

| Title: Create a RESTful API server in Express and Node.js. Implementation + Testing application using postman/Thurderclient |
| --- |

**AIM:** Create a RESTful API server in Express and Node.js. Implementation + Testing application using postman/Thurderclient

**Problem Definition:**

**Resources used:**

* **vscode**
* **google**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO 3:** **Test the concepts and components of various front-end, back-end web app development technologies & frameworks using web development tools.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Shelly Powers Learning Node O’ Reilly 2 nd Edition, 2016.

**Pre Lab/ Prior Concepts:**

**Write details about the following content**

* Testing in POSTMAN/Thurderclient

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

const PORT = process.env.PORT || 3000;

app.use(cors());

app.use(bodyParser.json());

// Sample data

let items = [

{ id: 1, name: 'Item 1' },

{ id: 2, name: 'Item 2' },

];

// Get all items

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

res.json(items);

});

// Get an item by ID

app.get('/api/items/:id', (req, res) => {

const item = items.find(i => i.id === parseInt(req.params.id));

if (item) {

res.json(item);

} else {

res.status(404).json({ message: 'Item not found' });

}

});

// Create a new item

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

const newItem = {

id: items.length + 1,

name: req.body.name

};

items.push(newItem);

res.status(201).json(newItem);

});

// Update an item

app.put('/api/items/:id', (req, res) => {

const item = items.find(i => i.id === parseInt(req.params.id));

if (item) {

item.name = req.body.name;

res.json(item);

} else {

res.status(404).json({ message: 'Item not found' });

}

});

// Delete an item

app.delete('/api/items/:id', (req, res) => {

const index = items.findIndex(i => i.id === parseInt(req.params.id));

if (index !== -1) {

items.splice(index, 1);

res.status(204).send();

} else {

res.status(404).json({ message: 'Item not found' });

}

});

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});

**Steps for execution:**

1. **Initialize the Project:**

bash

Copy code

mkdir express-api

cd express-api

npm init -y

1. **Install Dependencies:**

bash

Copy code

npm install express body-parser cors

1. **Create the Project Structure:**

bash

Copy code

mkdir src

touch src/index.js

**Step 2: Implementing the API**

1. **Edit src/index.js:**

javascript

Copy code

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

const PORT = process.env.PORT || 3000;

app.use(cors());

app.use(bodyParser.json());

// Sample data

let items = [

{ id: 1, name: 'Item 1' },

{ id: 2, name: 'Item 2' },

];

// Get all items

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

res.json(items);

});

// Get an item by ID

app.get('/api/items/:id', (req, res) => {

const item = items.find(i => i.id === parseInt(req.params.id));

if (item) {

res.json(item);

} else {

res.status(404).json({ message: 'Item not found' });

}

});

// Create a new item

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

const newItem = {

id: items.length + 1,

name: req.body.name

};

items.push(newItem);

res.status(201).json(newItem);

});

// Update an item

app.put('/api/items/:id', (req, res) => {

const item = items.find(i => i.id === parseInt(req.params.id));

if (item) {

item.name = req.body.name;

res.json(item);

} else {

res.status(404).json({ message: 'Item not found' });

}

});

// Delete an item

app.delete('/api/items/:id', (req, res) => {

const index = items.findIndex(i => i.id === parseInt(req.params.id));

if (index !== -1) {

items.splice(index, 1);

res.status(204).send();

} else {

res.status(404).json({ message: 'Item not found' });

}

});

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});

**Step 3: Running the Server**

1. **Start the Server:**

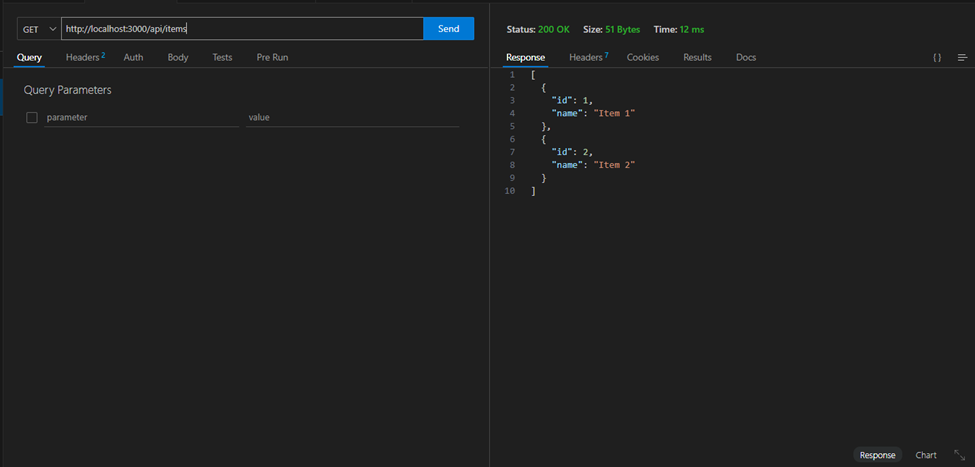
bash

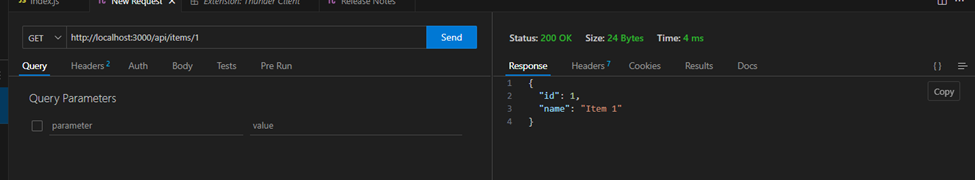
Copy code

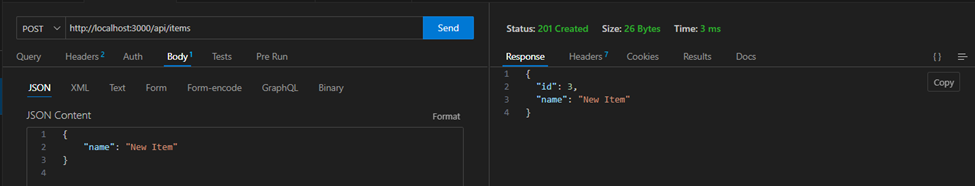
node src/index.js

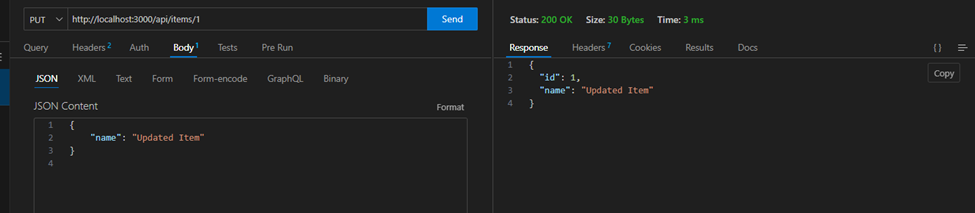
1. **Check the Server:** Open your browser and go to http://localhost:3000/api/items. You should see the initial array of items.

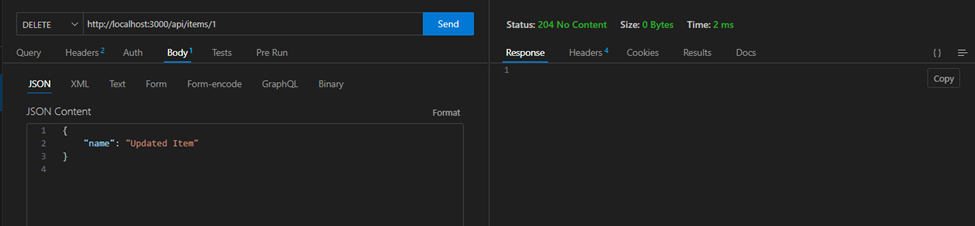
Output:











**Conclusion:**

**Learned to make a restful api and tested it using thunderclient.**