UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BENR 2221

COMPUTER ENGINEERING LABORATORY 2

LAB SESSION 1

Introduction to Restful Server

Prepared by:
Computer Engineering Department, FKEKK
2023

1.0 OBJECTIVES

1. To develop simple restful server to handle the basic http methods
2. To understand the network packet between the client and server for http request and response

2.0 EQUIPMENT

1. Personal computer running modern operating system with minimum of 8GB RAM

3.0 SAFETY

1. Ensure hands are dry when using the equipment or touching the switches.
2. Do not use plugs with cracks, signs of overheating (e.g. discoloration, charring or deformation) or loosen parts.
3. Stop using the equipment if abnormal operating conditions occur, such as being too noisy or having signs of overheating.
4. Keep all equipment in an organized manner, where they are easy to reach without bending or stretching.

4.0 THEORY

RESTful API stands for Representational State Transfer API. It is an architectural style for designing web services that uses HTTP methods to perform CRUD (Create, Read, Update, and Delete) operations on resources. RESTful API allows the client to interact with the server in a stateless manner, meaning each request contains all the necessary information for the server to process it without relying on any context from previous requests. RESTful API is widely used in modern web development as it provides a flexible and scalable way to expose data and functionality over the internet.

HTTP methods are an integral part of the RESTful API architecture. There are four primary HTTP methods used in RESTful API design: GET, POST, PATCH, and DELETE.

The **GET** method is used to retrieve data from the server. The server sends a response with the requested data back to the client.

The **POST** method is used to create new resources on the server. The client sends a request with the necessary data to create the resource, and the server responds with the newly created resource.

The **PATCH** method is used to update existing resources on the server. The client sends a request with the updated data to the server, and the server responds with the updated resource.

The **DELETE** method is used to remove existing resources from the server. The client sends a request to delete the resource, and the server responds with a confirmation that the resource has been deleted.

In summary, HTTP methods define the type of action that the client wants to perform on the resource, and RESTful API design relies heavily on these methods to perform CRUD operations on resources.

5.0 PROCEDURES

PART A: Basic Nodejs Server and Express Framework

1. Create a project directory: Create a new directory for your project using the following command:
   mkdir my-restful-server

2. Initialize the project: Navigate to the project directory and initialize the project using the following command:
   npm init -y

3. Install Express: Install Express framework using the following command:
   npm install express

4. Create a server file: Create a new file named **server.js** in the project directory and add the following code:

JS lab01.js U ●

JS lab01.js > ...

```
1   const express = require('express');
2   const app = express();
3   const port = 3000;
4
5   app.get('/', (req, res) => {
6       res.send('Hello World!');
7   });
8
9   app.listen(port, () => {
10      console.log(`Server listening at http://localhost:${port}`);
11  });
12
```

   This code imports the Express module, creates an instance of the app, sets the port to 3000, and creates a basic endpoint that responds with "Hello World!".

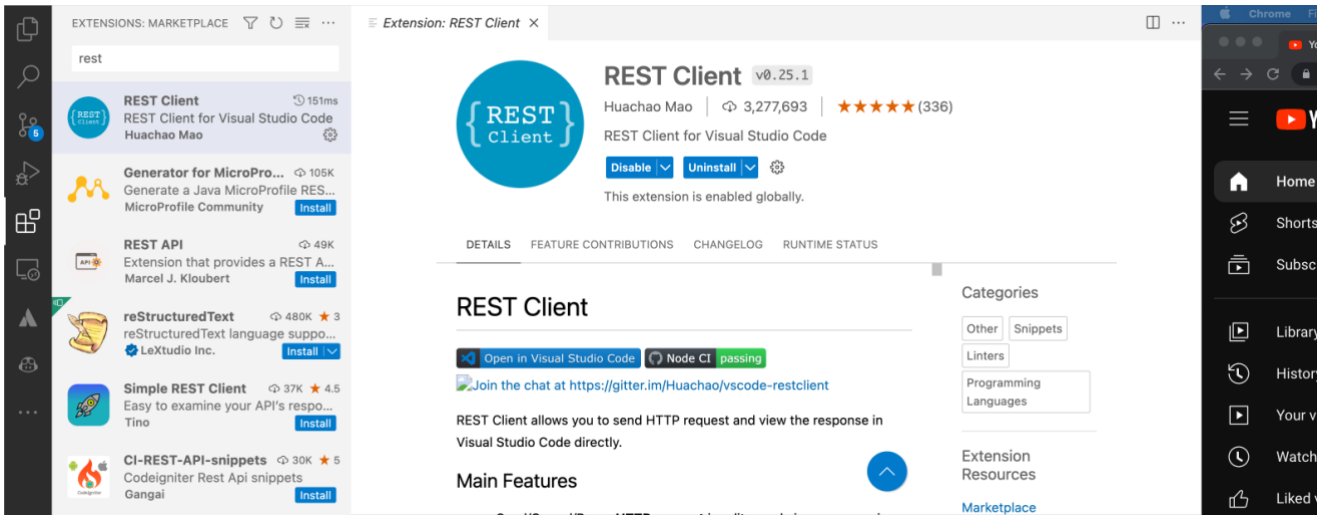
5. Start the server: Start the server by running the following command:
   node server.js

   You should see the message "Server listening at http://localhost:3000" in the console.

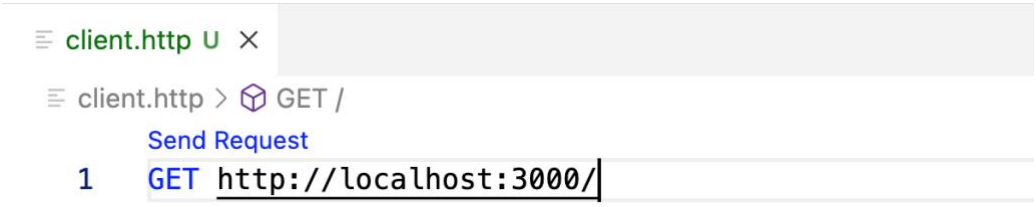
PART B: Testing Restful API

1. Install the Rest Client Extension from the Microsoft Visual Studio Code

https://marketplace.visualstudio.com/items?itemName=humao.rest-client



2. Create a test file called **client.http** into your project folder. Add the HTTP GET request as follow.



3. Then, click the **Send Request** to trigger a request to the server and monitor the response received.

PART C: Parsing HTTP Request Body

1. Add the following line into your server.js to enable the express framework to parse the body attached with the HTTP request

   app.use(express.json());

2. Then add a HTTP Post route into your server.js

JS lab01.js U ×

JS lab01.js > ...

```
1   const express = require('express');
2   const app = express();
3   const port = 3000;
4
5   app.use(express.json());
6
7   app.get('/', (req, res) => {
8       res.send('Hello World!');
9   });
10
11  app.post('/', (req, res) => {
12     res.send(req.body)
13  });
14
15  app.listen(port, () => {
16      console.log(`Server listening at http://localhost:${port}`);
17  });
18
```

3. Start the server: Start the server by running the following command:
   node server.js

4. Edit the **client.http** to add HTTP POST request as follow.

```
Send Request
5   POST http://localhost:3000/
6   Content-Type: application/json
7
8   {
9     "name": "John Doe",
10    "email": "john@m.com"
11  }
```

Discussion

Assuming the list of users will be stored using a normal array as the database. Complete the server to allow a client to register and login to through the Restful API.

Discuss the necessary to encrypt the user password before its store into the database rather than keeping its raw value. Then improve the Restful API using the bcrypt library in NodeJs.

https://www.npmjs.com/package/bcrypt