

CS3219 OTOTTaskB

Implement REST API for backend and frontend using JavaScript

Name: Kwan Chen Sheng

Matric No.: A0185264Y

[OTOTTaskB Repo](#)

Task B1: Running and Accessing API Locally

Required Application

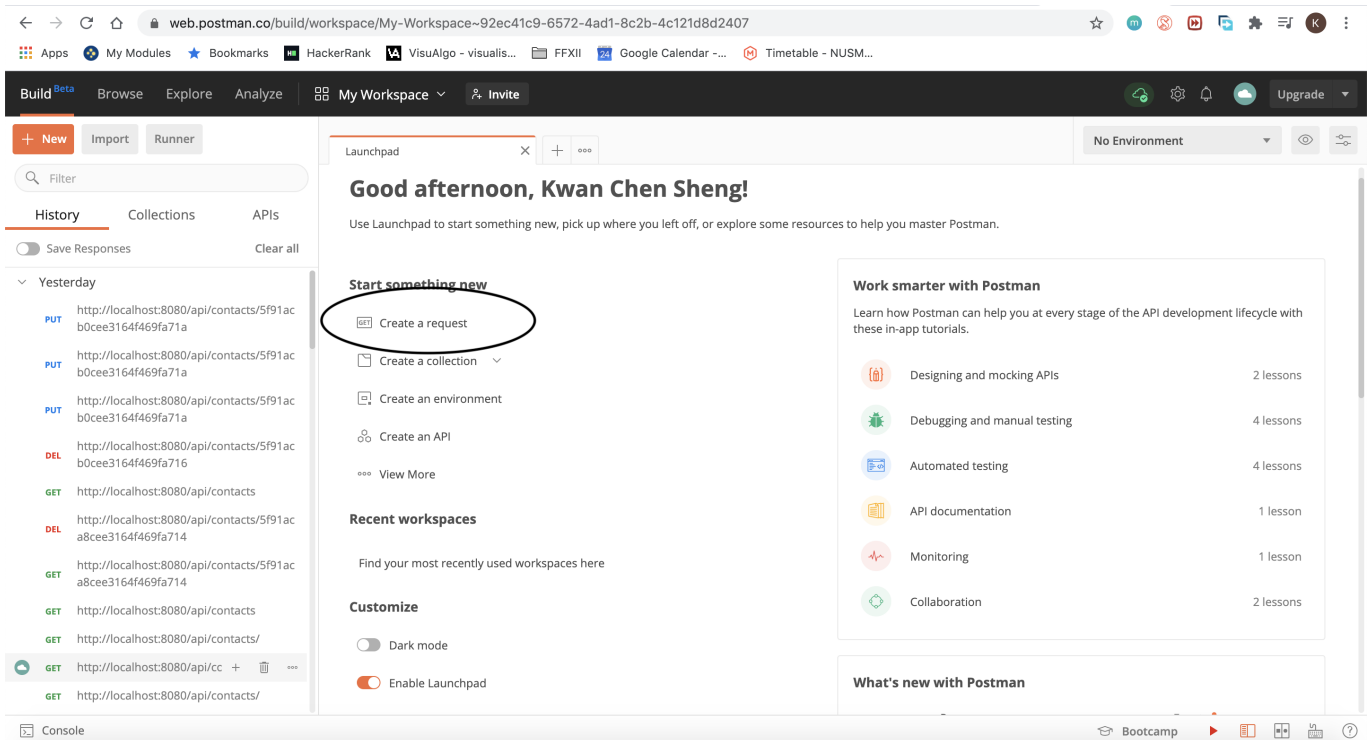
- NodeJS
- MongoDB
- Express
- NPM
- Postman

To install Express, use this command in the terminal

```
npm install express --save
```

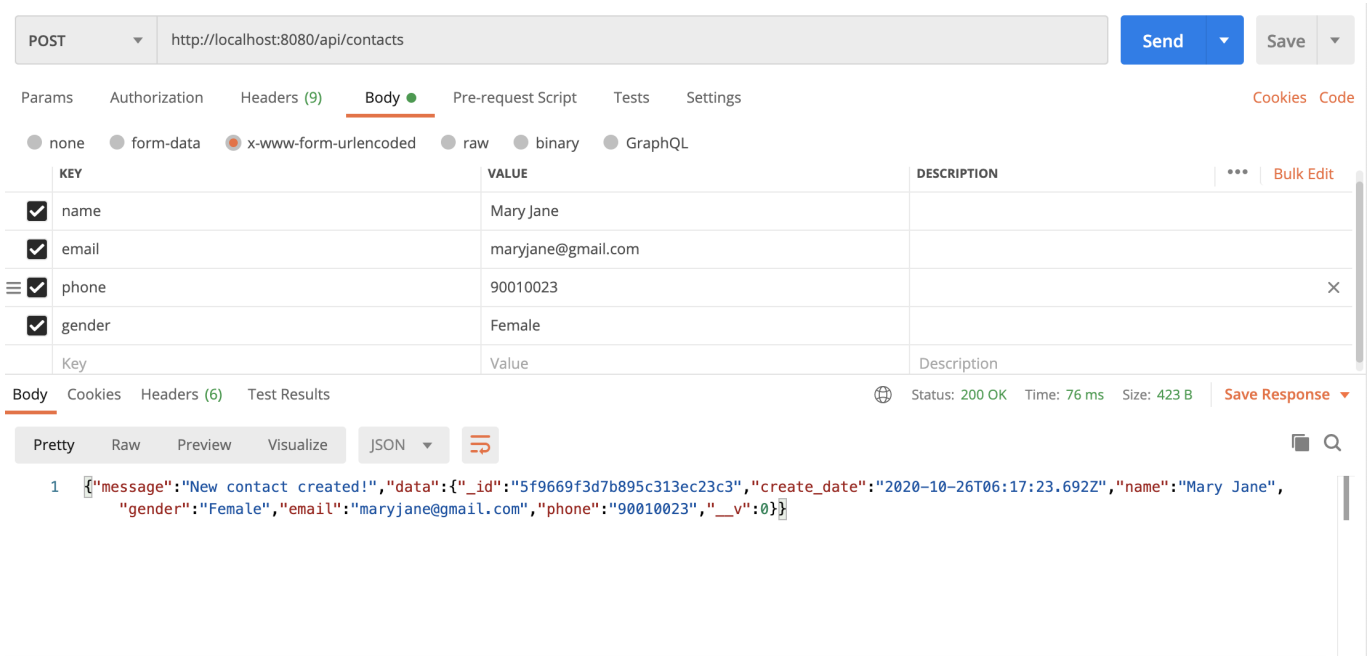
Here are the steps to follow:

1. Clone or Download the repo.
2. Use the command line `npm install` to install the necessary components.
3. In a new terminal, run `mongod` to ensure that the terminal is connected to mongoDB.
4. Upon ensuring terminal is connected to mongoDB, run `nodemon index` to start the app running in local server.
5. Access `http://localhost:8080`. If **Connected!** is shown, localhost is successfully deployed.
6. Go to `http://localhost:8080/api/contacts` to retrieve the contacts, if any.
7. For **GET**, **POST**, **UPDATE**, **DELETE**, we are using POSTMAN to call these APIs.
8. Click on create a request to begin executing API.



For **POST**:

1. Click on the body tab and select the **x-www-form-urlencoded**.
2. Input the following keys: **name**, **email**, **phone** and **gender** and fill in values with it.
3. Enter **http://localhost:8080/api/contacts** for the URL.
4. Once it is filled, press send and it will display **New contact created!**.



For **GET**:

1. Click on the body tab and select the **x-www-form-urlencoded**
2. Enter **http://localhost:8080/api/contacts** for the URL.
3. Press send and it will display the list of contacts.

GET <http://localhost:8080/api/contacts> Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 11 ms Size: 455 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [{"status": "success", "message": "Contacts retrieved successfully", "data": [{"_id": "5f9669f3d7b895c313ec23c3", "create_date": "2020-10-26T06:17:23.692Z", "name": "Mary Jane", "gender": "Female", "email": "maryjane@gmail.com", "phone": "90010023", "__v": 0}]}]
```

For UPDATE:

1. Ensure that there is at least a contact in the database.
2. Click on the body tab and select the `x-www-form-urlencoded`.
3. Enter `http://localhost:8080/api/contacts/{contactid}`, where `{contactid}` is the id of the contact we want to update.
4. Select `Patch` or `Put` and input the key and the value of the information you want to change.
5. Press send and it will update the contacts.

PUT <http://localhost:8080/api/contacts/5f9669f3d7b895c313ec23c3> Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> name	Jane Mary			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 30 ms Size: 423 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [{"message": "Contact Info updated", "data": {"_id": "5f9669f3d7b895c313ec23c3", "create_date": "2020-10-26T06:17:23.692Z", "name": "Jane Mary", "gender": "Female", "email": "maryjane@gmail.com", "phone": "90010023", "__v": 0}}]
```

For DELETE:

1. Ensure that there is at least a contact in the database.
2. Click on the body tab and select the `x-www-form-urlencoded`.
3. Enter `http://localhost:8080/api/contacts/{contactid}`, where `{contactid}` is the id of the contact we want to delete.

4. Press send and it will delete the contact.

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost:8080/api/contacts/5f9669f3d7b895c313ec23c3
- Buttons:** Send, Save
- Tabs:** Params, Authorization, Headers (7), Body, Pre-request Script, Tests, Settings
- Query Params:** A table with columns KEY, VALUE, and DESCRIPTION. It contains one entry: Key, Value, Description.
- Body:** A tab showing the response body in JSON format: `{ "status": "success", "message": "Contact deleted" }`
- Status:** 200 OK, Time: 21 ms, Size: 260 B
- Buttons:** Save Response

References

[Building a RESTFUL API](#)

Task B2: Running test locally and via Travis

Required Application

- Required application for B1
- Mocha
- Chai

Steps to follow:

1. Do step 1-3 for B1.
2. Input `npm run test` in the terminal to run the test cases, it should all pass.
3. Head to [Travis](#) and select the cloned repository.
4. Push something to the repo or click on restart build to start travis testing. Travis should pass.

References

[Testing node express using Chai and Mocha](#)

[Documentation on how to write .travis.yml file](#)

Task B3: Deploying Serverless

Steps to run serverless

1. Use the following [link](#) to access the contacts.

2. To perform the REST API, replace `http://localhost:8080/api/contacts/` with `https://10vh6khhb11.execute-api.ap-southeast-1.amazonaws.com/dev/api/contacts`

References

[References to set up AWS lambda with Access Key and Secret Access Key](#)

[Steps to deploy serverless using current app](#)

[Linking MongoDB Atlas with AWS](#)

[Learning to set up secrets and permissions for Travis](#)

Task B4: Using the frontend SPA using Vue and Bootstrap

Setting up

Setting up backend:

1. In the root folder, run `npm run dev` to start the backend application locally
2. Terminal should display `Db connected successfully` and running at port 8080.

Open another terminal to set up frontend.

Setting up frontend:

1. Enter in terminal `cd contacts-app` and run `npm run serve -- --port 4040` to run the frontend at port 4040

Steps to execute API at frontend:

For **POST**:

1. Navigate to the contact form by clicking on the **ContactForm** tab on the navigation bar.
2. Input **Email Address**, **Name**, **Phone Number** and choose **Gender**.
3. Click submit and the information will be posted to the database.

Contacts **Home** **ContactForm** **Update/Delete**

Email address:

maryjane@gmail.com

Your Name:

Mary Jane

Phone:

90021022

Gender:

Female

Submit

Reset

Contacts **Home** **ContactForm** **Update/Delete**

Email address:

Enter email

Your Name:

Enter name

Phone:

Enter phone

Gender:

Submit

Reset

Successful !

For **GET**: Home page will display all the contacts that is in the database.

Contacts Home ContactForm Update/Delete

Mary Jane

Contact: 90021022

Gender: Female

Email:
maryjane@gmail.com

5f966c24d7b895c313ec23c5

For **UPDATE**:

1. Navigate to **Update/Delete** tab of the navigation bar.
2. Copy the **contact id** which is found at the bottom of the contact at the home page.
3. Input the **contact id** and click update.
4. Input the fields that you want to update and leave blank to leave the data untouched.
5. Click update once done and the selectd contact should be updated.

Contacts Home ContactForm Update/Delete

Contact ID:

Contacts Home ContactForm Update/Delete

Email address:

Your Name:

Phone:

Gender:

Contacts Home ContactForm Update/Delete

Email address:
janemary@gmail.com

Your Name:
Jane Mary

Phone:
Enter phone

Gender:
Please select an item in the list.

Submit

Contact Updated!

Contacts Home ContactForm Update/Delete

Jane Mary
Contact: 90021022
Gender: Female
Email:
janemary@gmail.com
5f966c24d7b895c313ec23c5

For **DELETE**:

1. Navigate to **Update/Delete** tab of the navigation bar.
2. Copy the **contact id** which is found at the bottom of the contact at the home page.
3. Input the **contact id** and click delete.
4. The selected contact should be deleted from the database.

Contacts Home ContactForm Update/Delete

Contact ID:
5f966c24d7b895c313ec23c5

Update Delete

Contacts Home ContactForm Update/Delete

Contact ID:
Enter contactid

Update Delete

Delete Successful !

References

[Initialising Vue and Bootstrap-Vue](#)

[Bootstrap-Vue Documentation](#)

[Using axios to consume API](#)