

Building a RESTful API in Node, Express & Postgres

Test API with Supertest

Requirements

- Node and npm

Installation

- Clone the git repo: <https://github.com/Shabanpk/NodeSensorAPI.git>
- Install dependencies: `npm install`
- Create the database in Postgres
- Execute the notepad file content "CreateSensorTable.txt" in PostgreSQL editor to create table "sensorinfo"
- Run the command in terminal "node CsvToInsertDataInDb.js" that can read all CSV file and insert record in database or second option import Senors_TEMP_099 CSV file in Postgres database
- Set credential of your database name, password in "db.js" file
- Run command in node terminal to test API "Node app.js"

Testing the API Using Postman

Login API -Module

When you run the application with command "Node app.js" then automatically user table create in the database with the username "Shaban" and password "admin". In addition, I have also used the "bcrypt" and "jsonwebtoken" library to encrypt the password in a token.

1. Send the post request the given url is mentioned below:
<http://localhost:3001/user/auth>

Expected Output

```
{
  "result": {
    "id": 1,
    "name": "Shaban",
    "email": "shaban@admin.com",
    "createdAt": "2020-06-14T18:00:47.843Z",
    "updatedAt": "2020-06-14T18:00:47.843Z",
    "token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwibmFtZSI6IiNoYWJhbiIsImVtYWlsIjoic2hhYmFuQGFKbWluLmNvbSIsImNyZWVhbnR5b2ZWRBdCI6IjIwMjAtMDYtMTRUMTg6MDA6NDcuODQzWilsInVwZGF0ZWRBdCI6IjIwMjAtMDYtMTRUMTg6MDA6NDcuODQzWilsImIhdCI6MTU5MjlyNTU5NH0.gPosGX3XbaFnkLJEYOSuRXdxVN-urUyds7CHTPzeDLY"
  }
}
```

2. Then you can get token and get request the against the token pass in the authorization section

<http://localhost:3001/user/all>

Expected Output

```
{
  "users": [
    {
      "id": 1,
      "name": "Shaban",
      "email": "shaban@admin.com",
      "createdAt": "2020-06-14T18:00:47.843Z",
      "updatedAt": "2020-06-14T18:00:47.843Z"
    }
  ]
}
```

if the token is wrong then the error message "Failed to authorize request"

Sensor API – CRUD

I have created the CRUD API of sensor information which is mentioned below all four API endpoints.

Endpoint

Get all Sensor data from database:

GET Request: <http://localhost:3001/sensor/getAllSensors>

Get Sensor data by specific ID from database:

GET Request: <http://localhost:3001/sensor/getSensorById/1>

Insert Sensor data in database:

POST Request: <http://localhost:3001/sensor/createSensor>

Update Sensor data by specific ID:

PUT Request: <http://localhost:3001/sensor/updateSensor/1>

Delete Sensor data by specific ID:

Delete Request: <http://localhost:3000/sensor/deleteSensor/5>

Testing the endpoint with supertest

- I have created the endpoint in “JEST_API_TESTING.js” which can get only one row fetch from database and call this endpoint in test folder inside index.js.
- Just run the command in the terminal: npm test

Output Endpoint

