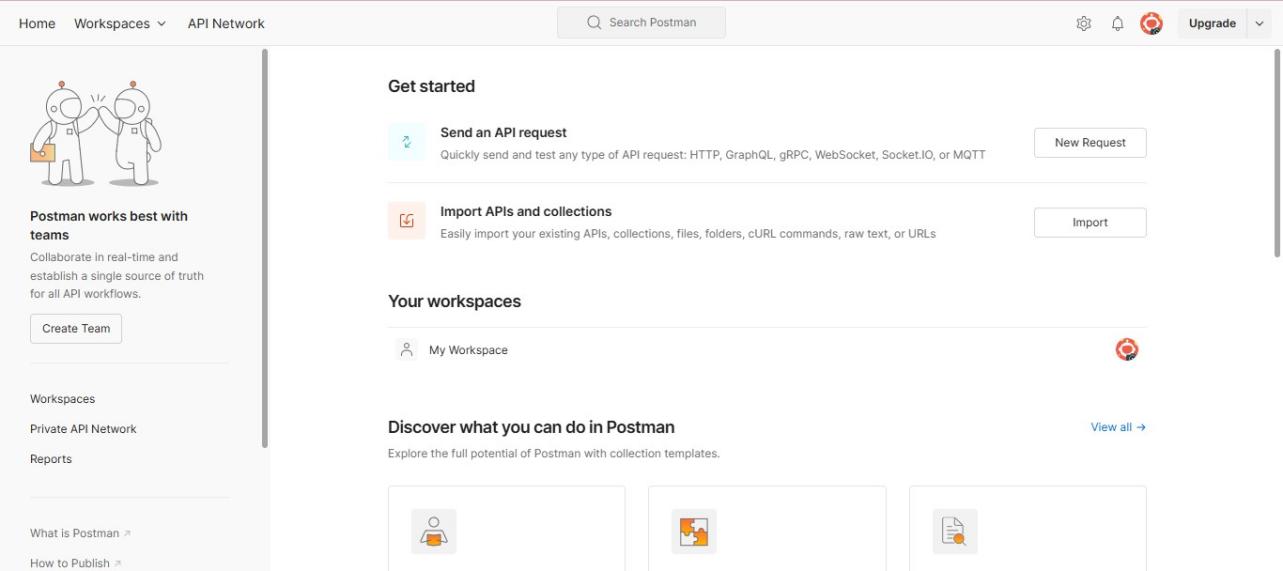
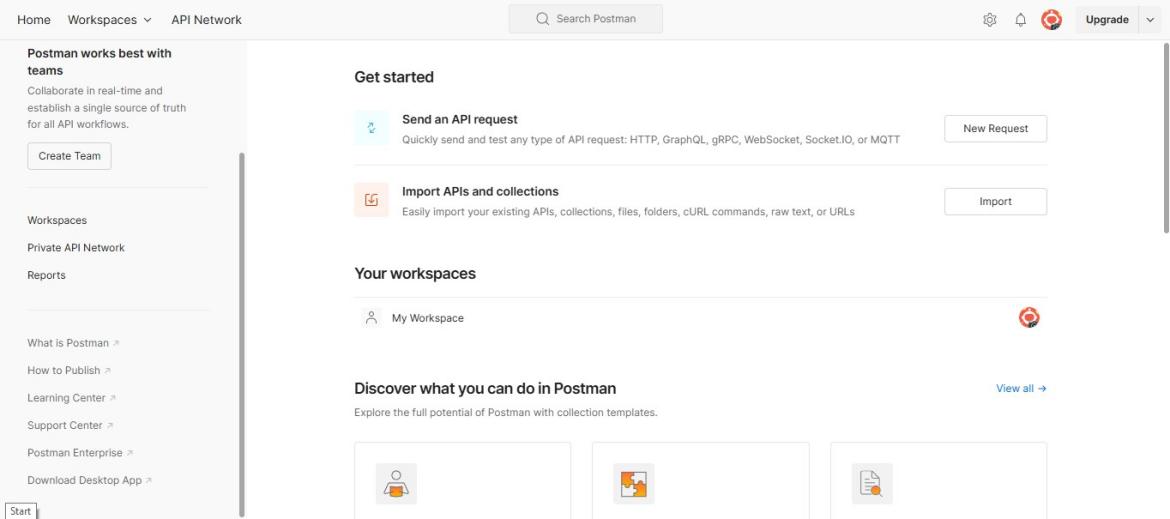
POSTMAN

# Installation:

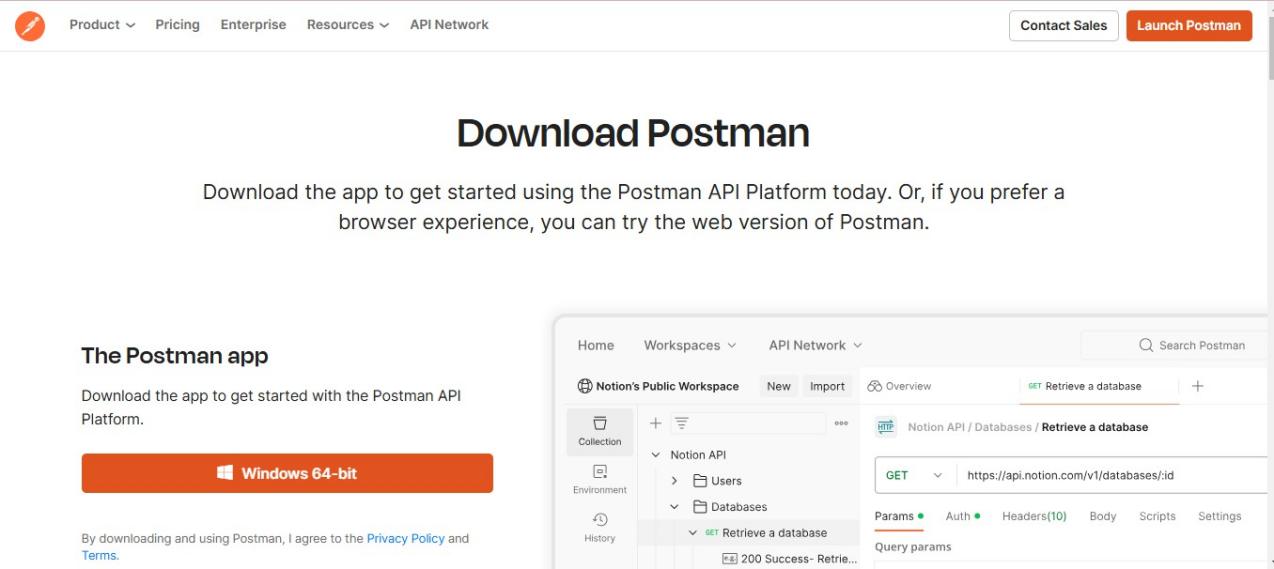
* Open browser and Search: <https://www.postman.com/>



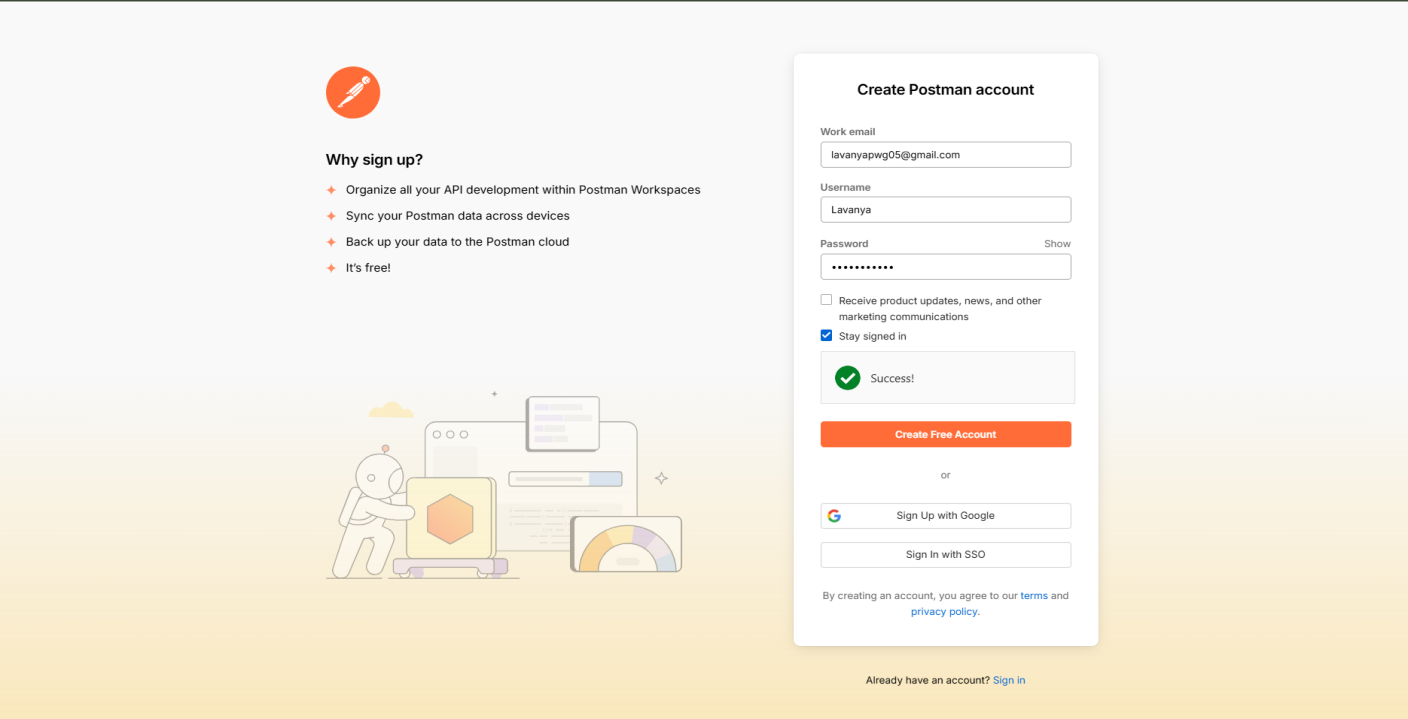
* In this page, in sidebar there us an option Download Desktop App.



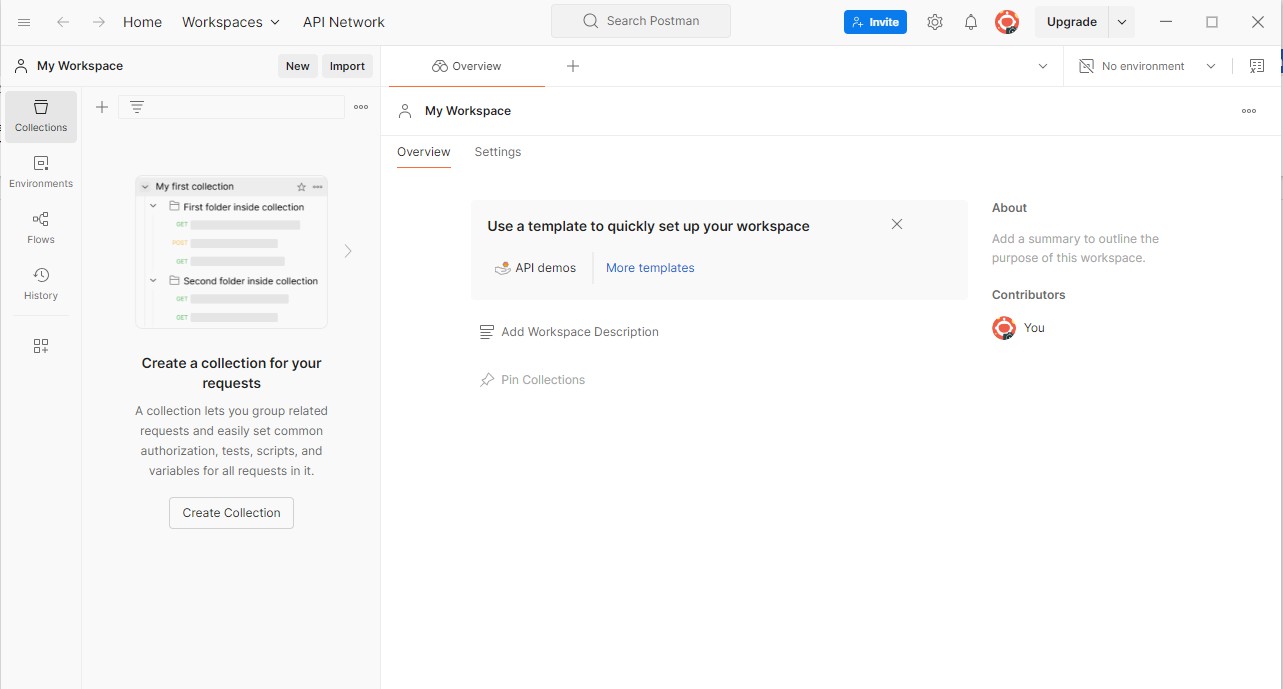
* Click on it to download the Postman App.
* It is Navigated to the following page. Choose the appropriate OS and click on it to download.



* Now open the postman App in system.
* Create Account in Postman using Google.

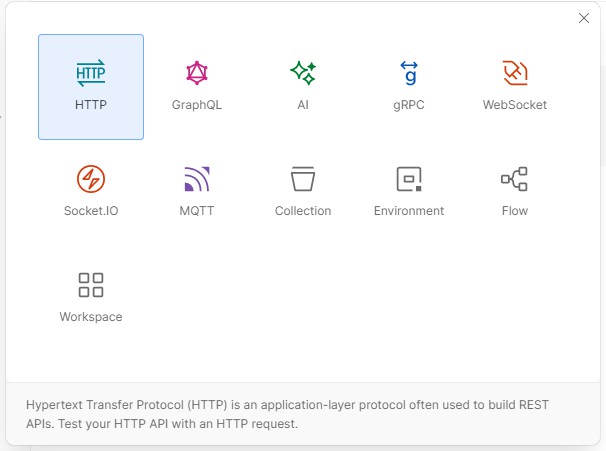


* ​

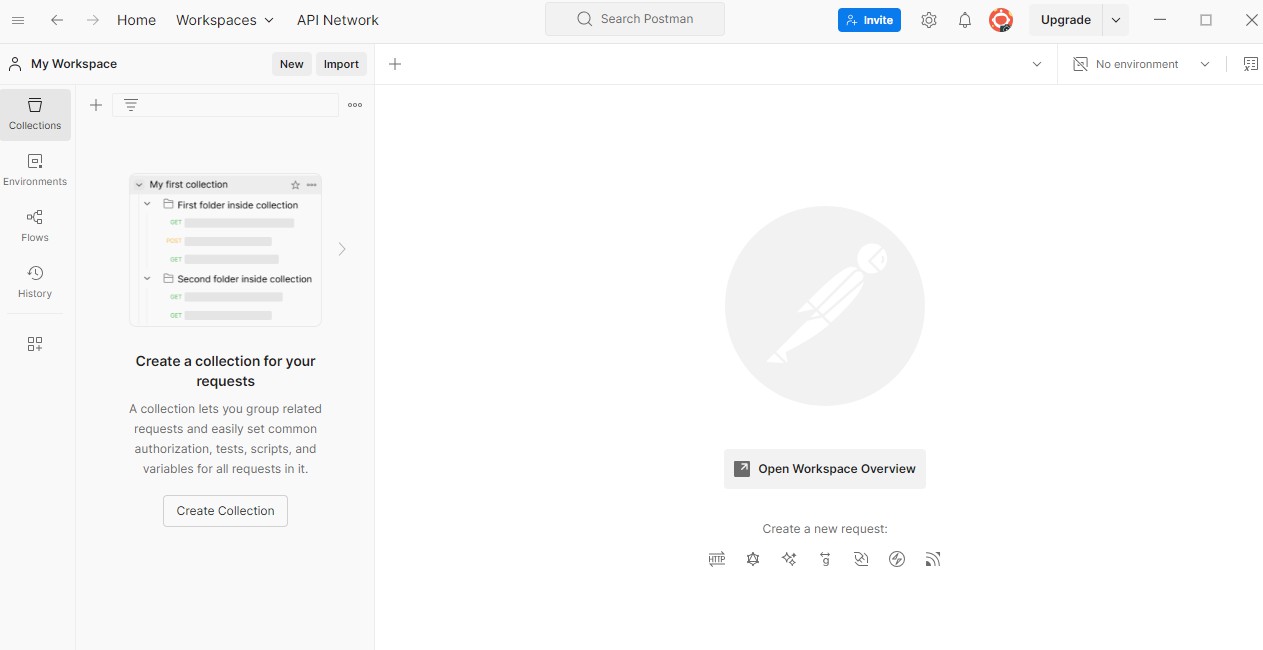


Click on New tab.

* Choose HTTP in the List



* Click on Create Collection



* Now give a name to Collection



* Open VS code and write the server code and start the server.

# Server.js:

const http = require('http'); const fs = require('fs');

const path = './employees.json'; function readEmployeeFile() { if (!fs.existsSync(path)) {

fs.writeFileSync(path, JSON.stringify([]));

}

const data = fs.readFileSync(path);

return JSON.parse(data);

}

function writeEmployeeFile(data) { fs.writeFileSync(path, JSON.stringify(data, null, 2));

}

const server = http.createServer((req, res) => { res.setHeader('Content-Type', 'application/json');

if (req.url === '/employees' && req.method === 'GET') { const employees = readEmployeeFile(); res.end(JSON.stringify(employees));

} else if (req.url === '/employees' && req.method === 'POST') { let body = '';

req.on('data', chunk => { body += chunk.toString();

});

req.on('end', () => {

const newEmployee = JSON.parse(body); const employees = readEmployeeFile(); employees.push(newEmployee); writeEmployeeFile(employees); res.statusCode = 201;

res.end(JSON.stringify({ message: 'Employee added successfully' }));

});

} else if (req.url.startsWith('/employees/') && req.method === 'PUT') { const id = parseInt(req.url.split('/')[2]);

let body = ''; req.on('data', chunk => {

body += chunk.toString();

});

req.on('end', () => {

const updatedEmployee = JSON.parse(body); const employees = readEmployeeFile();

const index = employees.findIndex(employee => employee.id === id);

if (index !== -1) {

employees[index] = { ...employees[index], ...updatedEmployee }; writeEmployeeFile(employees);

res.end(JSON.stringify({ message: 'Employee updated successfully' }));

} else {

res.statusCode = 404;

res.end(JSON.stringify({ message: 'Employee not found' }));

}

});

} else if (req.url.startsWith('/employees/') && req.method === 'PATCH') { const id = parseInt(req.url.split('/')[2]);

let body = ''; req.on('data', chunk => {

body += chunk.toString();

});

req.on('end', () => {

const partialUpdate = JSON.parse(body); const employees = readEmployeeFile();

const index = employees.findIndex(employee => employee.id === id); if (index !== -1) {

employees[index] = { ...employees[index], ...partialUpdate }; writeEmployeeFile(employees);

res.end(JSON.stringify({ message: 'Employee data updated successfully' }));

} else {

res.statusCode = 404;

res.end(JSON.stringify({ message: 'Employee not found' }));

}

});

} else if (req.url.startsWith('/employees/') && req.method === 'DELETE') { const id = parseInt(req.url.split('/')[2]);

const employees = readEmployeeFile();

const filteredEmployees = employees.filter(employee => employee.id !== id); if (employees.length !== filteredEmployees.length) {

writeEmployeeFile(filteredEmployees);

res.end(JSON.stringify({ message: 'Employee deleted successfully' }));

} else {

res.statusCode = 404;

res.end(JSON.stringify({ message: 'Employee not found' }));

}

} else if (req.url.startsWith('/employees/') && req.method === 'HEAD') { const id = parseInt(req.url.split('/')[2]);

const employees = readEmployeeFile();

const employeeExists = employees.some(employee => employee.id === id); if (employeeExists) {

res.statusCode = 200; res.end();

} else {

res.statusCode = 404; res.end();

}

} else if (req.method === 'OPTIONS') {

res.setHeader('Allow', 'GET, POST, PUT, PATCH, DELETE, OPTIONS');

res.end();

} else {

res.statusCode = 404;

res.end(JSON.stringify({ message: 'Route not found' }));

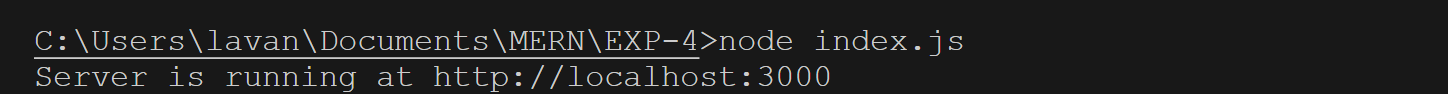
}

});

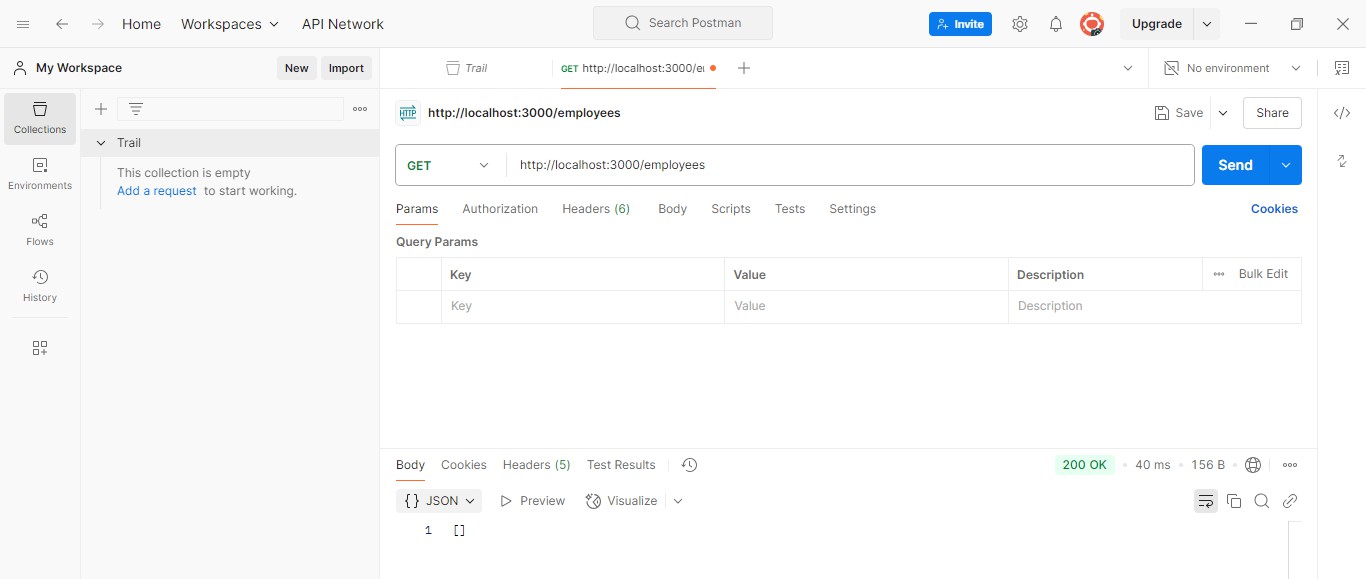
const PORT = 3000; server.listen(PORT, () => {

console.log(`Server running at [http://localhost:${](http://localhost/)PORT}/`);

});

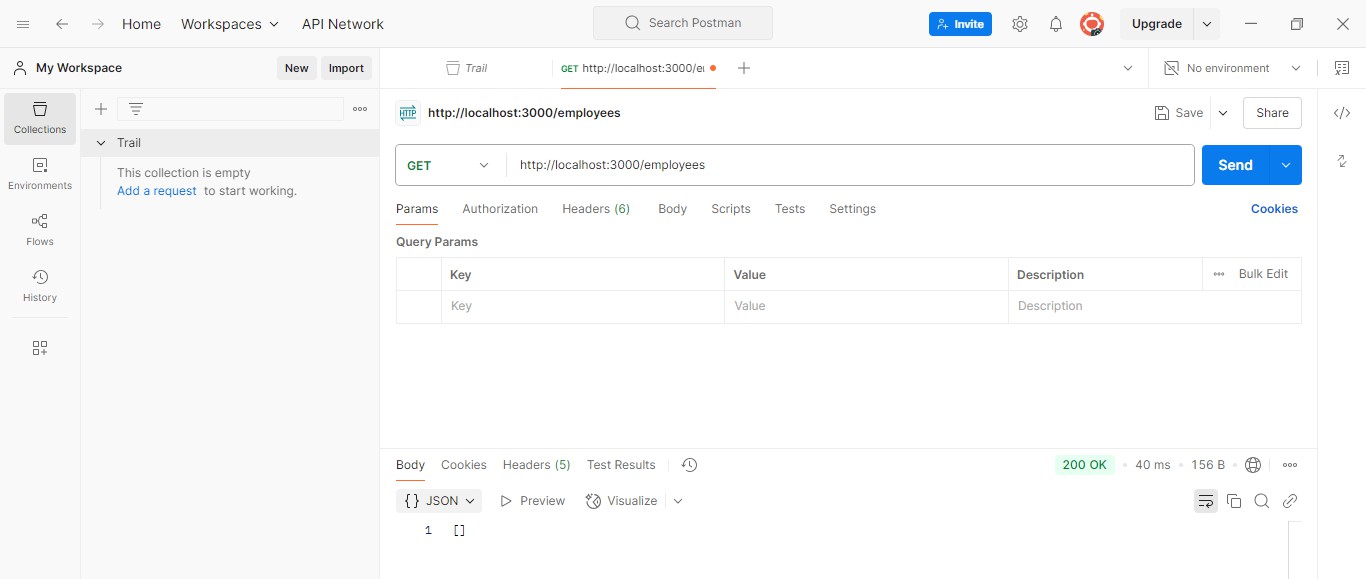


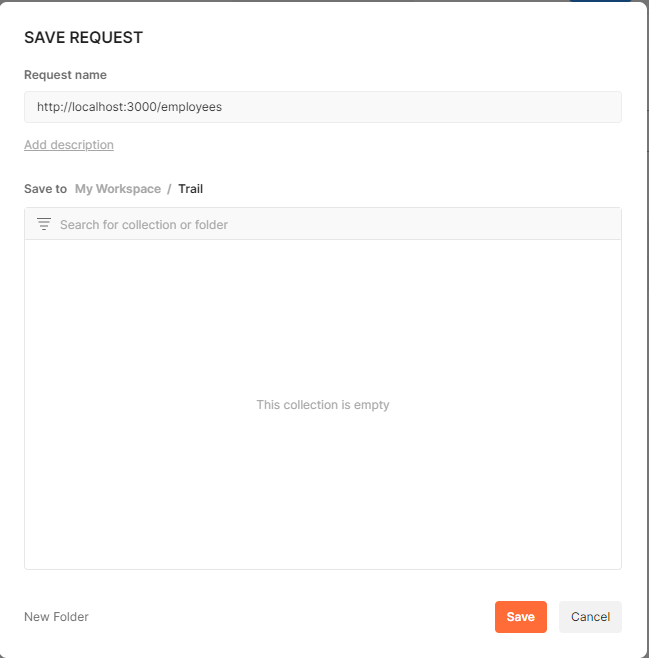
Now open the Postman and give URL and select the request type and send.



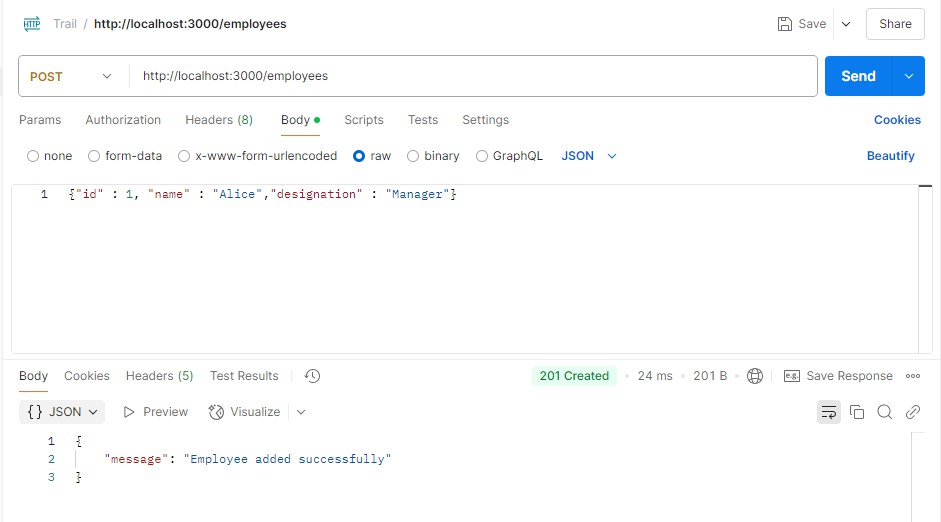
GET request

* To save the request in Collection click on Save button



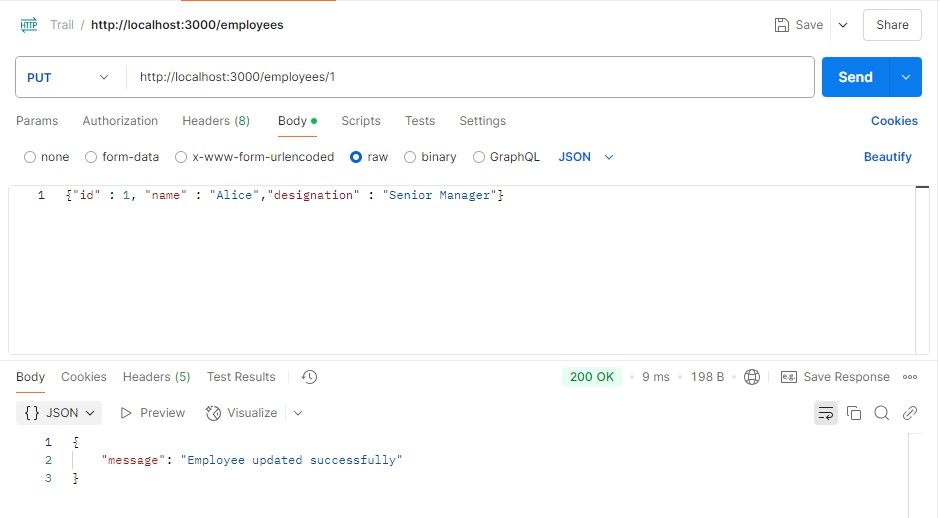


* Click on the Collect and click Save.

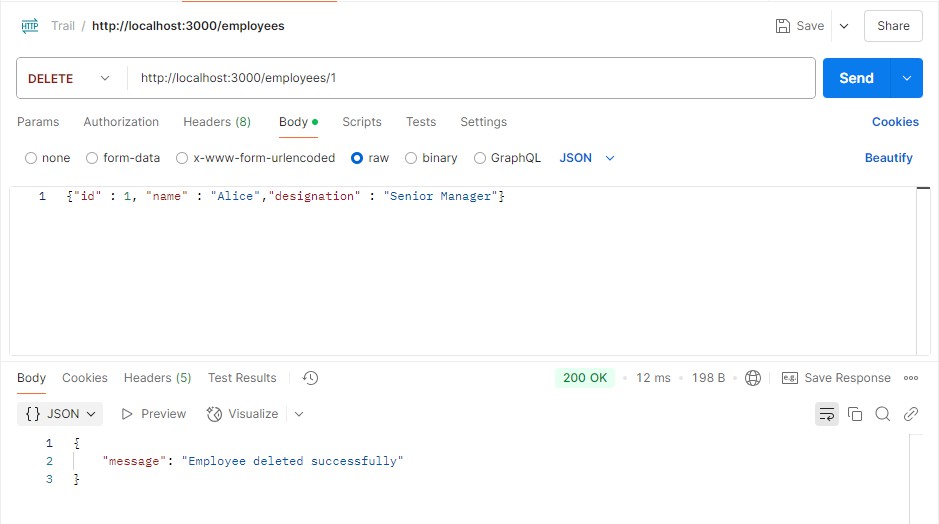


POST request

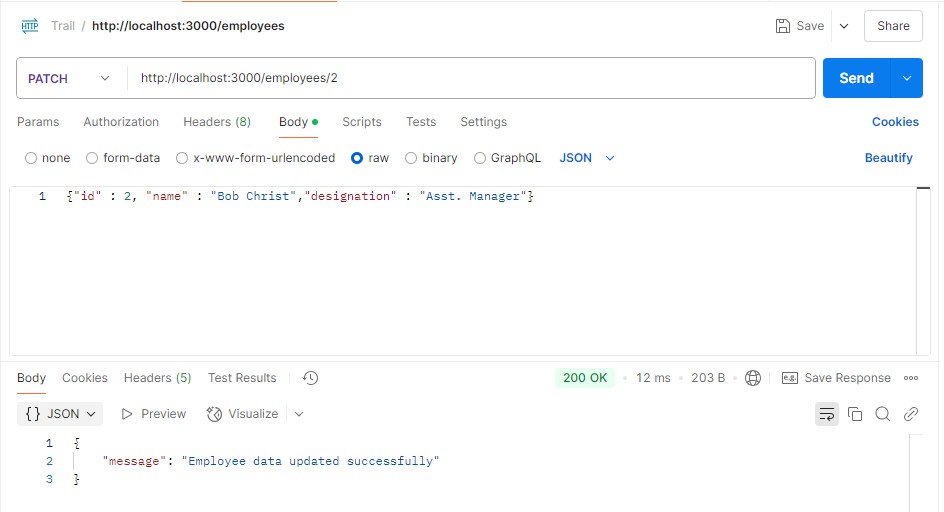
* Save it in the same collection.



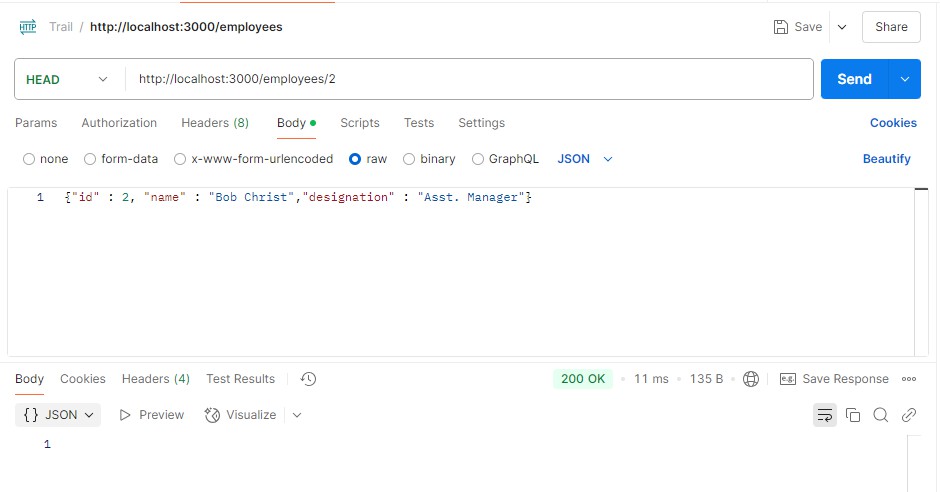
PUT request



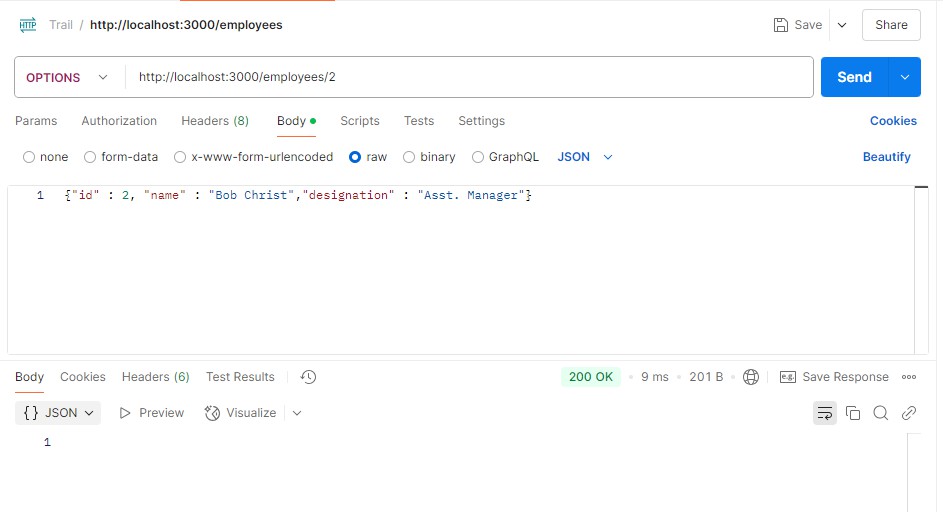
DELETE request



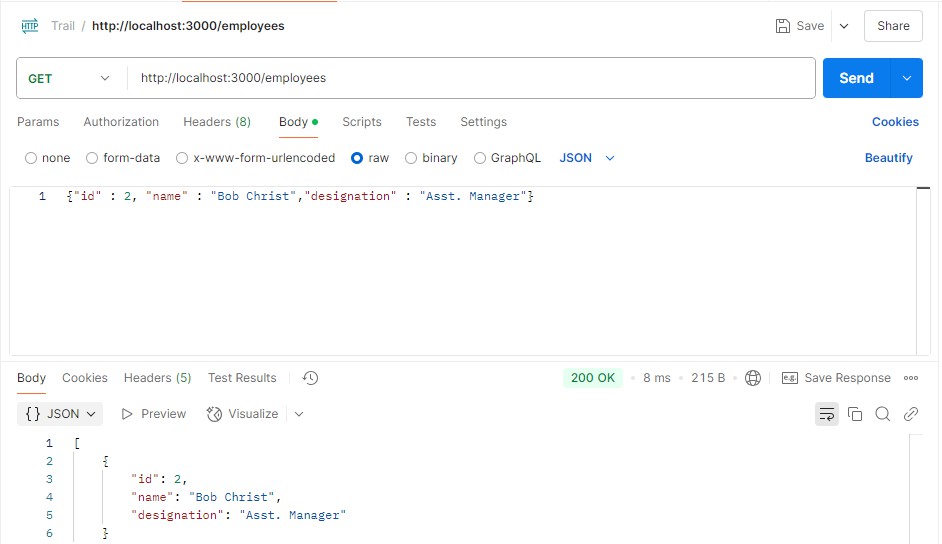
PATCH request



HEAD request

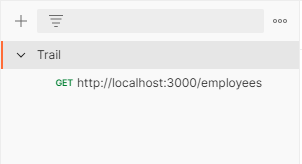


OPTIONS request



GET request

* Now Click on Collection and open the collection



* Click on View Complete documentation.

