

# CS3219 Task B

---

## Student Information

Name: Kor Ming Soon Matriculation Number: A0201975A Github Repo:

<https://github.com/kormingsoon/cs3219-taskb>

Credit: <https://medium.com/@dinyangetoh/how-to-build-simple-restful-api-with-nodejs-expressjsand-mongodb-99348012925d>

I have only completed Task B1 and Task B4.

Note that in the section, I am segmenting Task B1 and Task B4 separately. The sections are as follows:

---

## Task B4

In task B4, I have implemented the react front end. The styling can be found in style.css and bootstrap has been incorporated.

1. Demonstration of API calls with front end.

- 1.1 POST & GET

# Create Contact

## Contact List

ID: 61499e726d50d1b39d1d006c

Name: Kor Ming Soon

Email: kormingsoon@gmail.com

1.2 PUT: upon clicking the edit button, a pop up window will appear allowing the user to edit the old inputs. Observe that the ID remains the same.

## Edit Contact

Kor Ming Soon

kormingsoon@gmail.com

Submit

Close

## Contact List

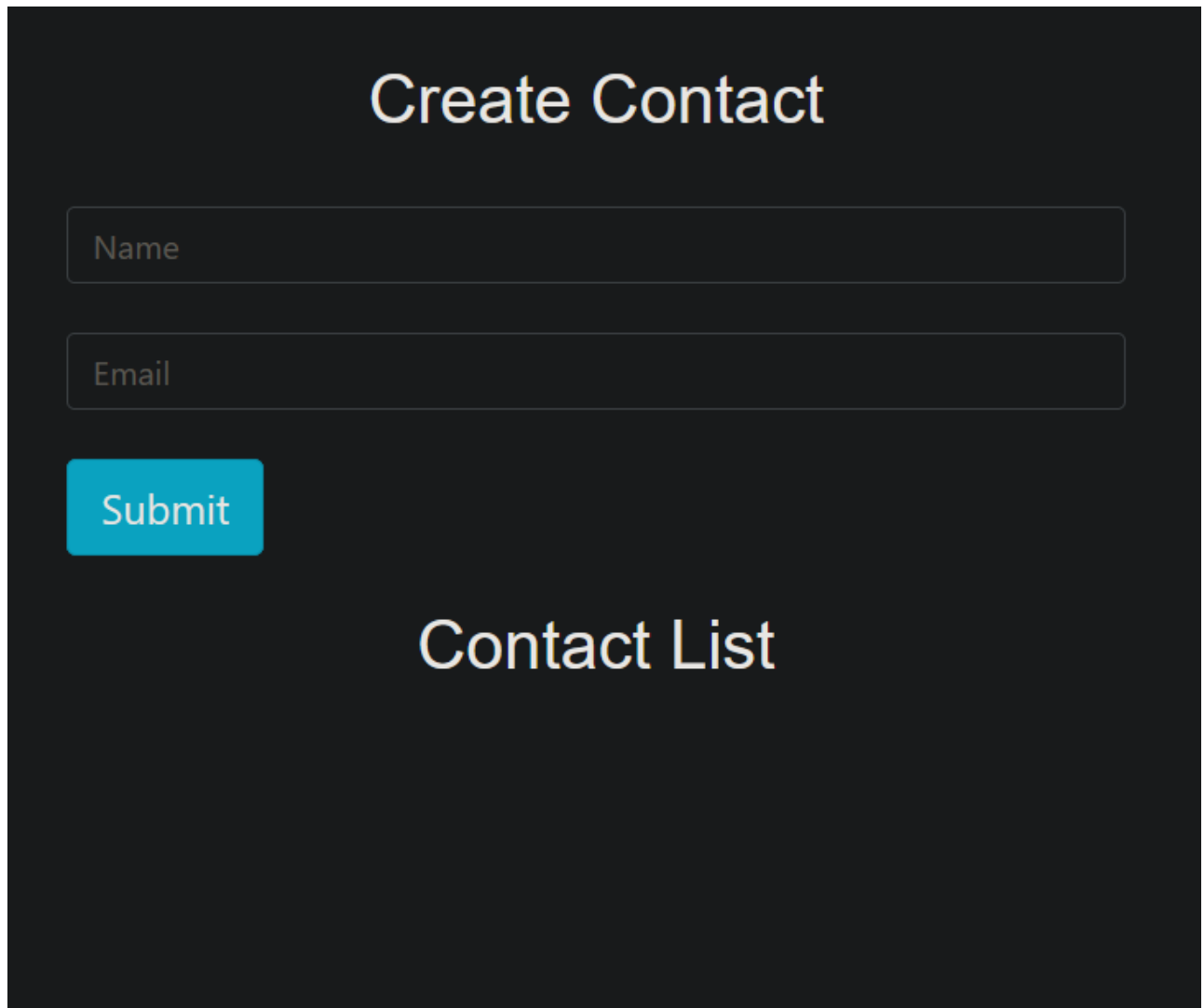
ID: 61499e726d50d1b39d1d006c

Name: Andrea Tan

Email: andreatanky@gmail.com

Edit Delete

1.3 DELETE : upon clicking the delete button. The contact list is now empty.



# Create Contact

# Contact List

---

## Task B1

- Local Deployment
- Hosted Deployment
- Error-Resiliency Testing

### Local Deployment

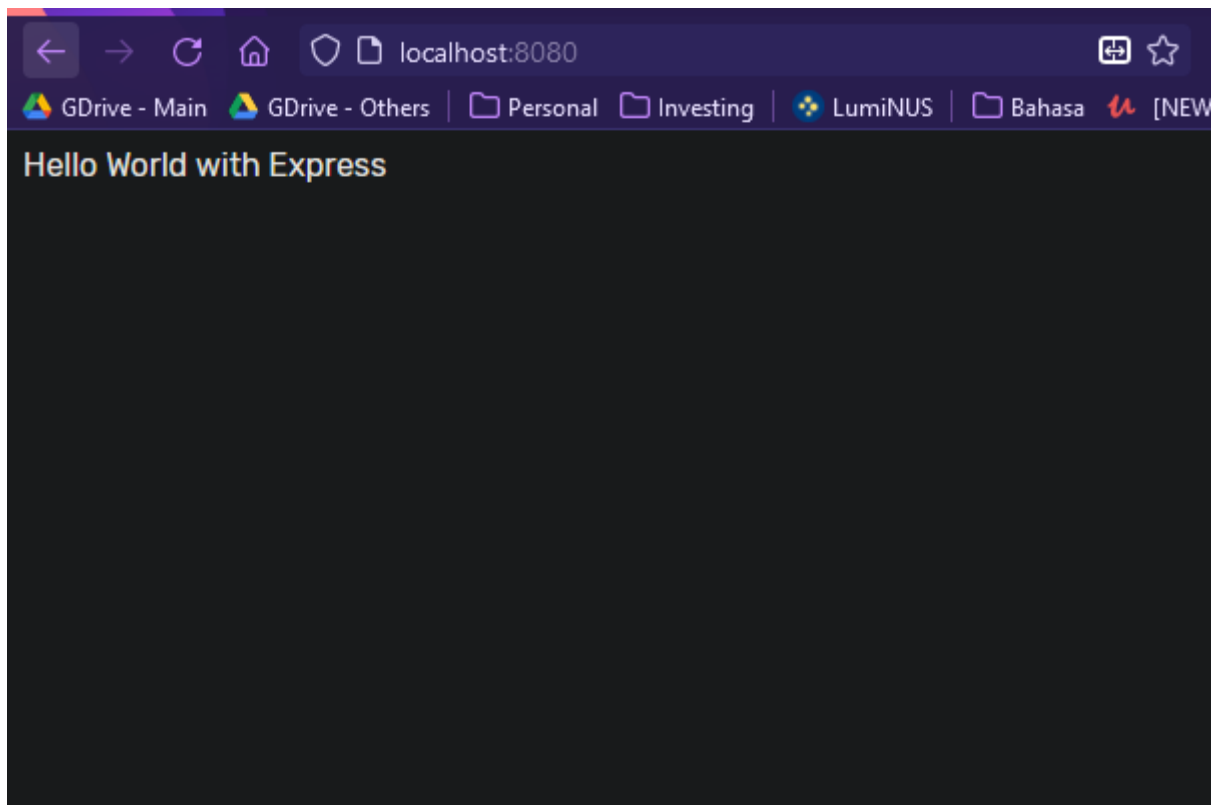
In this segment we will run and observe the GET, POST, PUT and DELETE API calls on a locally deployed web application using postman. In this example, we will use contacts, containing the name and email, as the model for our api calls.

1. Ensure that node.js is installed in your computer.
2. Run the app with the following command

```
npm start
```

3. After running the application, go to the following site and observe that the app is running on express.

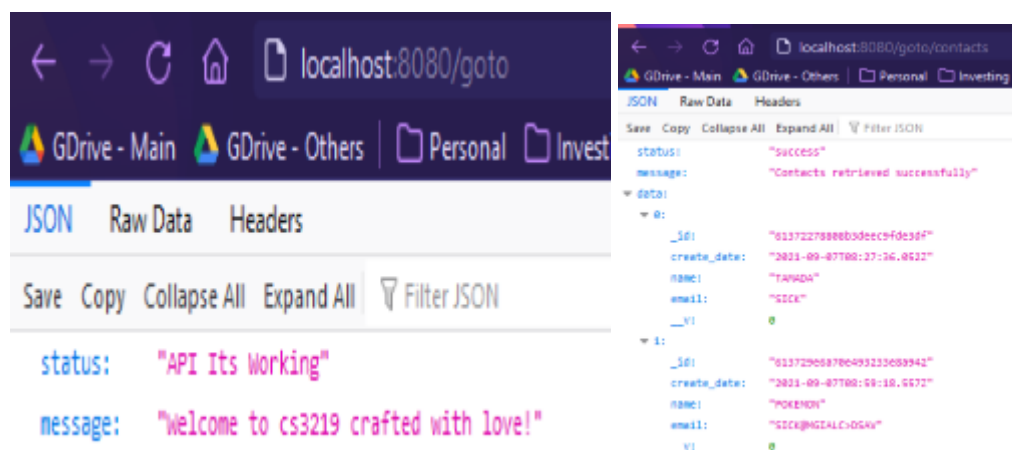
```
localhost:8080
```



```
localhost:8080
```

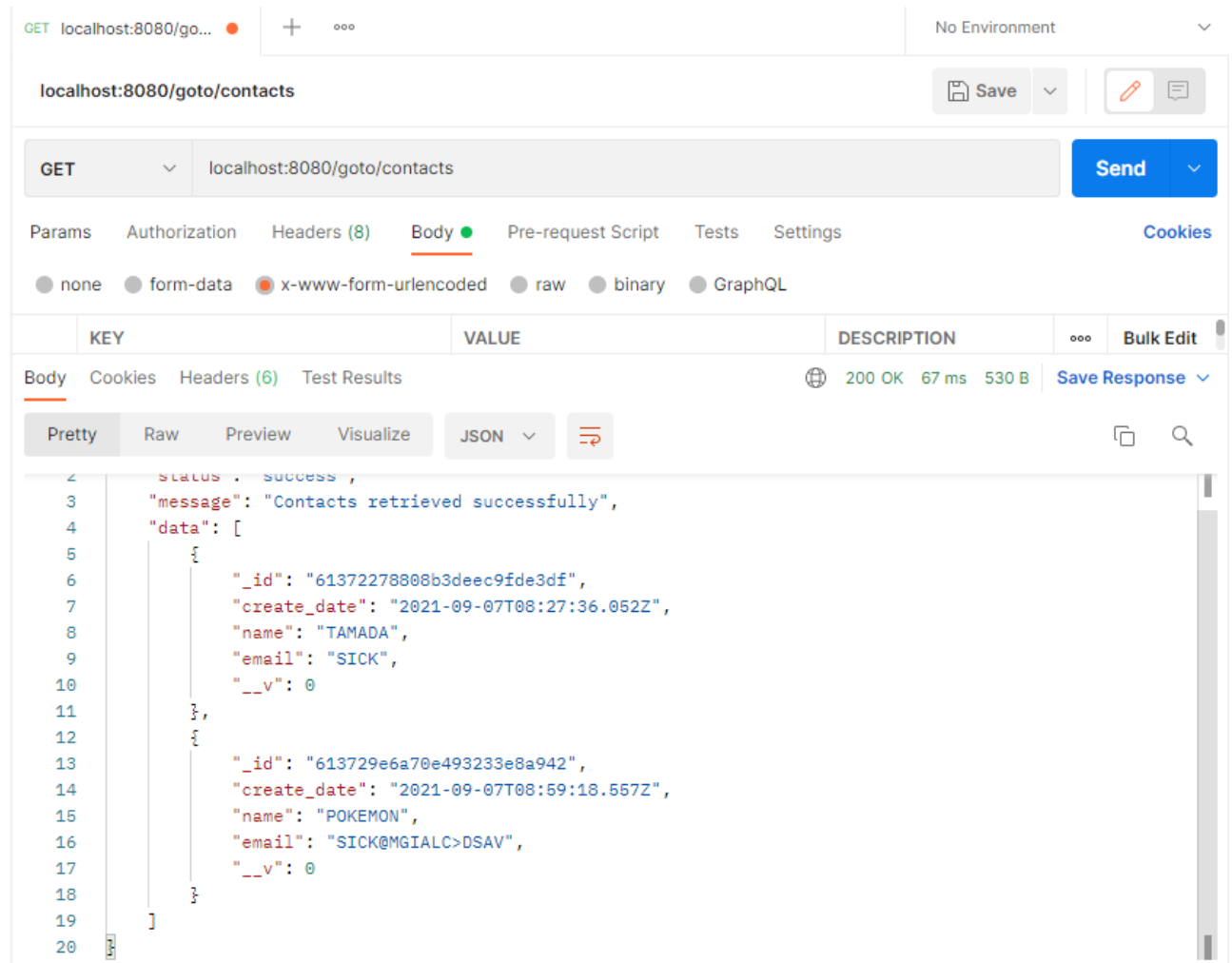
4. You can visit the following links to observe the routing. In `/contacts`, the data you observe may be different.

```
localhost:8080/goto  
localhost:8080/goto/contacts
```



5. Now, we will use postman to carry out the API calls.

5.1 For the **GET** API request. We can simply see that the requests returns the list of contacts in the database. Right now, there are two contact details. We access: `localhost:8080/goto/contacts`:



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

- Request:** GET `localhost:8080/goto/contacts`
- Response Status:** 200 OK, 67 ms, 530 B
- Response Body (JSON):**

```
{
  "status": "success",
  "message": "Contacts retrieved successfully",
  "data": [
    {
      "_id": "61372278808b3deec9fde3df",
      "create_date": "2021-09-07T08:27:36.052Z",
      "name": "TAMADA",
      "email": "SICK",
      "__v": 0
    },
    {
      "_id": "613729e6a70e493233e8a942",
      "create_date": "2021-09-07T08:59:18.557Z",
      "name": "POKEMON",
      "email": "SICK@MGIALC>DSAV",
      "__v": 0
    }
  ]
}
```

5.2 For the **POST** API request. For the post request, we will be adding in `name: Sahil is the best` and `email: sahilisthebest@gmail.com`. You can then observe that the contact is newly created. You can return back to `localhost:8080/goto/contacts` to see all of the contacts.

POST localhost:8080/goto/contacts

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Sahil Is the Best	
<input checked="" type="checkbox"/> email	llovesahil@gmail.com	
<input type="checkbox"/> gender	Male	
<input type="checkbox"/> phone	8222716	

Key Value Description

Body Cookies Headers (6) Test Results 200 OK 38 ms 396 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "message": "New contact created!",
3   "data": {
4     "_id": "613730aca6cabf8d33c2ca4b",
5     "create_date": "2021-09-07T09:28:12.363Z",
6     "name": "Sahil Is the Best",
7     "email": "llovesahil@gmail.com",
8     "__v": 0
9   }
10 }

```

5.3 For the **DELETE** API request. For the delete request, we will delete the first contact, which has the id `61372278808b3deec9fde3df`. We will key in `localhost:8080/goto/contacts/61372278808b3deec9fde3df` with a **DELETE** request.

```

4     "data": [
5       {
6         "_id": "61372278808b3deec9fde3df",
7         "create_date": "2021-09-07T08:27:36.052Z",
8         "name": "TAMADA",
9         "email": "SICK",
10        "__v": 0
11      },
12    ]

```

The above contains the contact with the id that we will delete

DEL localhost:8080/goto/contacts/61372278808b3deec9fde3df

Save Send

DELETE localhost:8080/goto/contacts/61372278808b3deec9fde3df

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION	Bulk Edit
Body	200 OK 26 ms 260 B	Save Response	

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "message": "Contact deleted"
4  }

```

The above illustrates the delete api through postman.

GET localhost:8080/goto/contacts/

Save Send

GET localhost:8080/goto/contacts/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION	Bulk Edit
Body	200 OK 23 ms 557 B	Save Response	

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "success",
3    "message": "Contacts retrieved successfully",
4    "data": [
5      {
6        "_id": "613729e6a70e493233e8a942",
7        "create_date": "2021-09-07T08:59:18.557Z",
8        "name": "POKEMON",
9        "email": "SICK@MGIALC>DSAV",
10       "__v": 0
11     },
12     {
13       "_id": "613730aca6cabf8d33c2ca4b",
14       "create_date": "2021-09-07T09:28:12.363Z",
15       "name": "Sahil Is the Best",
16       "email": "Ilovesahil@gmail.com",
17       "__v": 0
18     }
19   ]
20 }

```

The above illustrates the difference in the contact list after deletion.

5.4 For the PUT API request. We will edit the contact we added earlier (*Sahil is the best*) to a different name *Prof Bimlesh is the best*, along with the email. Notice that the id remains the same.



localhost:8080/goto/contacts/613730aca6cabf8d33c2ca4b

PUT localhost:8080/goto/contacts/613730aca6cabf8d33c2ca4b

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	Prof Bimlesh Is the Best			
<input checked="" type="checkbox"/>	email	Iloveprofbimlesh@gmail.com			
<input type="checkbox"/>	gender	Male			

Body Cookies Headers (6) Test Results 200 OK 28 ms 409 B Save Response

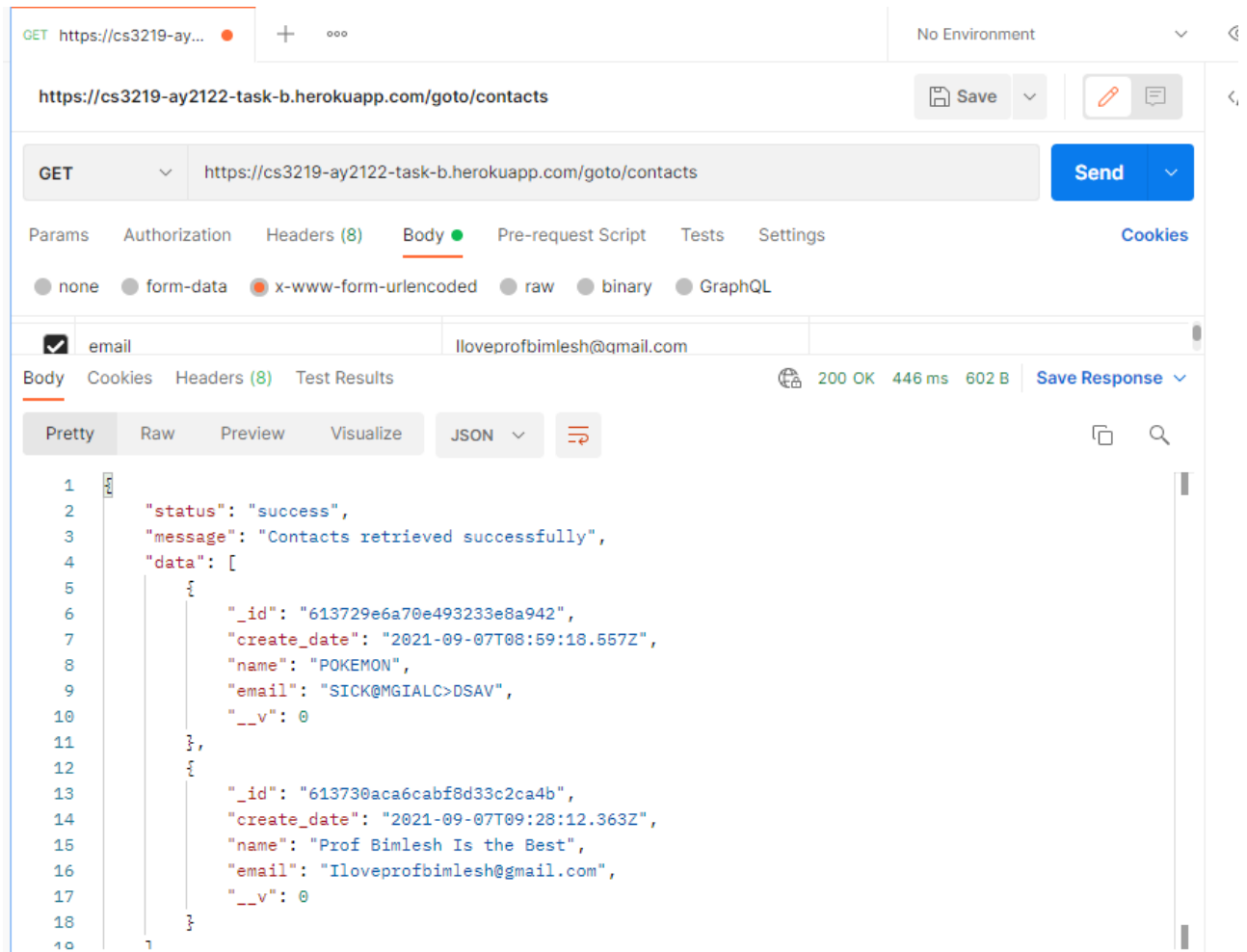
Pretty Raw Preview Visualize JSON

```
1  {
2    "message": "Contact Info updated",
3    "data": {
4      "_id": "613730aca6cabf8d33c2ca4b",
5      "create_date": "2021-09-07T09:28:12.363Z",
6      "name": "Prof Bimlesh Is the Best",
7      "email": "Iloveprofbimlesh@gmail.com",
8      "__v": 0
9    }
10 }
```

## Hosted Deployment

1. Similar to above, we will illustrate the API calls using the deployed endpoints instead of our localhost. The website is hosted on the link: <https://cs3219-ay2122-task-b.herokuapp.com/>. We will test the API calls on <https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts>.

1.1 For the **GET** API request. We can simply see that the requests returns the list of contacts in the database. Right now, there are two contact details. We access: <localhost:8080/goto/contacts>:



1.2 For the **POST** API request. For the post request, we will be adding in **name: Hosting is the best** and **email: ilovehosting@gmail.com**. You can then observe that the contact is newly created. You can return back to <https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts> to see all of the contacts.



1.3 For the **DELETE** API request. For the delete request, we will delete the first contact, which has the id **613729e6a70e493233e8a942**. We will key in <https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/613729e6a70e493233e8a942> with a DELETE request.

```
1  {
2    "status": "success",
3    "message": "Contacts retrieved successfully",
4    "data": [
5      {
6        "_id": "613729e6a70e493233e8a942",
7        "create_date": "2021-09-07T08:59:18.557Z",
8        "name": "POKEMON",
9        "email": "SICK@MGIALC>DSAV",
10       "__v": 0
11     },
12   ]
13 }
```

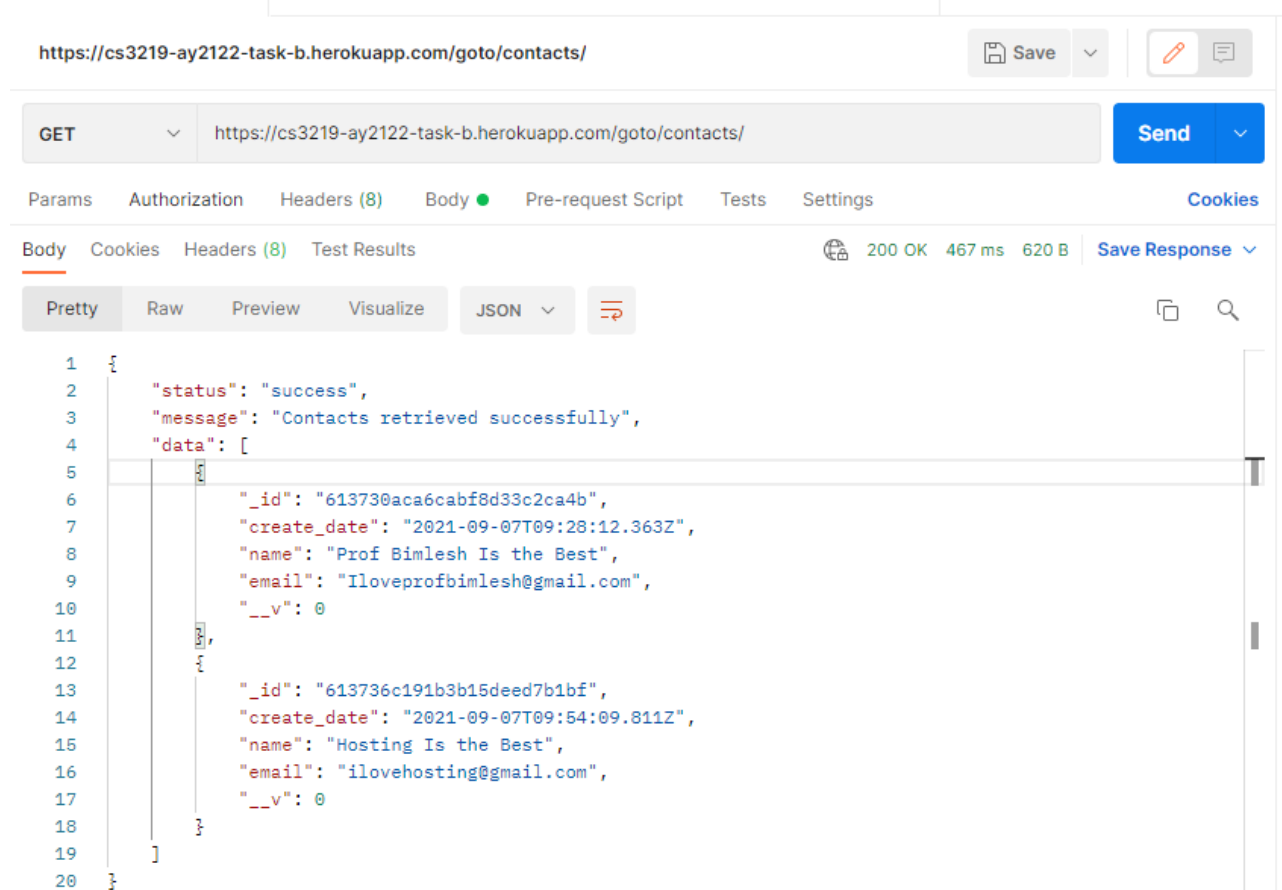
The above contains the contact with the id that we will delete

The screenshot shows a Postman interface for a DELETE request. The URL is `https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/613729e6a70e493233e8a942`. The request is configured with the method 'DELETE' and the body type 'x-www-form-urlencoded'. The response is a JSON object indicating success.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Hosting Is the Best	
<input checked="" type="checkbox"/> email	ilovehosting@gmail.com	
<input type="checkbox"/> gender	Male	

```
1  {
2    "status": "success",
3    "message": "Contact deleted"
4  }
```

The above illustrates the delete api through postman.



https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/

GET https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/ Send

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results 200 OK 467 ms 620 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "Contacts retrieved successfully",
4   "data": [
5     {
6       "_id": "613730aca6cabf8d33c2ca4b",
7       "create_date": "2021-09-07T09:28:12.363Z",
8       "name": "Prof Bimlesh Is the Best",
9       "email": "Iloveprofbimlesh@gmail.com",
10      "__v": 0
11    },
12    {
13      "_id": "613736c191b3b15deed7b1bf",
14      "create_date": "2021-09-07T09:54:09.811Z",
15      "name": "Hosting Is the Best",
16      "email": "ilovehosting@gmail.com",
17      "__v": 0
18    }
19  ]
20 }
```

The above illustrates the difference in the contact list after deletion.

1.4 For the PUT API request. We will edit the contact we added earlier (**Hosting is the best**) to a different name **Local deployment is the best**, along with the email. Notice that the id remains the same.

https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/613736c191b3b15deed7b1bf

PUT https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/613736c191b3b15deed7b1bf

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	Local deployment is the best			x
<input checked="" type="checkbox"/>	email	localdeploymentisthebest@gmail.com			
<input type="checkbox"/>	gender	Male			

Body Cookies Headers (8) Test Results 200 OK 1400 ms 453 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Contact Info updated",
3    "data": {
4      "_id": "613736c191b3b15deed7b1bf",
5      "create_date": "2021-09-07T09:54:09.811Z",
6      "name": "Local deployment is the best",
7      "email": "localdeploymentisthebest@gmail.com",
8      "__v": 0
9    }
10  }

```

## Error-Resiliency Testing

In the error-resiliency testing, we will be testing the error resiliency with the hosted domain.

Again, the domain is <https://cs3219-ay2122-task-b.herokuapp.com/>

a. Test Case 1:

<https://cs3219-ay2122-task-b.herokuapp.com/999999999>

https://cs3219-ay2122-task-b.herokuapp.com/999999999

GDive - Main GDive - Others Personal Investing LumiNUS Bahasa

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

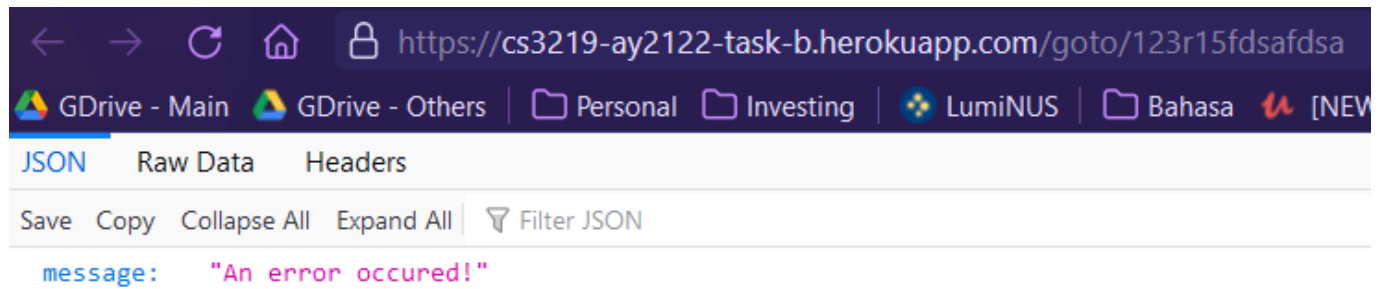
```

message: "An error occurred!"

```

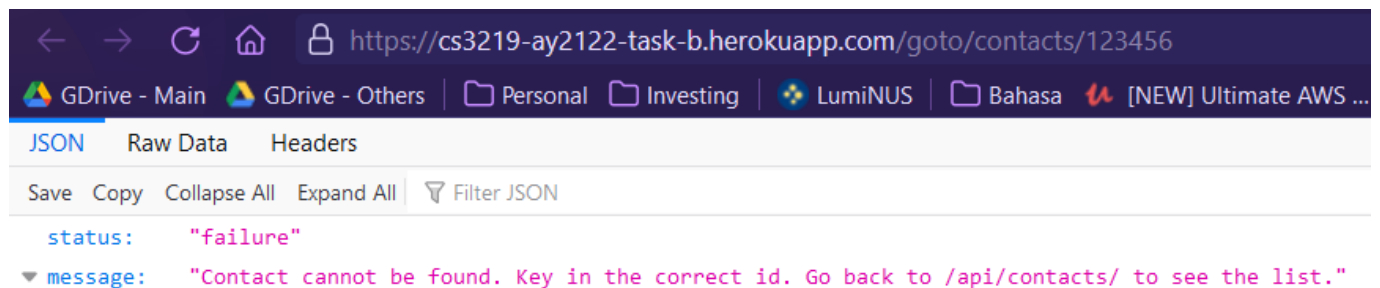
b. Test Case 2:

<https://cs3219-ay2122-task-b.herokuapp.com/goto/123r15fdsafdsa>



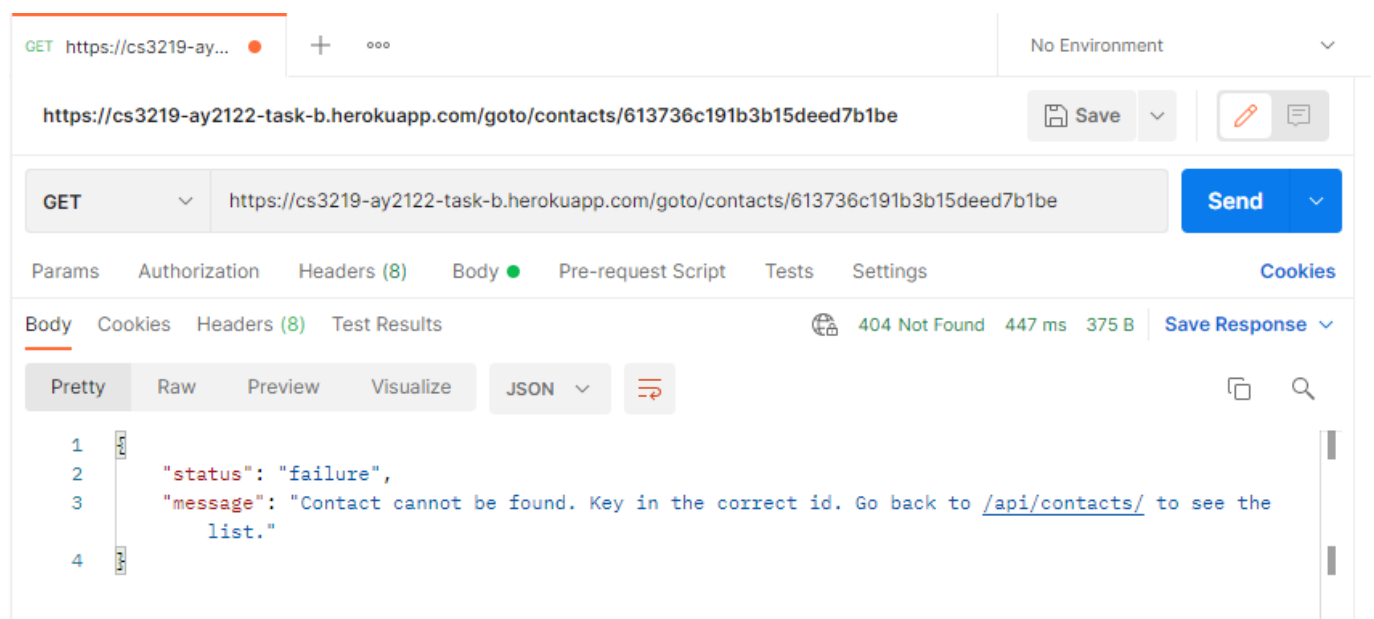
c. Test Case 3:

<https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/123456>



d. Test Case 4:

<https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/613736c191b3b15deed7b1be>



5. Test Case 5:

<https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/123456778> with DELETE request.

DELhttps://cs3219-ay...+...

No Environment

https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/123456778

Save

DELETE

https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts/123456778

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Body

Cookies

Headers (8)

Test Results

404 Not Found

734 ms

376 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

"status": "failure",

"message": "Contact cannot be found. Key in the correct id. Go back to /goto/contacts/ to see the list."

6. Test Case 6: PUT request with empty input field.

PUThttps://cs3219-ay...+...

No Environment

https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts

Save

PUT

https://cs3219-ay2122-task-b.herokuapp.com/goto/contacts

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input type="checkbox"/>	name	Local deployment is the best			x
<input type="checkbox"/>	email	localdeploymentisthebest@gmail.com			
<input type="checkbox"/>	gender	Male			

Body

Cookies

Headers (8)

Test Results

500 Internal Server Error

715 ms

294 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

"message": "An error occured!"