

# RESTFUL OPERATIONS

ABBIDI KRISHNA KOUSHIK

2303A51589

B-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(`
```

## Add Student

```
<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
```

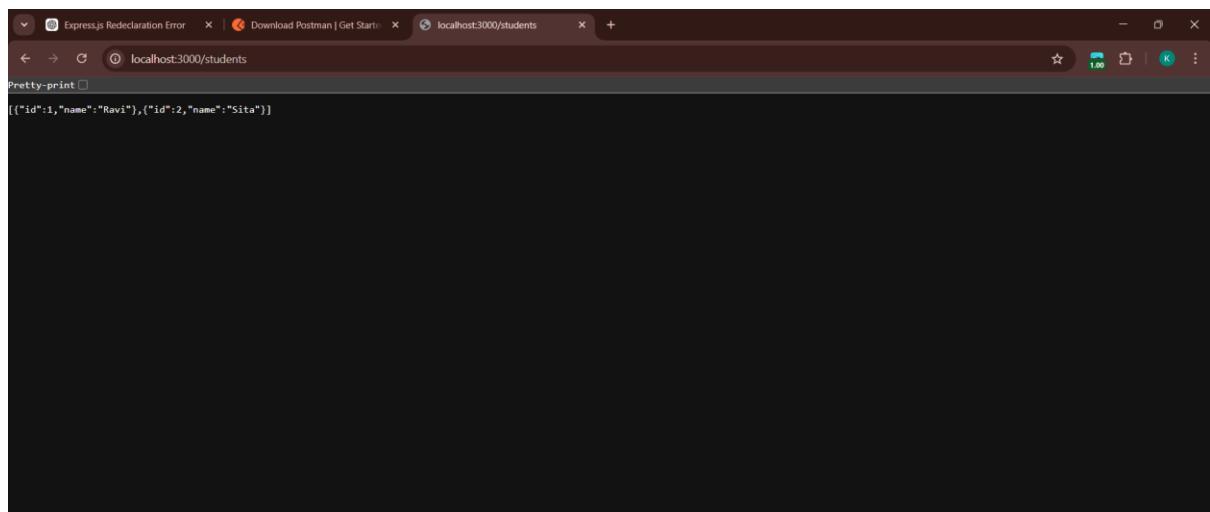
```
app.get('/students', (req, res) => {
  res.json(students);
});
```

#### GET OPERATION:

/ 123 GET - View all students

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

#### OUTPUT:



The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like 'My Collection'. In the main area, a 'GET Get data' request is selected. The URL is 'http://localhost:3000/students'. The request body is set to 'JSON' with the value '{ "name": "bunny" }'. The response status is 200, and the data returned is a table:

	id	name
0	1	Ravi
1	2	Sita

To the right, a sidebar titled 'Fixing the request' lists three items:

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 this, a section asks 'What would you like me to fix?' with several 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?

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

// 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 data' request is defined with the following details:

- Method: GET
- URL: `http://localhost:3000/students`
- Body: JSON (showing the same student data as the browser screenshot)

The response body is displayed as a table:

	id	name
0	1	Ravi
1	2	Sita
2	6	koushik
3	4	koushik
4	5	koushik

The right panel contains a 'Fixing the request' section with the following notes:

- 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.
- Request Body: You have a JSON body `{"name": "koushik"}` which is typical for POST requests, not DELETE requests.
- 404 Error: The server is returning "Cannot DELETE /students", which suggests the endpoint doesn't support DELETE operations.

A message at the bottom says: "What would you like me to fix?".

## / 404 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" });
  }
```

```

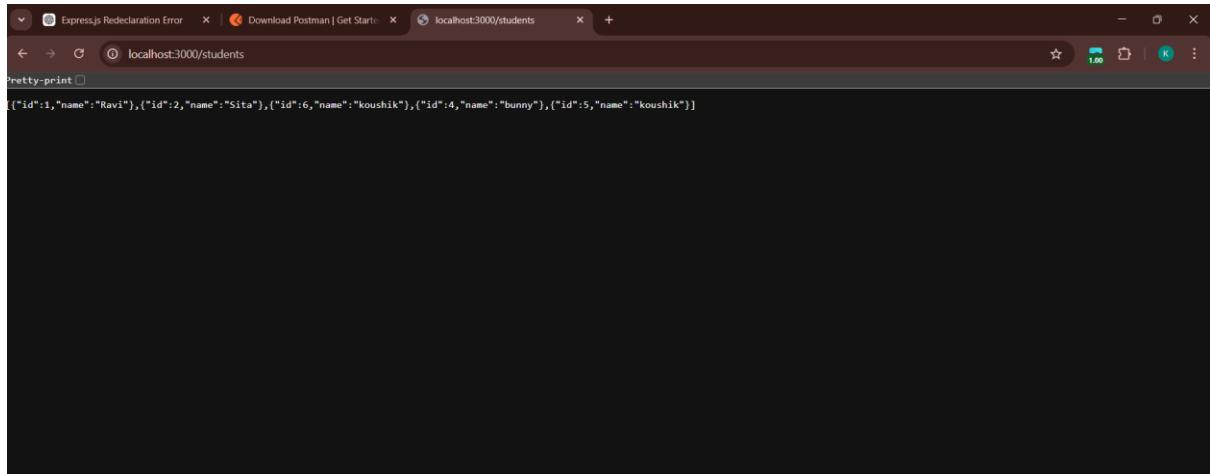
student.name = name;

res.json({ message: "Student updated", student });

});

```

OUTPUT:



```

// 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");
});

```

## OUTPUT:

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

**Koushik Reddy's Workspace**

**My Collection**

- Get data** (HTTP GET)
- Post data** (HTTP POST)

**DELETE** http://localhost:3000/students/4

**Body** (JSON)

```
{
  "name": "koushik"
}
```

**Response** (200)

**message**: Student deleted

**Fixing the request**

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

localhost:3000/students

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

