get("/help", (req, resp) => {

resp.send([

{

name:'Peter',

email:'peter@test.com'

},

{

name:'Bruce',

email:'bruce@test.com'

}

]);

});

app.listen(5000);

21. Make HTML page

html page ko kaise display kr skte hai

node js application me

how html tags

show JSON data

Link pages

make folder for html files and access it.

Make html files

Load html files

Interview questions

create folder and name it public

Public(folder)

about.html pages

about.html

<doctype html!>

<head>

<style>

</style>

</head>

<body>

<h1>this is about page </h1>

</body>

</html>

Index.js

const express = require(“express ”);

const path = require(“path”);

const app = express();

const publicpath = path.join(\_\_dirname, ‘public’); //points to a particular files

app.use(express.static(publicPath));

app.listen(5000);

output

localhost:5000/about.html

about page about page

about page about pages

about page about pages

home.html

<head>

<style>

</style>

</head>

<body>

<h1>this is home page</h1>

</body>

</html>

output

Localhost:5000/home.html

Home page

make root wala page

index.html page bna lete hai wo apne app hi access ho jata hai

localhost:5000

output : index wala page show hoga

path wala module kis kam me aata hai

ye hmare project ke folder ke access krne me help krta hai

humko koi bhi path chahiye : public : uske undar ka uske bahar ka path chahiye

sare ke sare path wale module ke help se leskte ho

static

static method what do

static pages ko load krta hai

static content ko load krta hai

Main folder

public (folder)

about.html,

help.html

Index.html

const express = require('express');

const path = require('path');

const app = express();

const publicPath=path.join(\_\_dirname,'public')

app.use(express.static(publicPath));

app.listen(5000);

Help.html

<!DOCTYPE html>

<html lang="en">

<head>

<title>Help Page</title>

<style>

h1{

color: red;

}

</style>

</head>

<body>

<h1>Help Page</h1>

</body>

</html>

App.use: it is a function .

Middleware

It is a function of express js

app.use(express.static), hamari jo static files hai , usko load krega, kha se load krega, public wale folder se

22. Remove extension from URL and 404 Page

urls se extension kaise remove kr skte hai

Localhost:5000/about.html

.html remove

urls or page dono me alag alag kis tarike se kr sk

te hai

galat urls krne se 404 page shows krane honge

Remove extension from urls

Apply get method

Remove an extension from urls

Make 404 pages

Apply 404 pages

Localhost:5000/home.html

.html remove krna hai urls

Urls or page dono me alag alag kaise kr skte hai

Make 404 page

galat urls ke liye specific pages

Index.js

const express = require(‘express’);

const path = require(“path”);

const app = express();

const publicPath = path.join(\_\_dirname, ‘public’);

app.get(“ ”, (\_\_, resp) => {

resp.sendFile(${publicPath}/index.html);

});

app.get(“/about ”, (\_\_, resp) => {

resp.sendFile(${publicPath}/about.html);

})

app.get(“ /help”, (\_\_, resp) => {

resp.sendFile(${publicPath}/help.html);

})

app.listen(5000);

localhost:5000

home page

localhost:5000/about

about page

localhost:5000/help

help page

agar koi bhi match nah o to ye wala dikh jaye

app.get(“ \* ”, (\_\_, resp) => {

resp.sendFile(${publicPath}/pagenotfound.html);

})

const express = require("express");

const app = express();

const path = require("path");

const filepath = path.join(\_\_dirname, "public");

app.get(" ", (\_\_,  resp) => {

    resp.sendFile(`${filePath}/index.html`);

    });

    app.get("/about", (\_\_, resp) =>{

        resp.sendFile(`${filepath}/about.html`);

    });

    app.get("/help", (\_\_, resp) =>{

        resp.sendFile(`${filepath}/help.html`);

    });

    app.get("\*", (\_\_, resp) =>{

        resp.sendFile(`${filepath}/notfound.html`);

    });

    app.listen(4500);

output

localhost:4500

404 – no page found

localhost:4500/about

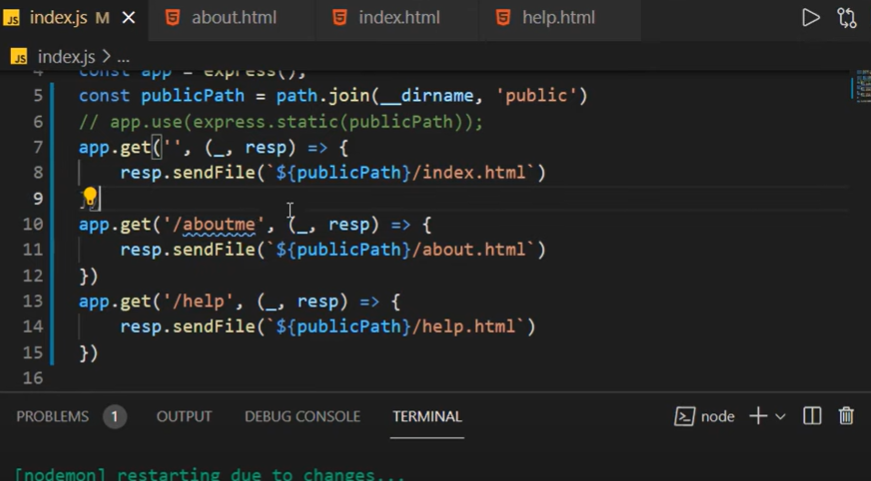
About page about page about page

localhost:4500/help

help page help page help page

localhost:4500/abcd

404 – no page found

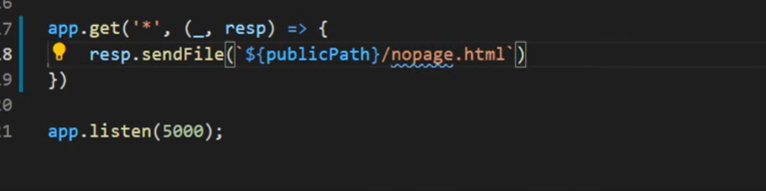


localhost:3000/about

about.html likhne ki jarurat nhi hai

About wala page show hoga

koi bhi galat urls rhega to



23. EJS Template Engine

template engine

what is template engine

install ejs template package

setup dynamic routing

make dynamic page

interview question

ejs dynamic pages bnane me kam aate hai

dynamic page : static page

static pages : simple html pages

dynamic pages

jab data database se udh ke aata hai

sari ki sari data base se udh ke aa rhi hai to wo ek dynamic page hota hai

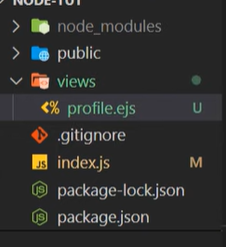
node package manager

<https://www.npmjs.com/package/ejs>

Folder structure

views pages

by default configuration



Views (folder)

profile.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Profile Page</title>

</head>

<body>

<h1>Welcome, <%= user.name %></h1>

<h3><%= user.email %></h3>

<h3><%= user.country %></h3>

</body>

</html>

Index.js

const express = require('express');

const path = require('path');

const app = express();

const publicPath=path.join(\_\_dirname,'public')

app.set('view engine','ejs');

app.get('' “ ,(\_,resp)=>{

resp.sendFile(`${publicPath}/index.html`)

});

app.get('/profile',(\_,resp)=>{

const user={

name:'Peter',

email:'peter@test.com',

country:'USA'

}

resp.render('profile',{user}) //profile – it is a pages names

});

app.get('/contact',(\_,resp)=>{

resp.sendFile(`${publicPath}/about.html`)

});

app.get('/help',(\_,resp)=>{

resp.sendFile(`${publicPath}/help.html`)

});

app.get('\*',(\_,resp)=>{

resp.sendFile(`${publicPath}/nopage.html`)

});

ejs template package

<https://www.npmjs.com/package/ejs/v/3.1.6>

Index.js  
const express = require();  
const path = require(‘path’);

const app = express();

const publicPath = path.join(\_\_dirname, ‘public’);

app.set(‘view engine’, ‘ejs’);

//first Parameter

We are using view engine

//second packages name

Which packages we are using - ejs

app.get(‘ ’, (\_, resp) => {

resp.sendFile(`${publicPath}/index.html `);

})

app.get(“/profile ”, (\_\_, resp) => {

const user = {

name : “anil sidhu”,

email : [anil@gmail.com](mailto:anil@gmail.com),

city : “noida”

}

resp.render(‘profile’, {user}); // as a object

})

app.get(“ /help”, (\_\_, resp) => {

resp.sendFile(${publicPath}/help.html);

})

app.listen(5000);

file folder structure

Public(folder)

views(folder)

by default configuration

profile.ejs(file)

<doctype html>

<head>

<style></style>

</head>

<body>

<h1>welcome <% = user.name %></h1>

<h1>email <% = user.email%></h1>

<h1>city <% = user.city %></h1>

<h1>profile page</h1>

</body>

<html>

Output :

Welcome anil sidhu

Email :anil@gmail.com

City : Noida

24. Dynamic Page with ejs

I will check this topic later

because middleware is important now

dynamic page

how to make a loop in ejs

make header file

show common header file

interview question

Index.js  
const express = require(‘express’);  
const path = require(‘path’);

const app = express();

const publicPath = path.join(\_\_dirname, ‘public’);

app.set(‘view engine’, ‘ejs’);

app.get(‘ ’, (\_, resp) => {

resp.sendFile(`${publicPath}/index.html `);

})

app.get(“/profile ”, (\_\_, resp) => {

const user = {

name : “anil sidhu”,

email : [anil@gmail.com](mailto:anil@gmail.com),

city : “noida”,

skills : [‘php’, ‘js’, ‘c++’]

}

resp.render(‘profile’, {user});

})

app.get(“ /help”, (\_\_, resp) => {

resp.sendFile(${publicPath}/help.html);

})

app.listen(5000);

profile.ejs(file)

profile.ejs(file)

<doctype html>

<head>

<style></style>

</head>

<body>

<h1>welcome <% = user.name %></h1>

<h1>email <% = user.email %></h1>

<h1>city <% = user.city %></h1>

<ul>

<% user.skills.forEach ((item) => ) { %>

<li> <% = item %> </li>

< % }) %>

</ul>

<h1>profile page</h1>

</body>

<html>

Output

Welcome, peter

[peter@test.com](mailto:peter@test.com)

USA

Php

Js

C++

Folder structure

Index.js

views(folder)

profile.ejs

login.ejs

public(folder)

about.html

help.html

index.html

nopage.html

login.ejs

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a login page</h1>

</body>

</html>

Index.js

const express = require();  
const path = require(‘path’);

const app = express();

const publicPath = path.join(\_\_dirname, ‘public’);

app.set(‘view engine’, ‘ejs’);

app.get(‘ ’, (\_, resp) => {

resp.sendFile(`${publicPath}/index.html `);

})

app.get(“/profile ”, (\_\_, resp) => {

const user = {

name : “anil sidhu”,

email : [anil@gmail.com](mailto:anil@gmail.com),

city : “noida”,

skills : [‘php’, ‘js’, ‘c++’]

}

resp.render(‘profile’, {user});

})

app.get(‘/login’, (\_\_, resp) => {

resp.render(‘login’);

})

app.get(“ /help”, (\_\_, resp) => {

resp.sendFile(${publicPath}/help.html);

})

//

app.listen(5000);

output:

localhost:5000/login

this is login page

//yha par common header chahiye

//common header krne ke liye ..kahi pe bhi file bna lo views : folder ke undar

folder structure

views

common(folder)

header.ejs

login.ejs

profile.ejs

header.ejs

<nav>

<h3>header file </h3>

</nav>

login.ejs

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<%- include(`common/header`) %>

<h1>This is a login page</h1>

</body>

</html>

//<%- include(`common/header`) %> 🡪 add hypen

Output

localhost:5000/login

Header File

This is a login page

we can also add this common page 🡪 in profile page as well

//<%- include(`common/header`) %> 🡪 add hypen

25. express js Middle Ware

What are middle ware

How to make a middleware

Apply middleware on routes

Types of middleware

Interview questions

Middleware ko routes ke uppar apply krte hai

Middleware ko hum bas routes ke sath hi use kr skte hai

Middlesware ye kuch functions hote hai : jo routes ke sath hi use hote hai

Eski help se jo request or response hota hai : usko access kr skte hai

Or usko modify kr skte hai

Bahut bar uthentication check krni hoti hai : user logged in hai yan hi hai

wo hum middle ware ke undar kr skte hai

specific country ke undar hamari site block krni hoti hai

Jaha pe bhi hume request or response ke undar kuch changes krne hote hai

wha pe middleware use krte hai

Request or response me changes krne hai to route me kr lo…middle ware me kyu krne hai…..????

Man lo ki meri application ke undar chahe wo website or chahe wo apis wale application ke undar 50 route hai..agar hum routes ke undar har bar check krne lg jau to ..same code 50 bar likhna pdega..

middleware ke sath kaisa hai ..ek bar mai middleware bna skta hu..or usko mai bar bar use kr skta hu.

folder structure

Index.js

const express = require(‘express’);

const app = express();

const reqFilter = (req, resp, next) => {

console.log(“request filter”);

next();

}

app.use(reqFilter);

app.get(‘ / ’, (req,resp) => ){

resp.send(‘welcome to home page’);

})

app.get(‘ / users’, (req,resp) => ){

resp.send(‘welcome to users page’);

})

app.listen(5000);

output

localhost:5000

Welcome to Home Page

middleware ke liye hum separate file bhi rkh skte hai

abhi same file me bna lete hai

types of middleware

Application level middleware

Router level middleware

ErrorHandling middleware

Build – in middleware

Third party middleware

localhost:5000

welcome to home page

: ek user ka age 18 se jyada hai

to hi hamari website ko access kr paye otherwise wo na kr paye

sbse phle ye check krte hai ki wo age dal bhi rha hai ki nhi dal rha user

request parameter ke undar

age urls ke undar dalwa lete hai…request parameter ke liye.

const reqFilter = (req, resp, next) =>{

if(!req.query.age){

resp.send(“Please Provide Age”);

}

Else{

next();

}

}

Output

localhost:5000

Please Provide Age

localhost:5000/users

Please Provide Age

localhost:5000/?age = 10

Welcome to home page

age check bhi krni hai uski age 18 honi chahiye

index.js

const express = require(“express”);

const app = express();

const reqFilter = (req, resp, next) => {

if(!req.query.age){

resp.send(“Please Provide Age”);

}

Else if (req.query.age<18){

Resp.send(“you can not access this page”);

}

Else{

next();

}

//next – next basically a function , as a function use krenge

Jo bhi route call kiya uske aage esko proceed kr fdega,

Agar ye wala function , agar ye wala function call nhi ho rha hai to aage proceed nhi krega.

app.get(‘/ ’ , (req, resp) =>{

resp.send(“welcome to home pages”);

})

app.get(“/users”, (req, resp) =>{

resp.send(“welcome to user pages”);

})

app.listen(3000);

}

Output

Condition1

localhost:5000

please provide age

Condition2

localhost:5000/?age = 10

you can not access this page

Condition3

localhost:5000/?age = 20

welcome to home page

APPLICATION LEVEL MIDDLEWARE

Ek specific group of route ke uppar nhi lg skta tha, pure ke pure application me jitne bhi routes honge sab ke uppar apply hota tha.

But ROUTE LEVEL MIDDLEWARE

Route level middleware ek route ke uppar bhi lg skta hai , group of routes ke uppar bhi lg skta hai

Or sare ke sare routes ke uppar bhi lg skta hai

Ek single routes ke uppar

Ya group of routes ke uppar

Sab ke sab routes ke uppar ek middleware kis tarike se lga skte hai

26. Route level Middleware Express Js

route level middleware

apply middleware on single route

make middleware in different file

apply middleware in the group of route

jab ek bar hum middle ware bna lete hai or use krte hai

wo sare ke sare route ke uppar ek bar apply ho jata tha

man lijiye humare pass 50 route hai unme se hume ek route uppar

two route ke uppar

or 5 route ke uppar middleware apply krna hai

to hum kis tarike se kr skte hai

pichle wale video me humne same file me middleware bnaya tha or use kiya tha

different file me middleware rkh ke usko kiss tarike se use kr skte hai

middle ware group of route ke uppar lagana hai..50 me se 10, 15 20 aise lagana hai

what is route level middleware

Aise middleware jo single route ke undar bhi kam krenge

Routing ke group ke undar bhi kam krenge..50 me se 10, 15 20 aise lagana hai

Sare ke sare middleware cover krne hai

sare ke sare route cover krne hai

ek middleware ke sath me

same middleware sare ke sare route me bhi lg jayega

Pichle video me

application level middleware

ek specific route or specific group of route ke undarn hi lgta tha.

Pure ke pure application me jitne bhi route honge

sbke uppar apply hota tha…

But now route level middleware hai wo ek route ke undar bhi lg skta hai

group of route me bhi lg skta hai

or sare ke sare routes ke undar bhi lg skta hai

Ek single route ke uppar

or group of route ke uppar

yap hir sab ke sab route ke uppar ek middleware kiss tarike se lga skte hai

example 1

const express = require(“express”);

const app = express();

const reqFilter = (req, resp, next) =>{

if(!req.query.name){

resp.send(“please provide your age”);

}

Else if(req.query.name<18) {

Resp.send(“you are under aged”)

}

Else{

next();

}

}

app.use(reqFilter);

app.get(‘/ ’ , (req, resp) =>{

resp.send(“welcome to home pages”);

})

app.get(“/users”, (req, resp) =>{

resp.send(“welcome to user pages”);

})

app.get(‘/about’, (req, resp) =>{

resp.send(“welcome to about pages”)

})

app.listen(3000);

}

Output :

localhost:3000

please provide your age

localhost:3000?age = 10

you are under aged

localhost:3000?age = 30

welcome to home page

--

localhost:3000/users

Please provide your age

localhost : 3000/users?age=10

you are under aged

localhost:3000/users?age=40

welcome to users page

--

Localhost:3000/about

Please enter your age

Localhost:3000/about?age = 10

You are under aged

Localhost:3000/about?age=40

Welcome to about page

Above example jo hai sare ke sare pages me apply ho rhe hai

Ab hum chahte hai ki sare ke sare page ke uppar restriction na lge

Home page, about page or user page . kisi ek ke upppar restriction lage

So we can do that

Example

Firstly remove application route

app.use(reqFilter);

Second – reqFilter –

Kisi specific route ke uppar lgana , whi pe lga do

example 1

const express = require(“express”);

const app = express();

const reqFilter = (req, resp, next) =>{

if(!req.query.name){

resp.send(“please provide your age”);

}

Else if(req.query.name<18) {

resp.send(“you are under aged”)

}

Else{

next();

}

}

app.get(‘/ ’ , (req, resp) =>{

resp.send(“welcome to home pages”);

})

app.get(“/users”, reqFilter, (req, resp) =>{ //this will work for only user routes other will run …………………………………………………………………//without it

resp.send(“welcome to user pages”);

})

app.get(‘/about’, (req, resp) =>{

resp.send(“welcome to about pages”)

})

app.listen(3000);

}

Output

Localhost : 3000

Welcome to home page

Localhost: 3000/about

Welcome to about page

Localhost : 3000/users // apply middleware on a single route

Please provide your age

Localhost:3000/users?age = 10

You are under ages

Localhost:3000/users?age = 40

Welcome to users page

//Done above : apply middleware on single route

Kya two ke uppar bhi apply kr skkte hai

app.get(“/users”, reqFilter, (req, resp) =>{ //this will work for only user routes other will run …………………………………………………………………//without it

resp.send(“welcome to user pages”);

})

app.get(‘/about’, ”, reqFilter, (req, resp) =>{

resp.send(“welcome to about pages”)

});

Output:

localhost:4500

please provide your age

localhost:4500?age = 10

you are under aged

localhost:4500?age=100

welcome to user pages

: ab ye about or user ke uppar aayega

:: middleware in the separate file

Index.js

middleware.js

Index.js

const express = require(“express”);

const reqFilter =require(“./middleware);

const app = express();

app.get(“ ”, (req, resp) =>{

resp.send(“welcome to home page”);

})

app.get(“/about”, reqFilter, (req, resp) =>{

resp.send(“welcome to about page”);

})

app.get(“/users”, reqFilter, (req, resp) =>{

resp.send(“welcome to users page”);

})

app.listen(3000);

middleware.js

module.exports = reqFilter = (req, resp, next) =>{

if(!req.query.name){

resp.send(“please provide your age”);

}

Else if (req.query.age< 18){

Resp.send(“you are unser aged”);

}

Else{

Next();

}

}

Output:  
users / about – route

Separate file me kyu rkhna hai, middleware hamre file me 20 , 25 unke liye ek separate file me bnana, ek good practice hai, ek accha project structure

Agar hamre pass 100, 200 routes hai

Hum chahte hai ki 20, 25 ke uppar apply ho , to ek ek ke uppar dalna , es tarike se sahi chijh nhi hai

Uske liye routed middleware ka ek

Agar 400, 500 me apply krna hai to

Routed middleware ka ek export krna padega

Example

Const express = require(“express”);

Const reqFilter = require(“./middleware”)

const route = express.Router();

const app = express();

route.use(reqFilter);//added

app.get(“ ”, (req, resp) =>{

resp.send(“welcome to home page”);

})

route.get(“/about, (req, resp) =>{ //remove reqFilter // add instance

resp.send(“welcome to about page”);

})

app.get(“/users”, (req, resp) =>{ //remove reqFilter // add instance

resp.send(“welcome to users page”);

})

route.get(“/contact”, (req, resp) =>{ //remove reqFilter

resp.send(“welcome to contact page”);

})

app.use(“/”, route); // added

app.listen(3000);

//instance route wala ,

const route = express.Router();

use jha pe middleware apply krna hi

//about

//contact

Enke uppar hi middleware apply hoga

Group route middleware

middleware.js

module.exports = reqFilter = (req, resp, next) =>{

if(!req.query.name){

resp.send(“please provide your age”);

}

Else if (req.query.age< 18){

Resp.send(“you are unser aged”);

}

Else{

Next();

}

}

Es video ko bad me ek barr or ache se dekhenge

//middle ware scenario

//1. middle ware apply for every pages

//2. middle ware apply alag alag pages

//ways to apply - for different different pages

//1. reqFilter , as a parameter

//app.use(reqFilter);

//2. as an route

27. Install MongoDB

download DB

install Mango DB

set environment for mango

mango db compass tool

interview question

browser download mango db

difference between mango db and mysql

mango db ka environment setup krna

gui too ka name

Cmd

show db

mango compass

gui

graphic user interface

how to install Mango DB

imp

<https://youtu.be/gB6WLkSrtJk>

programming knowledge codebind.com - imp

mango db compass is graphical user interface which act like a client for your mango db server.

Connecting to a default mongo DB server data base , you can use this default URLs,

mongodb://localhost:27017

Mondo DB Compass

Here we have few Databases

Admin

Config

Local

Some of the default databases : which are already present in Mongo DB

Database

Create Databases

Database name :

Collection name :

Cmd : Mongo

Mongo db shell download

Inside the mando db extract this zip files – and then set environment variable

cmd : mongosh

mongo shell is work this time

and we are inside the mango shell

it has connected to our default mango db databases – which is running on local host

here – we can run command line to fetch the database

or create some collections

: show dbs

Create new databases

: use database\_name

28 : mango db basic -1

what is mango db

mango db and mysql

mango db basic command

mango db is a no sql database

the data stored in a collection

collection do not have rows and columns

data is stored in the form of object

mango db np sql database hai

eske undar data hai wo tables ke undar store nhi krte ,, collection ke undar store krte hai

collection ke undar rows or columns nhi hote

to kiss form me data store hota hai

object ke form me

object kya hota hai

we can create array of an object or same concepts yha pe apply hota hai

waise hi object yha pe bnte hai

const person = [

{

Name : “monika”,

Skills : “javascript”

}]

object kya hota

we can create array of an object or same concepts ypa pe apply hota hai

waise hi objects yha pe bnte hai

const person = {

Name : “monika”,

Skills : “javascript”

}]

no sql ka matlab kya hai

oracle

mysql

mysql server

sare ke sare structured query language wale databases hote hai

proper structure hota hai

jiske undar queries apply krte hai

mango db

no sql

eska apna koi structure nhi hota

sometimes if we take colums and rows

sometimes we need extra colums and rows

that time no sql is perfect data would be

data store hota hai collection ke undar

collection ke undar rows or columns nhi hote

Jab ek table bnate hai to uske undar rows and columns hote hai

yha pe nhi hote

data object ke form me store hote hai

sql : row column

Mango db

Const person = [{

Name : “monika”,

Skills : “javascript”

}]

Cmd

mango

mango db compass

window command line

mango db compass

cmd

show dbs

create database

database name

collection name

create database using command line : mango db compass : MANGOSH

use youtube

jab tak database me koi bhi collection nhi hoti to yha pe youtube nhi dikhta

db.createCollection(‘videos’);

now created database

show dbs

collection check krni hai

show collection

db.createCollection(“playlist”);

delete : db.videos.drop();

show dbs

db.dropDatabase();

db

use admin

db

use admin

🡪28

Mango db basic

What is mango db

Mango db vs SQL

Mango db basic command

What is mango db

Mongodb is a nosql data base.

The data stored in a collection

Collection don't have row and columns

Data is stored in the form of object.

Oracle, MySQL, SQL this are structured query language

Eske undar proper structure hota hai.

Jiske undar queries apply krte hai

Mongo db no SQL

Eska apn koi structure nhi hota.

Check mango db in a two ways

Mango db compass

Cmd - mangodb sh

Show dbs

Create new database

Use database-name

Create collection

db.createCollection("videos");

Show dbs

show collections

Video's

db.createCollection("playlist");

db.videos.drop()// delete collection

Show dbs

Show collections

Show dbs

db.dropDatabase()

Data base delete

Db

Current db will show

Use admin

29. mango db crud operation

crud operation in mango db

how to insert data collection

how to check inserted data

how to update data

how to delete data

interview questions

in upcoming videos we will learn

: database connectivity

: authentication

: apis bnana   
: installation  
: basic operations : crud operations

interview question

How to use

mango db compass use kr lete hai : GUI tool

command line se kr lo

crud operation : create database : e-commerce(database name) : products(collection)

mango db compass : MANGOSH

insert data : add data / command line

Go to product table : insert data krna hai : MANGOSH

Mangosh :

use e-comm

E-comm :

db.products.insertOne({name : “m 40”,

Brand : “samsung”,

Price : 100,

Category : “mobile” });

Acknowledge true : insert ho gyi hai

check data using command line

mango db compass : MANGOSH

MANGOSH :

db.products.find();

now we can see data

update data

mango db compass : MANGOSH

e-comm : db.products.updateOne({name : ‘m 40’}, {$set :{ brand : “oppo”}});

db.products.updateOne({name:"abcd"}, {$set :{modal: "qwer"}});

// name : ‘m 40’

Delete data

mango db compass : MANGOSH

e-comm :

db.products.deleteOne({brand : “samsung”});

30 : Connect MANGO DB WITH NODE JS

<https://github.com/anil-sidhu/node-js/tree/mongoDb-node-connect>

<https://github.com/anil-sidhu/node-js/blob/mongoDb-node-connect/index.js>

CONNECT mango db with node js

install mango db package

show data from mango db

intervew question

browser mango db npm

Es package me aisa code likha hota hai jo ki node js ko mango db ke sath connect

npm I mango db

<https://www.npmjs.com/package/mongodb>

index.js

Const {MangoClient} = require(‘mangodb’);

Or

Const MangoClient = require(‘mangodb’).MangoClient;

const url = ‘mangodb://localhost:27017’;

const database = ‘e-comm’;

const client = new MongoClient(url);

async function getData(){

let result = await client.connect(); //promise return krta hai

let db = result.db(database);

let collection = db.collection(‘products’);

let response = await collection.find({}).toArray();

console.log(response);

}

getData();

Run :  
sare data object form me show honge

Interview question

kya hum do database ko ek sath connect kr skte hai : yes

Git hub code step by step node js and mango DB

const {MongoClient} = require('mongodb')

const url= 'mongodb://localhost:27017';

const databaseName='e-comm'

const client= new MongoClient(url);

async function getData()

{

    let result = await client.connect();

    db= result.db(databaseName);

    collection = db.collection('products');

    let data = await collection.find({}).toArray();

    console.log(data)

}

getData();

31 : Read Data From Mango DB

Read data from mango db

make file for db collection

Handle Promises

interview question

Example :

index.js

const {MongoClient} = require('mongodb');

const url= 'mongodb://localhost:27017';

const databaseName='e-comm';

const client= new MongoClient(url);

async function getData()

{

    let result = await client.connect();

    db= result.db(databaseName);

    collection = db.collection('products');

    let data = await collection.find({}).toArray();

    console.log(data)

}

getData();

Const {MangoClient} = require(‘mangodb’);

Or

Const MangoClient = require(‘mangodb’).MangoClient;

const url = ‘mangodb://localhost:27017’;

const database = ‘e-comm’;

const client = new MangoClient(url);

async function getData(){

let result = await client.connect(); //promise return krta hai

let db = result.db(database);

let collection = db.collection(‘products’);

let response = await collection.find({name : “ m 40 ”}).toArray(); // agar ek result chahiye kuch search krna hai to

console.log(response);

}

getData();

output:

data show ho jayega : ({name : “ m 40 ”});

: esko hum alag file me kaise rkhenge

Mongodb.js, index.js

index.js

Const {MangoClient} = require(‘mangodb’);

Or

Const MangoClient = require(‘mangodb’).MangoClient;

const url = ‘mangodb://localhost:27017’;

const database = ‘e-comm’;

const client = new MangoClient(url);

async function dbConnection(){

let result = await client.connect(); //promise return krta hai

let db = result.db(database);

return db.collection(‘products’);

comment//let collection = db.collection(‘products’);

comment//let response = await collection.find({name : “ m 40 ”}).toArray(); // agar ek result chahiye //kuch search krna hai to

comment//console.log(response);

}

// console.warn(dbConnection());//pending state//return promises

dbConnection.then((resp) =>{

resp.find().toArray().then((data) =>{

console.warn(data);

})

//for single data – node :

dbConnection.then((resp) =>{

resp.find().toArray({name: “nord”}).then((data) =>{

console.warn(data);

})

//when we use toArray : again it is returning promises

})

Another way : second way

Const main = async () =>{

Const data = await dbConnect();

Const data = await data.find().toArray();

console.warn(data);

}

main();

---------------------

Mongodb.js

const {MangoClient} = require(“mongodb”);

const url = “mangodb://localhost:27017”;

const databaseName = “e-comm”;

const client = new MongoClient(url);

async function dbConnect(){

let result = await client.connect();

db = result.db(databaseName);

return db.collection(“products”);

}

Module.exports = dbConnect;

Index.js

Const dbConnection = require(“./mangoDB”);

dbConnection.then((resp) =>{

resp.find({name: “nord”}).toArray().then((data) =>{

console.warn(data);

})

})

Const main = async () =>{

Const data = await dbConnect();

Const data = await data.find({name: “nord”}).toArray();

console.warn(data);

}

main();

32 : Insert Data From Mango DB

Insert data from mango db

Make new file for insert data

Import mango db connection

Insert single and multiple records

Files

Mongodb.js

Index.js

Insert.js

Mongodb.js

const {MangoClient} = require(“mongodb”);

const url = “mangodb://localhost:27017”;

const databaseName = “e-comm”;

const client = new MongoClient(url);

async function dbConnection(){

let result = await client.connect();

db = result.db(databaseName);

return db.collection(“products”);

}

Module.exports = dbConnect;

Index.js

Const dbConnection = require(“./mangoDB”);

dbConnection.then((resp) =>{

resp.find({name: “nord”}).toArray().then((data) =>{

console.warn(data);

})

})

Const main = async () =>{

Const data = await dbConnect();

Const data = await data.find({name: “nord”}).toArray();

console.warn(data);

}

main();

insert.js

const dbConnection = require(‘./mango’);

const insert = async () =>{

const db = await dbConnection(); //it will return a promise

const result = db.insert({

name : “note 5”,

brand : “vivo”,

price : 320,

category : “mobile”

});

console.log(“insert function”, result);

}

Insert();

Terminal

Nodemon ./insert.js

Command run already data insert ho gya hai

Insert.js

Const dbConnect = require(./mangodb);

Const insert = async () => {

Const db = await dbConnect();

Const result = await db.insert({

name : “note”,

brand : “vivo”,

price : 320,

category : “mobile”

});

If (result.acknoledged){

}

Console.log(“data inserted”);

}

Output :

Data inserted

Multiple result insert kran eke liye

As an array

[{

name : “note”,

brand : “vivo”,

price : 320,

category : “mobile”

}];

Node JS in Hindi #33 Update Data in MongoDB

Make new files for update data

Import MangoDB connection

Update Single and multiple records

Interviews questions

Mongodb.js

const {MangoClient} = require(“mongodb”);

const url = “mangodb://localhost:27017”;

const databaseName = “e-comm”;

const client = new MongoClient(url);

async function dbConnect(){

let result = await client.connect();

db = result.db(databaseName);

return db.collection(“products”);

}

Module.exports = dbConnect;

Update.js

Const dbConnect = require(./mango);

Const updateData = async () =>{

Let data = await dbConnect(); //console.warn(data);//promises return, apne app me async hota hai

Let result = data.updateOne( //phla object condition ke liye ho gya, or dusra set krne ke liye, jo ……………………………………………..//chichje change krni hai

{name : “max 5 ”} ,

{ $set : {name : “max pro 5”} })

}

updateData();

multiple records update

Const dbConnect = require(./mango);

Const updateData = async () =>{

Let data = await dbConnect(); //console.warn(data);//promises return, apne app me async hota hai

Let result = data.update( //phla object condition ke liye ho gya, or dusra set krne ke liye, jo ……………………………………………..//chichje change krni hai

{name : “max 5 ”} ,

{ $set : {name : “max pro 5”} })

}

updateData();

const dbConnect= require('./mongodb')

const updateData=async ()=>{

    let data = await dbConnect();

    let result = await data.update(

        { name:'max 5'},

        {

            $set:{name:'max pro 5', price:1000}

        }

        )

    console.log(result)

}

updateData();

output :

DATA UPDATE HO GYA

Node JS in Hindi #34 Delete Record in MongoDB

Make new file for delete data

Import mango db connection

Delete single and multiple records

Delete.js

Const dbConnect = require(“./mangodb”);

Const deleteData = async () =>{

Let Data = await db.connect(); //console.warn(data);  
let result = await data.deleteOne({

Name : “max pro 7 ”

});

Console.warn(“result”);

If(result.aknowledged){

Console.log(record deleted);

}

}

deleteData();

🡪delete multiple records

Const dbConnect = require(“./mangodb”);

Const deleteData = async () =>{

Let Data = await db.connect(); //console.warn(data);  
let result = await data.deleteMany({

Name : “max pro 7 ”

});

Console.warn(“result”);

If(result.aknowledged){

Console.log(record deleted);

}

}

deleteData();

output  
nodemon ./delete.js

Node JS in Hindi #34 Delete Record in MongoDB

Make new files for delete data

Import mangodb connection

Delete single and multiple records

Interview question

<https://github.com/anil-sidhu/node-js/tree/delete>

delete.js

const dbConnect = require(“./mongodb”);

const deleteData = async () =>{

console.log(“function called”);

let data = await dbConnect();

let result = await data.deleteOne({name : “max pro 7 ”}); //delete single result

console.warn(data);// data will return promises

if(result.aknowledged){

console.log(“record deleted”);

}

}

Output

nodemon delete.js

//delete multiple record

delete.js

const dbConnect = require(“./mongodb”);

const deleteData = async () =>{

console.log(“function called”);

let data = await dbConnect();

let result = await data.deleteMany({name : “max 7 ”}); //delete multiple result // max 7 ke sare ////////////////////////////////////////////////////////////data delete

console.warn(data);// data will return promises

if(result.aknowledged){

console.log(“record deleted”);

}

}

Node JS in Hindi #35 Basic GET API with MongoDB

Api with mongodb

Make new file for API

Import and use mongo Config

Make API for get data

Test with postman

Interviews questions

Mango db .js,

Api.js

Mongodb.js

const {MangoClient} = require(“mongodb”);

const url = “mangodb://localhost:27017”;

const databaseName = “e-comm”;

const client = new MongoClient(url);

async function dbConnect(){

let result = await client.connect();

db = result.db(databaseName);

return db.collection(“products”);

}

Module.exports = dbConnect;

App.js

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.get("/", (req, resp) =>{

resp.send({"name" : "monika"});

});

app.listen(3000);

output:3000

({"name" : "monika"});

That means api perfectlty chal rhi, ab eske undar data base ko import krni hai

App.js

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()        //return promise

console.log(data);

res.send(data);

});

app.listen(3000);

localhost.json

json data mil jayega : ye sare json data , databases ka hai , wo show ho rha hai

postman

methods – Get, Post, Put, Delete

api urls – for examples

https ://localhost:3000 🡪 get me click krte hai 🡪 sare data data se postman me show hoga

es tarike se get apis perfectly ban gyi hai

this is basic apis, later we will install mangoose , with mangoose – schemas , modals, sari ki sari chije bnayenge

interview –

get method me body pass nhi kr skte

browser me query, param pass kr skte hai urls se

Node JS in Hindi #36 Post API | insert data in MongoDB

Node js post api method

Make Post method for Apis

Send data from postman

Get data in node js by request

Write code for insert data in mango db

Interviews questions

Jab bhi database ke undar new data ko save krana hota hai, apis ke through tab hum post data ka use krte hai

Jab hume api ko through data ko read krna hota hai – tab hum get method use krte hai

Same data ko update krana hota hai – put

Data ko delete krana hai to delete method use krte hai

Databases ke undar hmare pass 8 entries hai

mangodb.js – code same as previous

App.js

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", (req, resp) =>{

resp.send({name : "anil"});

})

app.listen(3000);

lets check in postman

post method – <https://localhost:5000/>

postman post me click krne par

{name : “anil”}

That means post apis are working

Next question –

Postman se data ko send krke – nodejs me get kaise kr skte hai

Ye ek major chijj hai

Postman option-

Formdata

x-www-form-unicode data

node js me data ko send – json data hi save krte hai

raw – json data

dummuy data

{

“name” : “peter”

}

Check postman se data node js me aa bhi gya hai ki nhi, uske liye middleware use krna hota hai

Postman or request se data get krte hai to uske liye yha pe hume use krna hota hai

app.use(express.json());

App.js

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", (req, resp) =>{

console.log(req.body);

//resp.send({name : "anil"});

resp.send(req.body);

})

app.listen(3000);

output

run - node ./app.js

{

“name” : “peter”

}

Postman me post method me jo json data bheja tha wo aa rha hai

Hum log ye data ko mangodb me kaise save krayenge

Req.body – postman se or react or angular se node js ke undar data send krte hai

Postman me data add krte hai –

{

Name : “i phone”,

Brand : “I Phone 14”,

Price : “45,000$”,

Category : “Mobile”

}

Ab postman se esko send krte hai

Jaise hi send me click krenge

Acknolege true

insertCount:1

Ab hum mango db compass tool me check krenge to sara data postman se mangodb compass tool me aa jayega

Post api ki help se database ke undar entries kr skte hai

Abhi postman se data , agar postman me or bhi data add krenge to add ho jayegi , esliye es apis ke undar koi security nhi hai , koi schema nhi hai, basic apish ai

After that we will use mongoose , with the help of mongoose , make schema and then uske undar security tokens wagera add krenge

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", async(req, resp) =>{

//console.log(req.body);

let data = await dbConnect();

let result = await data.insert(req.body);

resp.send(result);

})

app.listen(3000);

Body parser npm

Node js me body parser, bahut phle, ab use nhi kiya,

Jo node express 4.6 me body parser

Express 4.6 ke bad wali uslke bad express.json();

app.use(express.json());

req.body parsor

Node JS in Hindi #37 PUT API | Update data in MongoDB

-put method – update data with Apis

- Send data from postman

- handle data in node js by Request

- write code for update data in MangoDB

- interview questions

Interview question – kya post method ke through data ko update kra skte hai

But standard way to update data using PUT METHOD

For example –

Ecomm.Products

Name

Brand]

Price

Update krwani hoti hai

Using put method we can do that

<https://github.com/anil-sidhu/node-js/tree/basic-post-api>

Api.js

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", async(req, resp) =>{

//console.log(req.body);

let data = await dbConnect();

let result = await data.insert(req.body);

resp.send(result);

})

//put method

app.put("/", (req, resp) =>{

resp.send({result:update});

});

app.listen(3000);

postman – put –

urls

<https://localhost:3000>

click on send button

status : 200

{

Result:”update”

}

Postman - Put- urls- body- row – json data

Json data

{

“name”:”I phone 15”,

“brand”: “I phone”,

“price”:5600,

“category”:”mobile”

}

ab esko get kaise kr skte hai

app.use(express.json());

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", async(req, resp) =>{

//console.log(req.body);

let data = await dbConnect();

let result = await data.insert(req.body);

resp.send(result);

})

//put method

app.put("/", (req, resp) =>{

    console.log(req.body);

resp.send({result:update});

})

app.listen(3000);

postman me send me click krne par

orr

terminal – nodemon ./App.js

so postman se data terminal me show ho jayega

postman se nodejs me data aa gya – ab hum esko update kaise krwa skte hai

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", async(req, resp) =>{

//console.log(req.body);

let data = await dbConnect();

let result = await data.insert(req.body);

resp.send(result);

})

//put method

app.put("/",async(req, resp) =>{

    let data = await dbConnect();

    let result = data.updateOne(

        {name : "I Phone 13"},

        {$set : {price : 900}}

    )

    console.log(req.body);

resp.send({result:”update”});

})

app.listen(3000);

output

abhi MongoDB Compass me data update

hit postman – perfectly working –

{

Result : “Update”

}

Ye humne static deta dal diya – ab eske jagah dynamic data dalna hoto kya kre

Dynamic data postman se aa rha

 let result = data.updateOne(

        {name : "I Phone 13"},

        {$set : req.body}

    )

const express = require('express');

const dbConnect = require("./mangodb");

const app = express();

app.use(express.json());

app.get("/", async(req, res) =>{

let data = await dbConnect(); //return promise

data = data.find().toArray()  //return promise

console.log(data);

res.send(data);

});

app.post("/", async(req, resp) =>{

//console.log(req.body);

let data = await dbConnect();

let result = await data.insert(req.body);

resp.send(result);

})

//put method

app.put("/",async(req, resp) =>{

    let data = await dbConnect();

    let result = data.updateOne(

        {name : "I Phone 13"},

        {$set : req.body}

    )

    console.log(req.body);

resp.send({result:update});

})

app.listen(3000);

output :

postman – click on send – update ho gya

{

“name”:”I phone 15”,

“brand”: “I phone”,

“price”:5600,

“category”:”mobile”

}

Mongodb compass me bhi update ho gya

Condition Dynamic

{

“name”:”I phone 15”,

“brand”: “I phone”,

“price”:5600,

“category”:”mobile”

}

//put method

app.put("/",async(req, resp) =>{

    let data = await dbConnect();

    let result = data.updateOne(

        {name : req.body.name},

        {$set : req.body}

    )

    console.log(req.body);

resp.send({result:update});

})

Postman click on send – successful

Check – mongoDB Compass –

Data update ho jayega

Parameter se hum chahte hai ki - data jaye –

Query param se data ko kiss tarike se send kr skte hai or kiss tarike se le skte hai

//put method

app.put("/",async(req, resp) =>{

    let data = await dbConnect();

    let result = data.updateOne(

        {name : req.body.name},

        {$set : req.body}

    )

    console.log(req.body);

resp.send({result:update});

})

Postman

{

“Name” : “IPhone 13”,

“Brand”: “apple”,

“Price” : 100,

“Category” : “mobile”

}

Mongodb compass me update ho jayega 100

PARAMTER se chahte hai data jaye – parameter ke undar ho jo actual data hai

Actual data parameter me , updated data postman xse jaye

Because ho skta hai , name hi update kr rhe hai,

Postman – put – urls

<https://localhost:3000/Iphone13>

postman

{

“Name “ : “iphone 12”,

“Price”: 12000,

“Brand”:”iphone”

“Category”:”mobile”

}

//put method

app.put("/: name",async(req, resp) =>{

    let data = await dbConnect();

    let result = data.updateOne(

        {name : req.params.name},

        {$set : req.body}

    )

    console.log(req.body);

resp.send({result:update});

})

app.listen(3000);

ab mangodb compass me check krenge to

name,

price update ho jayega

actual data parameter se send kiye

query params ke undar data ko kis tarike se send kr skte hai orrr or kis tarike se le skte hai

Node JS in Hindi #38 Delete API | delete data from MongoDB

Node js delete api method

Make delete method from apis

Send data from postman

Handle data in node js by request

Write code for delete data in mango DB

Interview question

//delete method

app.delete("/:id", (req, resp) =>{

console.log(req.params.id);

resp.send("done");

})

Postman – urls – delete

Delete – <https://localhost:3000/id-adssgfdgffhgh> (id mangodb compass se copy kiye hai)

Application runnn – terminal –

Node ./App.js – enter

Id console.log me mil gyi

Postman

Delete – <https://localhost:3000/id-adssgfdgffhgh>

Click on run ,

Done

Id mil gyi, ab usko delete krte hai

//delete method

app.delete("/:id", async(req, resp) =>{

console.log(req.params.id);

const data = await dbConnect();

const result = await data.deleteOne({\_id : new mangodb.ObjectId(req.params.id)}); //const mangodb = require("mangodb");

resp.send("done");

})

app.listen(3000);

Mangodb compass se data delete ho jayega

Node JS in Hindi #39 Mongoose with node

What is mangoose

Difference between mangoose vs mangodb packages

Install mangoose

What is schemas

What is model

Connect node js and mangoDB with mangoose

Node js ko mango db se connect krne ke liye npm packages

MangoDB npm

Mangoose npm

MangoDB npm me kuch limitation hai ,usko puri krta hai Mangoose npm

MangoDB npm as compared to this ,Mangoose npm advanced chije deta hai

Jab bhi hum mango db use krte the uske undar modal or schemas use nhi kr pate the

For example databases we have

Id

Name

Brand

Price

Or hum nhi chahte fifth field ko fill kare

Only 4 hi fixed field honi chahiye id, name, brand, price

To mangoose me schema bnake usko rok skte hai , 5th field enter hone se, but mangodb ke undarn hi rok skte

Mangoose – hum uske types defines kr skte hai , jaise ki

For example – price – it is a number

String and number ko identifies kr skte hai, validation apply kr skte hai

Mangoose – node js se mango DB Ko connect krne ke liye , va.idation, extra field

Schemas

Model

<https://www.npmjs.com/package/mongoose>

install

npm i mongoose

index.js

document insert jaha pe extra field ko enter krne se rokna

schemas - ?

database ki field hoti hai usko , validate krti hai

we have to define database me koun koun si field hogi

id,

name,

brand,

price,

categories

5 field hi bnana hai , essse mango db kp smujh me aa jayega mujhe 5 hi field enter krni hai

Agar 6 , 7, 8 field enter krne ki kosis krta hai to wo mangoose nhi hone dega

Or hum schemas ke undar validation lgayenge

Yha pe type validation bhi lgate hai

Modal

Node js ko mango db se connect, structure hota hai pura ka pura

Schema

Hamare jitni bhi field hai usko schemas bolte hai , datanbase ke undar fild hai usko define krte hai data bases ke undarrr field hai wo define krte hai , schema ke undarrr

Taki validation apply ho paye

Modal

Modal basically un schemas ko use krke

Node js or mango DB ko connect krta hai

Output – nodemon

Object ki id plus name – mongo dB compass or terminal me check kr skte Hai

const mongoose = require("mongoose");

const main = async() =>{

    await mongoose.connect("mongodb://localhost:27017/e-comm");//databases name

    const productSchema = new mongoose.Schema({

        name : String,

        price : Number

    });

    const productmodels = mongoose.model("products", productSchema);

    let data = new productmodels({name : "m8", price : 1000});

    let result = await data.save();

    console.log(result);

}

main();

Node JS in Hindi #40 CRUD with Mongoose

Update record

Delete record

Find and read result

// const express = require("express");

// const app = express();

// app.get(" ", (req, res) => {

// res.send("hello this is home page");

// });

// app.listen(3000);

const mongoose = require("mongoose");

const main = async() =>{

    await mongoose.connect("mongodb://localhost:27017/e-comm");//databases name

    const productSchema = new mongoose.Schema({

        name : String,

        price : Number,

        brand : String,

        category : String

    });

    const productmodels = mongoose.model("products", productSchema);

    let data = new productmodels(

        {name : "m8",

        price : 1000,

        brand : Max,

        category : "mobile" });

    let result = await data.save();

    console.log(result);

}

main();

terminal

nodemon

sare ke sare field enter ho jayegi koi problem nhi aayegi

mangodb compass me check krenge

sare ke sare field aa jayegi

es tarike se schema define kr skte hai

ab esko update krne ki kosis krenge

UPDATE

es tatike se data ko update

// const express = require("express");

// const app = express();

// app.get(" ", (req, res) => {

// res.send("hello this is home page");

// });

// app.listen(3000);

const mongoose = require("mongoose");

const main = async() =>{

    await mongoose.connect("mongodb://localhost:27017/e-comm");//databases name

    const productSchema = new mongoose.Schema({

        name : String,

        price : Number,

        brand : String,

        category : String

    });

    const productmodels = mongoose.model("products", productSchema);

    let data = new productmodels(

        {name : "m8",

        price : 1000,

        brand : Max,

        category : "mobile" });

    let result = await data.save();

    console.log(result);

}

    const updateDB  = async () =>{

        const Product = mongoose.model("products", productSchema);

        let data = await Product.updateOne(

           {name : "max 7"} ,

           {

            $set   : {price : 700}

          }

        )

          console.log(data);

    }

    updateDB();

main();

DELETE

 const Product = mongoose.model("products", productSchema);

// const express = require("express");

// const app = express();

// app.get(" ", (req, res) => {

// res.send("hello this is home page");

// });

// app.listen(3000);

const mongoose = require("mongoose");

const main = async() =>{

    await mongoose.connect("mongodb://localhost:27017/e-comm");//databases name

    const productSchema = new mongoose.Schema({

        name : String,

        price : Number,

        brand : String,

        category : String

    });

    const productmodels = mongoose.model("products", productSchema);

    let data = new productmodels(

        {name : "m8",

        price : 1000,

        brand : Max,

        category : "mobile" });

    let result = await data.save();

    console.log(result);

}

    const updateINDB  = async () =>{

        const Product = mongoose.model("products", productSchema);

        let data = await Product.updateOne(

           {name : "max 7"} ,

           {

            $set   : {price : 700}

          }

        )

          console.log(data);

    }

    updateINDB();

main();

    const deleteINDB = async() =>{

        const Product = mongoose.model("products", productSchema);

        let data = await Product.deleteOne({ {name : "max 7"}});

        console.log(data);

    }

    deleteINDB();

max : 7 tha wo delete ho chuka hai

find – findinDB

// const express = require("express");

// const app = express();

// app.get(" ", (req, res) => {

// res.send("hello this is home page");

// });

// app.listen(3000);

const mongoose = require("mongoose");

const main = async() =>{

    await mongoose.connect("mongodb://localhost:27017/e-comm");//databases name

    const productSchema = new mongoose.Schema({

        name : String,

        price : Number,

        brand : String,

        category : String

    });

    const productmodels = mongoose.model("products", productSchema);

    let data = new productmodels(

        {name : "m8",

        price : 1000,

        brand : Max,

        category : "mobile" });

    let result = await data.save();

    console.log(result);

}

    const updateINDB  = async () =>{

   const Product = mongoose.model("products", productSchema);

        let data = await Product.updateOne(

           {name : "max 7"} ,

           {

            $set   : {price : 700}

          }

        )

          console.log(data);

    }

    const deleteINDB = async() =>{

     const Product = mongoose.model("products", productSchema);

        let data = await Product.deleteOne({ {name : "max 7"}});

        console.log(data);

    }

    const findINDB = async() =>{

        const Product = mongoose.model("products", productSchema);

        let data = await Product.find();

        console.log(data)

    }

    findINDB();

    deleteINDB();

    updateINDB();

    main();

output:

mongoDB Compass se sara data nikal ke aa jayegi

specific data ko find krna hai to

 const findINDB = async() =>{

  const Product = mongoose.model("products", productSchema);

        let data = await Product.find({name : "max 131"});

        console.log(data)

    }

Jitne bhi eske related record honge sare ke sare hamare pass aa jayenge

  let data = await Product.find({name : "max 131"});

Node JS in Hindi #41 Post API with mongoose , How to make post API in Node js

Post api with mongoose

Make config file for mangoDB

Make post route

Get data from the postman and save in DB

Interview questions

Config.js

Product.js

Model

Or schema define krenge

Index.js

Postman - urls – <https://localhost:5000/create>

Body

{

“Name” : “Samsung m 40”,

“Price” : 400,

Brand : “samsung”,

Category : “mobiles”

}

Jaise hi postman se send me click kiya – pura ka pura data perfectly aa jayega

Product.js

const mongoose = require("mongoose");

const productSchema = new mongoose.Schema({

    name : String,

    price : Number,

    brand : String,

    Category : String

});//schema

module.exports = mongoose.model('products', productSchema);//model

config.js

const mongoose = require("mongoose");

mongoose.connect("mongodb://localhost:27017/e-comm");//db name

index.js

const express= require("express");

require("./Config");

const Product = require("./product");

const app = express();

app.use(express.json());

app.post("/create", (req, resp) =>{

console.log(req.body);

resp.send("Done");

});

app.listen(5000);

terminal

nodemon

terminal me data postman se aa jayega, data data data

ab esko database me kaise store krna hai

index.js

const express= require("express");

require("./Config");

const Product = require("./product");

const app = express();

app.use(express.json());

app.post("/create", async(req, resp) =>{

    let data = new Product(req.body);

    let result = await data.save();

    console.log(result);

    resp.send(result);

});

app.listen(5000);

databases me data addd ho jayegi

Node JS in Hindi #42 GET, DELETE and PUT api method with mongoose

Example of get method APIS

Example of put method apis

Example of delete method apis

Index.js

const express= require("express");

require("./Config");

const Product = require("./product");

const app = express();

app.use(express.json());

app.post("/create", async(req, resp) =>{

    let data = new Product(req.body);

    let result = await data.save();

    console.log(result);

    resp.send(result);

});

app.get("/list", async(req, resp) =>{

let data = await Product.find();

resp.send(data);

});

app.listen(5000);

postman :

<http://localhost:3000/list>

send me click krne par data perfectly aa jayega

mongoDB Compass me bhi 3 hi data hai

DELETE METHOD

app.delete("/delete/:\_id", async(req, resp) =>{

console.log(req.params);

let data = await Product.deleteOne(req.params);

resp.send(data);

})

Postman

<https://localhost:3000/delete/idass45676576df-id-particular>-id

send me click krne parr everything delete ho jayega

terminal –

id print ho jayegi

Update – Put Methods

Postman

Put request – urls –

<https://localhost:5000/update/idrandomididif566788755577757858id>

esme body bhejhni pdegi

only price hi change krni hai

{

Price: 200

}

app.put("/update/:\_id",async (req, resp) => {

    console.log(req.params)

    let data = await Product.updateOne(

        req.params,

        {$set: req.body}

    );

    resp.send(data);

});

Flow

terminal – data -

Postman – hit the browser

const express = require('express');

require("./config");

const Product = require('./product');

const app = express();

app.use(express.json());

app.post("/create", async (req, resp) => {

    let data = new Product(req.body);

    const result = await data.save();

    resp.send(result);

});

app.get("/list", async (req, resp) => {

    let data = await Product.find();

    resp.send(data);

})

app.delete("/delete/:\_id", async (req, resp) => {

    console.log(req.params)

    let data = await Product.deleteOne(req.params);

    resp.send(data);

})

app.put("/update/:\_id",async (req, resp) => {

    console.log(req.params)

    let data = await Product.updateOne(

        req.params,

        {$set: req.body}

    );

    resp.send(data);

})

app.listen(5000)

Node JS in Hindi #43 Search API with multiple filed

Make simple get route for APIS

Search with single fields

Search with Multiple field

Test APIS

🡪Terminal – nodemon run

🡪Postman

const express = require('express');

require("./config");

const Product = require('./product');

const app = express();

app.use(express.json());

app.get("/search/:key",async (req,resp)=>{

 let data = await Product.find(

        {

            "$or":[

                {name:{$regex:req.params.key}},

                {brand:{$regex:req.params.key}}

            ]

        }

    )

    resp.send(data);

})

app.listen(5000);

Node JS in Hindi #44 Upload File

Install multer npm packages

Make router for upload files

Write code for upload files

Interview questions

Basically how files basically upload

<https://www.npmjs.com/package/multer>

npm I multer

const express = require("express");

const multer = require("multer");

const app = express();

app.post('upload', (req, resp) =>{

resp.send('file upload');

});

app.listen(5000);

postman – htttp://localhost:5000/upload

Node JS in Hindi #45 OS module

Node JS in Hindi #46 Events and Event Emitter

Node JS in Hindi #47 REPL - Read-Eval-Print-Loop  
  
Node JS in Hindi #48 connect with mysql

Download xampp php mysql  
<https://www.npmjs.com/package/mysql>

npm i mysql  
<https://github.com/anil-sidhu/node-js/tree/mysql-connect>

Node JS in Hindi #49 GET API with MySQL  
<https://github.com/anil-sidhu/node-js/tree/mysql-get-api>

Node JS in Hindi #50 POST API with MySQL

Node JS in Hindi #52 DELETE API with MySQL