GET, POST, PUT,DELETE in express

Hana Aden documentation

Hey I am Hana Aden today I wanna guide you doing the 4 magic words of express
1: get : use that when you like to get/read something like data from the server side
2: put : is update existing data on the server side
3: delete : is the deleting the data on server side
4 : post : adding or creating data to you server side data like database

The first thing that I tried to understand is What is express ?
Express is Framework that handles HTTP request in simple way you can say it is a tool that helps make a server in node js
It is like helper that handles requests and responses between clients to servers

Since you understand that let we start
First I created a folder and I named it express CRUD after that I created a file inside that folder and I named it server.js
I started coding immediately the first line const express = require(“express”) what happened was Vs code saying wait “who is express” then I figured out that I should install express in my folder
Better for you to do so now write this command as I did
In my terminal :
PS D:\Hana\online study\visual st\Express crud>npm init -y
PS D:\Hana\online study\visual st\Express crud>npm install express

See now you see the magic you got the package json in your Folder as do I after that VS code recognized the express
Now I have express installed package json installed and my first line of code
const express = require(“express”)
Now I added
Const app = express( ) that means I created express app ahead the third line is app.use(express.json( )) means any time that some sends a JSON data to the server side please change it into javaScript object so I can send responses and use it as req.body without this line the req.body will stay undefined
Now we have our first 3 lines

Const express = require(“express”)

Const app = express( )
app.use(express.json( ))

Let we go ahead what we are waiting we need a data so let we create a fake database by our selves
Let users = [
{id : 1 , name : “John”},
{id :2 , name : “Jane”}
]
As Basic javaScript the word [LET] allowing as the changes means if we make const the PUT in express is impossible as well as POST and what we have is array inside it a objects so that is our database by now

Now let we start the four words but before that we need to know or check if our server is running
So start the server
app.listen(5000, ( ) => console.log(“the server is running on port 5000”))

Let we prepare the postman
The postman is tool helps us to test our APIs you can directly send requests to you server side instead of building the frontend everytime
Click new in your post man and paste the URL “http://localhost:5000/users”
Before pasting like we do the get in our server side
Now above you app.listen(5000, ( ) => console.log(“the server is running on port 5000”))
Create your POST GET PUT DELETE things
Let we fo the GET then code with me :
app.get(“/users” , (req ,res)=>{
res.json(users)
}
Now you wonder what is req , res and “/users” the “/users” is route path that you add you api see “http://localhost:5000/users” now req(request) is all incoming information about the request the client made example req.body.name you will get name and res(response) means is what you send to the client side

It get method see you send all users in your database to the client side that is why you used res

No in your terminal :
PS D:\Hana\online study\visual st\Express crud> node server.js

You have to get
the server is running on port 5000

Ok let we go the postman paste this URL “http://localhost:5000/users” choose get and click send

So you will get in your postman this

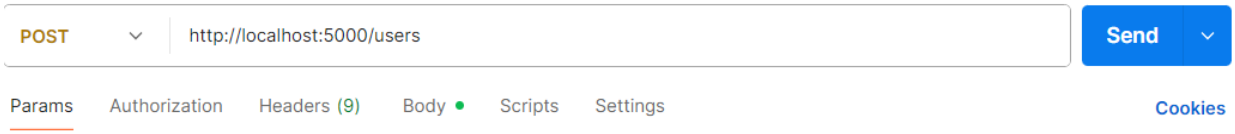
One my first mistake the postman was giving we error saying not found while now is giving me response look I figured our running the server always will gonna solve the problem

Now let me go to the POST
app.post(“/users” , (req ,res)=>{
Const newUser = {
id : users.length + 1 ,
name : req.body.name
}
users.push(newUser)
res.status(201).json(newUser)
}

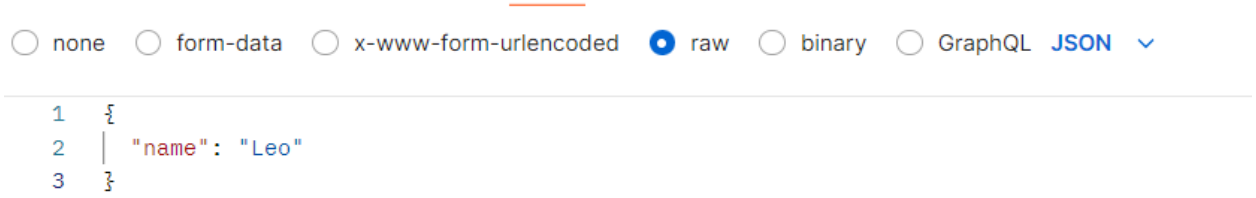
EXPLANATION
app.post : creates POST route in express

“/users” : this is URL path as you know so when someone sends request to http://localhost:5000/users this code will run
Users.length + 1 : automatically gives id
req.body.name : read the name requested in the body
And pushes into your user

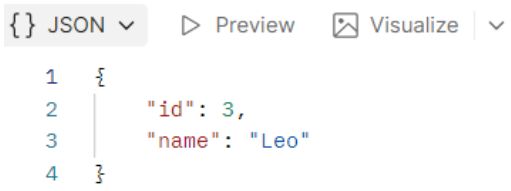
Now go to your postman in this URL http://localhost:5000/users choose POST then body -> row then JSON



Choose body then row then JSON
And write



Click send
Now you see in your postman

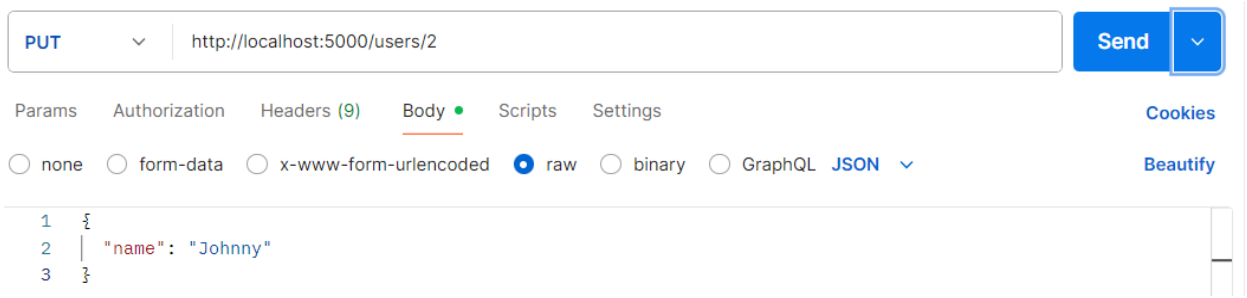


Why we are not going back to get and see if our database is updated

See now I added a user

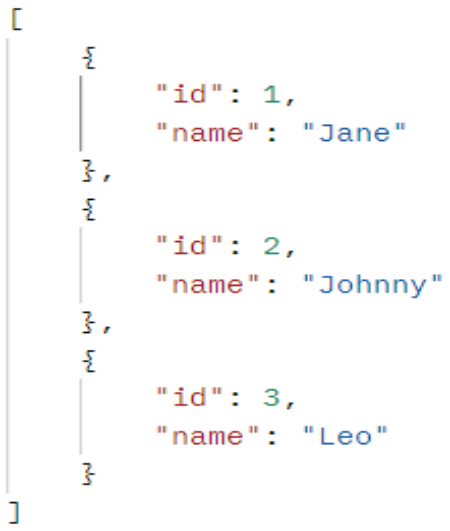
Let we go to PUT
As usual do but in our route we add ID so we have to get the user id that we are updating

app.post("/users/:id" , (req ,res)=>{
Const userId = parseInt(req.params.id)
Const user = users.find(u=>u.id === userId)
if(!user) return res.status(404).json({messege: "user not found"})
user.name = req.body.name
res.json(user)
}
Now in your postman copy the link http://localhost:5000/users/2
Choose PUT => Body =.> row => json



You will get in your post man

Now check again the get and see the update



Now let we do the DELETE the only one that is remaining

app.delete(“/users/:id”){

Const userId = parseInt(req.params.id)

Const userIndex = users.findIndex(u=>u.id === userId)

if (user.id === -1) return res.status(404).json({messege: “user not found”})

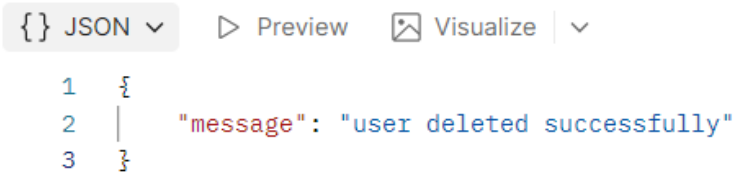
users.splice(userIndex , 1)

res.json({messege : “user deleted successfully”})

}

Now our postman

The output



Let we get our users

Hope this documentation helped as intended

FULL CODE

```
const express = require("express")
const app = express()
app.use(express.json())

let users = [
    {id: 1 , name : "Jane"},
    {id: 2 , name : "John"}
]


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

app.post("/users" , (req ,res)=>{
    const newUser = {
        id : users.length + 1 ,
        name : req.body.name
    }
    users.push(newUser)
    res.status(201).json(newUser)
})

app.put("/users:id" , (req ,res)=>{
    const userId = parseInt(req.params.id)
    const user = users.find(u=>u.id === userId)
    if(!user){
    return res.status(404).json({messege : "user not found"})
    }
    user.name = req.body.name
    res.json(user)
})

app.delete("/users/:id" ,(req ,res)=>{
    const userId = parseInt(req.params.id)
    const userIndex = users.findIndex(u => u.id === userId)
    if(userId === -1){
```

```
        return res.status(303).json({messege : "user not found"})

    }
    users.splice(userIndex , 1)
    res.json({messege : "deleted successfully"})
})
app.listen(5000, ()=> console.log("server is running on port 5000"))
```