Week10

10.Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)

Aim:

☛ Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)

Solution :

Firstly we need to create a new folder and open the folder in the command prompt and enter a command as below:

npminit -y

Open that folder in the vscode by entering code.

Next in the terminal we need to install all the packages we need, so we mainly use express and sqlite3.

The Command to install express and sqlite3 is

npm install express sqlite3

Then create file named as the app.js and db.js

**db.js**

const sqlite3 = require('sqlite3').verbose();

// Function to initialize the database schema

function initializeDatabase() {

const db = new sqlite3.Database('./mydatabase.db', (err) => {

if (err) {

console.error(err.message);

} else {

console.log('Connected to the SQLite database.');

createStudentsTable(db);

}

});

// Close the database connection when the Node process exits

process.on('exit', () => {

db.close((err) => {

if (err) {

console.error(err.message);

} else {

console.log('Disconnected from the SQLite database.');

}

});

});

}

// Function to create the 'students' table if it doesn't exist

function createStudentsTable(db) {

const createTableQuery = `

CREATE TABLE IF NOT EXISTS students (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT,

age INTEGER,

grade TEXT

);

`;

db.run(createTableQuery, (err) => {

if (err) {

console.error(err.message);

} else {

console.log('The students table has been created or already exists.');

}

});

}

module.exports = { initializeDatabase };

app.js

const express = require('express');

const sqlite3 = require('sqlite3');

const{ initializeDatabase } = require('./db');

const app = express();

const port = 3000;

// Connect to SQLite database

const db = new sqlite3.Database('./mydatabase.db', (err) => {

if (err) {

console.log(err.message);

} else {

console.log('Connected to the SQLite database.');

}

});

// Middleware to parse request body as JSON

app.use(express.json());

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

res.send('Welcome to the Student');

});

// Get all Students

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

db.all('SELECT \* FROM students', [], (err, rows) => {

if (err) {

return console.error(err.message);

}

res.json(rows);

});

});

// Get a single student by id

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

const id = req.params.id;

db.all('SELECT \* FROM students WHERE id = ?', [id], (err, row) => {

if (err) {

return console.error(err.message);

}

res.json(row);

});

});

// Create a new student

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

const{ name, age, grade } = req.body;

db.run('INSERT INTO students (name, age, grade) VALUES (?, ?, ?)', [name, age, grade], function (err) {

if (err) {

return console.error(err.message);

}

res.status(201).json({ id:this.lastID });

});

});

// Update a student

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

const id = req.params.id;

const{ name, age, grade } = req.body;

db.run('UPDATE students SET name = ?, age = ?, grade = ? WHERE id = ?', [name, age, grade, id], function (err) {

if (err) {

return console.error(err.message);

}

res.json({ updatedID:id });

});

});

// Delete a student

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

const id = req.params.id;

db.run('DELETE FROM students WHERE id = ?', id, function (err) {

if (err) {

return console.error(err.message);

}

res.json({ deletedID:id });

});

});

app.listen(port, () => {

console.log('Server running at http://localhost:${port}');

});

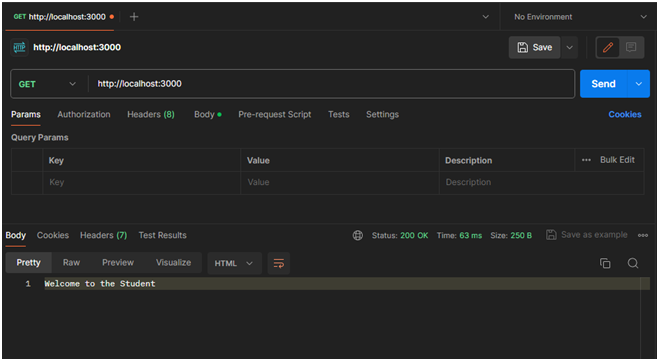
Output :

GET:

Open Postman.

Set the request type to GET.

Enter the URL: <http://localhost:3000/students>.



POST : Create a New Student

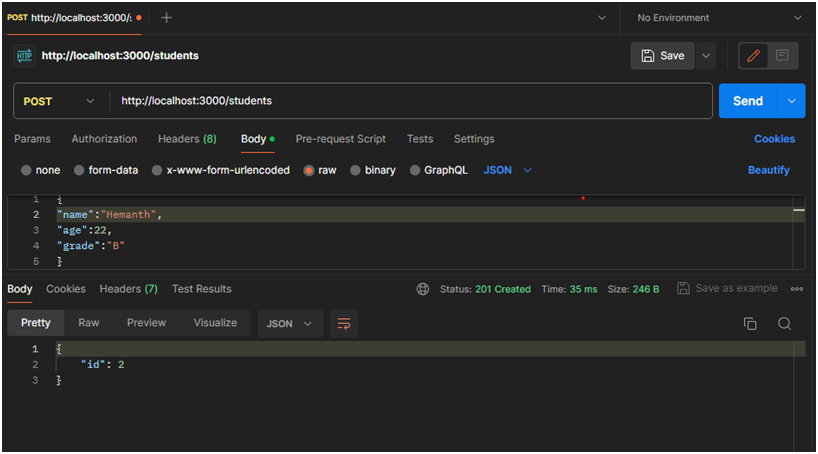
Open Postman.

Set the request type to POST.

Enter the URL: http://localhost:3000/students.

Go to the "Body" tab.

Select raw and set the body to JSON format.



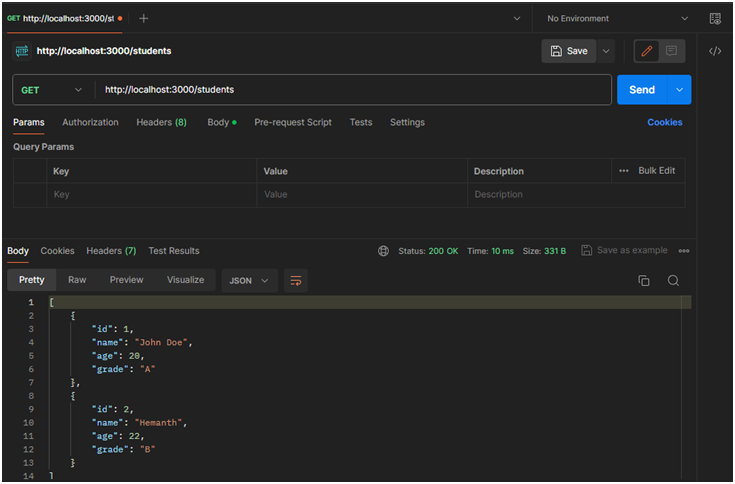
GET: #all Students

Set the request type to GET.

Enter the URL: http://localhost:3000/students.

Click on the "Send" button

You should receive a response with details of all students in your SQLite database.



DELETE:

Set the request type to DELETE.

Enter the URL for the student you want to delete (replace: id with an actual student ID): http://localhost:3000/students/:id

Place instead of ID which replace with number that is ID to be deleted.

Then click send

