

RESTFUL OPERATION(devops)

M.Nishitha

2303A51540
bt no 19

simple Restfull operation

```
const express = require('express');
const app = express();
app.use(express.json());
app.get('/students', (req, res) => {
  res.json([ { id: 1, name: "Ravi" } ]);
});
app.post('/students', (req, res) => {
  const student = req.body;
  res.status(201).json({ message: "Student added", data: student });
});
app.listen(3000, () => {
  console.log("Server running on port 3000");
});
```

restfull operation with HTML

```
const express = require('express');
const app = express();
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
let students = [];
app.get('/', (req, res) => {
  res.send(`<h2>Add Student</h2>
<form method="POST" action="/students">
<input type="text" name="name" placeholder="Enter name" />
<button type="submit">Add</button>
```

```
</form>

');

});

app.post('/students', (req, res) => {

const name = req.body.name;

const newStudent = {

id: students.length + 1,

name: name

};

students.push(newStudent);

res.send("Student Added Successfully!");

});

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

res.json(students);

});

app.listen(3000, () => {

console.log("Server running on port 3000");

});

#### resfull operation with POSTMAN desktop app(https://www.postman.com/downloads/)

const express = require('express');

const app = express();

app.use(express.json());

let students = [

{ id: 1, name: "Ravi" },

{ id: 2, name: "Sita" }

];

// 1 GET - View all students

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

res.json(students);

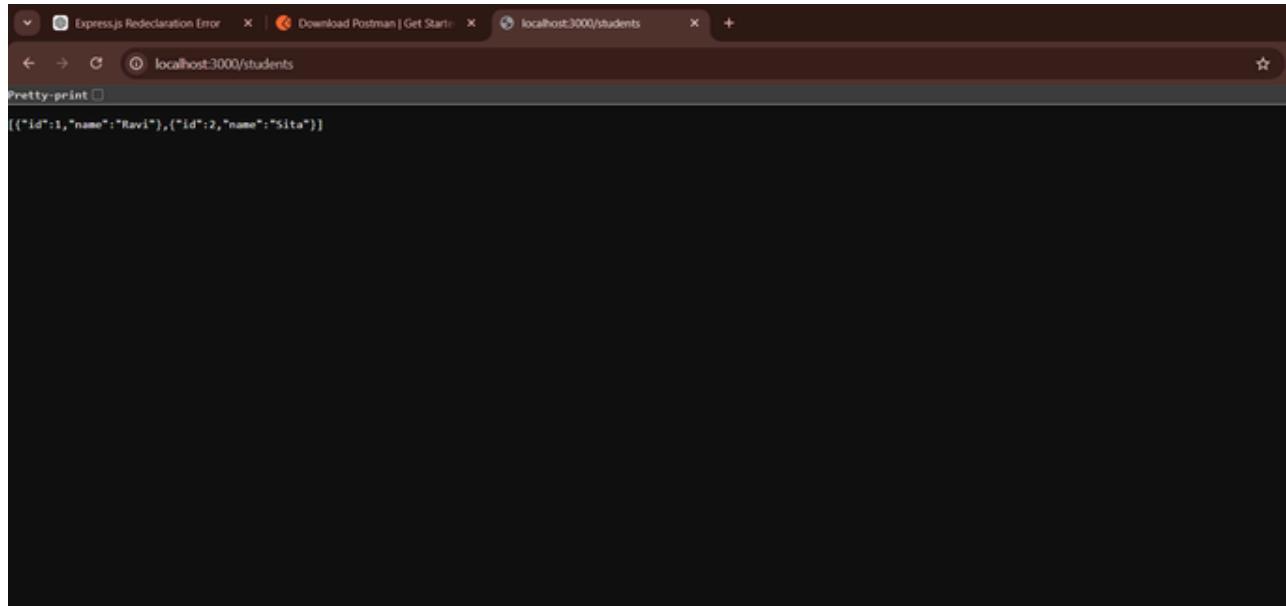
}); GET OPERATION:
```

/ 1 GET - View all students

```
app.get('/students', (req, res) => {  
  res.json(students);  
});
```

OUTPUT:

```
dev2.html          package-lock.json      student2.js  
dev01.js          package.json  
node_modules      result.js  
nishithamarepally@nishithas-MacBook-Air devopps % node result.js  
Server running on port 3000  
SyntaxError: Bad control character in string literal in JSON at position 20 (line 2 column 19)  
    at JSON.parse (<anonymous>)  
    at parse (/Users/nishithamarepally/Desktop/devopps/node_modules/body-parser/lib/types/json.js:72:19)  
    at /Users/nishithamarepally/Desktop/devopps/node_modules/body-parser/lib/read.js:162:18  
    at AsyncResource.runInAsyncScope (node:async_hooks:214:14)  
    at invokeCallback (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:238:16)  
    at done (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:227:7)  
    at IncomingMessage.onEnd (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:287:7)  
    at IncomingMessage.emit (node:events:508:28)  
    at endReadableNT (node:internal/streams/readable:1701:12)  
    at process.processTicksAndRejections (node:internal/process/task_queues:89:2)  
1)
```



! We're updating our plans and pricing on March 1. See [our blog](#) for more details.

The screenshot shows the Postman interface. On the left, the sidebar includes 'Collections' (selected), 'Environments', 'History', 'Flows', and 'Files (BETA)'. The main area shows a 'POST Get' request to 'http://localhost:3000/students/3'. The 'Params' tab is selected, showing a single parameter 'id' with value '3'. The 'Body' tab shows a JSON response: 'message': 'Student deleted'. To the right, a 'Visualize Response as Table' section contains a script to convert the JSON response into an HTML table. An approval dialog for 'Get data' is open, asking 'Allow agent to update post response of request?' with 'Approve' and 'Reject' buttons. A status bar at the bottom indicates 'Auto'.

// 2 GET - View single student

```
app.get('/students/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const student = students.find(s => s.id === id);
  if (!student) {
    return res.status(404).json({ message: "Student not found" });
  }
  res.json(student);
});
```

The screenshot shows a browser window with three tabs open. The active tab is titled 'localhost:3000/students' and displays the following JSON data:

```
[{"id":1,"name":"Ravi"}, {"id":2,"name":"Sita"}, {"id":6,"name":"koushik"}, {"id":4,"name":"koushik"}, {"id":5,"name":"koushik"}]
```

OUTPUT:

The screenshot shows the Postman application interface. A collection named 'My Collection' is selected. A 'GET' request is defined with the URL 'http://localhost:3000/students'. The response body is displayed as a table:

ID	Name
0	Ravi
1	Sita
2	koushik
3	koushik
4	koushik

The right panel of Postman contains a troubleshooting section with the following points:

1. Wrong HTTP Method: The request is using DELETE, but based on your description (which mentions "This is a POST request, submitting data to an API via the request body"), this should likely be a POST request.
2. Request Body: You have a JSON body `{"name": "koushik"}` which is typical for POST requests, not DELETE requests.
3. 404 Error: The server is returning "Cannot DELETE /students", which suggests the endpoint doesn't support DELETE operations.

Below the list, there's a section titled "What would you like me to fix?" with three options:

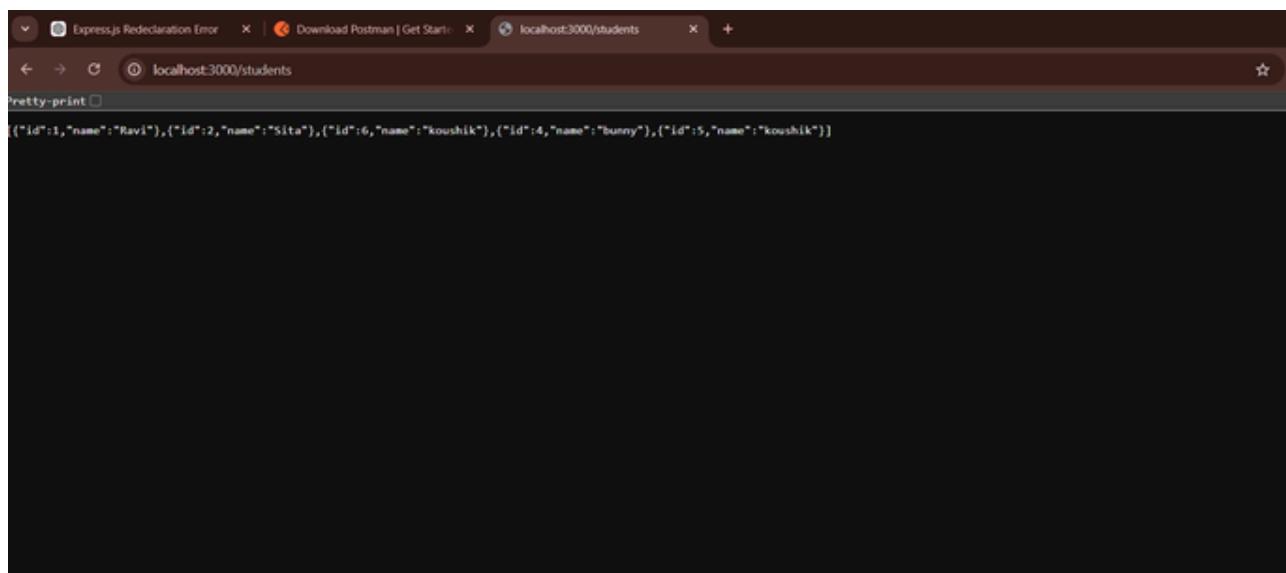
- Should I change this to a POST request instead of DELETE?
- Or do you want to keep it as DELETE but change the endpoint to something like /students/{id} to delete a specific student?
- Or is there a different endpoint you should be using?

At the bottom, it says "Let me know which direction you'd like to go, and I'll update the request for you!"

/ 4 PUT - Update student

```
app.put('/students/:id', (req, res) => {  
  const id = parseInt(req.params.id);  
  const { name } = req.body;  
  const student = students.find(s => s.id === id);  
  if (!student) {  
    return res.status(404).json({ message: "Student not found" });  
  }  
  student.name = name;  
  res.json({ message: "Student updated", student });
```

}); OUTPUT:



1. Wrong HTTP Method: The request is using DELETE, but based on your description (which mentions "This is a POST request, submitting data to an API via the request body"), this should likely be a POST request.

2. Request Body: You have a JSON body `[{"name": "koushik"}]` which is typical for POST requests, not DELETE requests.

3. 404 Error: The server is returning "Cannot DELETE /students", which suggests the endpoint doesn't support DELETE operations.

What would you like me to fix?

- Should I change this to a POST request instead of DELETE?
- Or do you want to keep it as DELETE but change the endpoint to something like /students/[id] to delete a specific student?
- Or is there a different endpoint you should be using?

Let me know which direction you'd like to go, and I'll update the request for you!

// 5 DELETE - Remove student

```
app.delete('/students/:id', (req, res) => {
  const id = parseInt(req.params.id);
  students = students.filter(s => s.id !== id);
  res.json({ message: "Student deleted" });
});

app.listen(3000, () => {
  console.log("Server running on port 3000");
})
```

The screenshot shows the Postman application interface. On the left sidebar, there are sections for 'Collections' (with 'My Collection' expanded), 'Comments', 'History', 'Notes', and 'Files'. The main workspace shows a 'My Collection' folder with a 'Get data' and a 'POST Post data' item. A 'DELETE' request is selected, targeting the URL `http://localhost:3000/students/4`. The 'Body' tab is set to 'JSON' and contains the following JSON payload:

```
1 [  
2 |   {"name": "bunny"}  
3 ]
```

The response status is 200, and the message is 'Student deleted'. To the right of the request details, there is a list of three issues:

1. Wrong HTTP Method: The request is using DELETE, but based on your description (which mentions "This is a POST request, submitting data to an API via the request body"), this should likely be a POST request.
2. Request Body: You have a JSON body `[{"name": "koushik"}]`, which is typical for POST requests, not DELETE requests.
3. 404 Error: The server is returning "Cannot DELETE /students", which suggests the endpoint doesn't support DELETE operations.

A section titled 'What would you like me to fix?' lists three options:

- Should I change this to a POST request instead of DELETE?
- Or do you want to keep it as DELETE but change the endpoint to something like `/students/{id}` to delete a specific student?
- Or is there a different endpoint you should be using?

Below these options, a note says 'Let me know which direction you'd like to go, and I'll update the request for you!'. At the bottom of the interface, there are buttons for 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and 'Help'.

The screenshot shows a browser window with multiple tabs open. The active tab is titled 'localhost:3000/students' and displays the following JSON response:

```
{"id":1,"name":"Ravi"}, {"id":2,"name":"Sita"}, {"id":6,"name":"koushik"}, {"id":5,"name":"koushik"}
```

The browser's address bar shows the URL `localhost:3000/students`. Other tabs visible in the background include 'Expressjs Redclaration Error' and 'Download Postman | Get Started'.