

SUBMISSION FOR B1 AND B2

Name: Gordon Foo

Matriculation Number: A0199554L

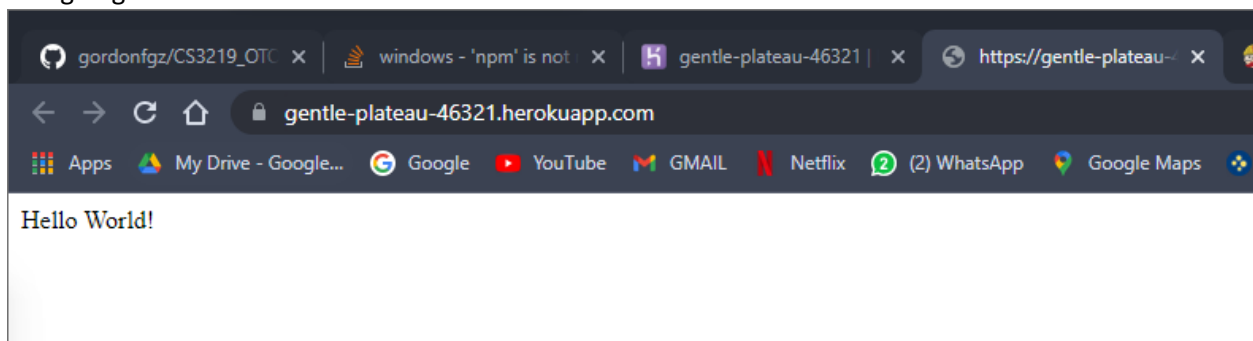
github link: https://github.com/gordonfgz/CS3219_OTOT_TASK_B.git

Running locally:

1. Open project in vscode
2. Open terminal and run npm start

Running on Heroku:

1. Make sure git is updated
2. Open terminal and run git push heroku main
terminal will run and give the url to the web app
going to the link shows the default Hello World



LOCALHOST TESTING

GET request results from localhost: (Get all existing users)

GET https://...

POST https://...

POST http://...

GET http://l...

PUT https://...

DEL https://l...

+

...No Environment

Save

Send

http://localhost:3000/users

GET

http://localhost:3000/users

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 10 msSize: 290 BSave Response

PrettyRawPreviewVisualizeJSON

1{2 "name": "Jeremy",3 "age": 224 },5 {6 "name": "Kelvin",7 "age": 218 },9 },10}

POST request results from localhost: (Add a new user John)

The screenshot displays a REST client interface with a tab for a POST request to `http://localhost:3000`. The request body is a JSON object: `{ "name": "John", "age": 33 }`. The response status is 201 Created, with a time of 5 ms and a size of 320 B. The response body is a JSON array of three user objects: `[{ "name": "Jeremy", "age": 22 }, { "name": "Kelvin", "age": 21 }, { "name": "John", "age": 33 }]`.

GET https://... POST http://... POST http://... GET http://l... PUT https://... DEL https://l... + ... No Environment

http://localhost:3000 Save

POST http://localhost:3000 Send

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

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

```
1 {
2   "name": "John",
3   "age": 33
4 }
```

Body Cookies Headers (7) Test Results Status: 201 Created Time: 5 ms Size: 320 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "name": "Jeremy",
4     "age": 22
5   },
6   {
7     "name": "Kelvin",
8     "age": 21
9   },
10  {
11    "name": "John",
12    "age": 33
13  }
14 ]
```

PUT request results from localhost: (Update Jeremy's name to Jerry)

GET https://... POST http://... POST http://... GET http://l... PUT http://l... DEL https://... + ... No Environment

http://localhost:3000/users/0 Save

PUT http://localhost:3000/users/0 Send

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

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

```
1 {
2   "name": "Jerry",
3   "age": 22
4 }
```

Body Cookies Headers (7) Test Results Status: 201 Created Time: 7 ms Size: 319 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "name": "Jerry",
4     "age": 22
5   },
6   {
7     "name": "Kelvin",
8     "age": 21
9   },
10  {
11    "name": "John",
12    "age": 33
13  }
14 }
```

DELETE request results from localhost: (delete first user aka Jerry)

DELETE http://localhost:3000/users/0

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 201 Created Time: 3 ms Size: 293 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "name": "Kelvin",
4     "age": 21
5   },
6   {
7     "name": "John",
8     "age": 33
9   }
10 }
```

DEPLOYED SERVER TESTING

GET request results from web server:

GET https://gentle-plateau-46321.herokuapp.com/users

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 784 ms Size: 299 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "name": "Jeremy",
4     "age": 22
5   },
6   {
7     "name": "Kelvin",
8     "age": 21
9   }
10 }
```

POST request results from web server:

The screenshot shows a REST client interface with a POST request to `https://gentle-plateau-46321.herokuapp.com/`. The request body is a JSON object: `{ "name": "John", "age": 33 }`. The response status is 201 Created, with a time of 264 ms and a size of 332 B. The response body is a JSON array containing three user objects: `[{ "name": "Jeremy", "age": 22 }, { "name": "Kelvin", "age": 21 }, { "name": "John", "age": 33 }]`.

```
1 {
2   "name": "John",
3   "age": 33
4 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 264 ms Size: 332 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "name": "Jeremy",
4     "age": 22
5   },
6   {
7     "name": "Kelvin",
8     "age": 21
9   },
10  {
11    "name": "John",
12    "age": 33
13  }
14 ]
15
```

PUT request results from web server:

The screenshot shows a REST client interface with a PUT request to `https://gentle-plateau-46321.herokuapp.com/users/0`. The request body is a JSON object: `{ "name": "Jerry", "age": 22 }`. The response status is 201 Created, with a time of 243 ms and a size of 303 B. The response body is a JSON array containing two user objects: `[{ "name": "Kelvin", "age": 21 }, { "name": "Jerry", "age": 22 }]`.

```
1 {
2   "name": "Jerry",
3   "age": 22
4 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 243 ms Size: 303 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "name": "Kelvin",
4     "age": 21
5   },
6   {
7     "name": "Jerry",
8     "age": 22
9   }
10 ]
```

DELETE request results from web server:

The screenshot shows a REST client interface with a DELETE request to `https://gentle-plateau-46321.herokuapp.com/users/0`. The response status is 201 Created, with a time of 242 ms and a size of 277 B. The response body is displayed in JSON format:

```
1 {
2   "name": "Kelvin",
3   "age": 21
4 }
```

EDGE CASE AND ERROR RESILIENCY

Trying to post a user with incorrect attributes throws an error: (ageX is not a correct attribute)

The screenshot shows a REST client interface with a POST request to `http://localhost:3000`. The request body is:

```
1 {
2   "name": "John",
3   "ageX": 33
4 }
```

The response status is 404 Not Found, with a time of 4 ms and a size of 238 B. The response body is displayed in HTML format:

```
1 error
```

Trying to post a user with additional attributes also throws an error:

The screenshot shows a Postman interface for a POST request to `http://localhost:3000`. The request body is a JSON object: `{ "name": "John", "age": 33, "phone": "98765432" }`. The response status is 200 OK, with a time of 22 ms and a size of 242 B. The response body is empty.

Trying to delete a user at index that is invalid: (there are only 2 users so index 2 is not valid)

The screenshot shows a Postman interface for a DELETE request to `http://localhost:3000/users/2`. The response status is 200 OK, with a time of 10 ms and a size of 247 B. The response body contains the error message: `Index out of bounds`.

TESTING USING MOCHA AND CHAI

```
PS C:\Users\gordo\OneDrive\Desktop\School\CS3219\OTOT_TASK_B> npm test

> otot_task_b@1.0.0 test C:\Users\gordo\OneDrive\Desktop\School\CS3219\OTOT_TASK_B
> mocha --exit
```

Example app listening at http://localhost:3000

Tests

GET /users

✓ should return Jeremy age 22 and Kelvin age 21

POST /

✓ should return Jeremy age 22 and Kelvin age 21 and Keane 33

PUT /users/:id

✓ should return Jerry age 21 and Kelvin age 21

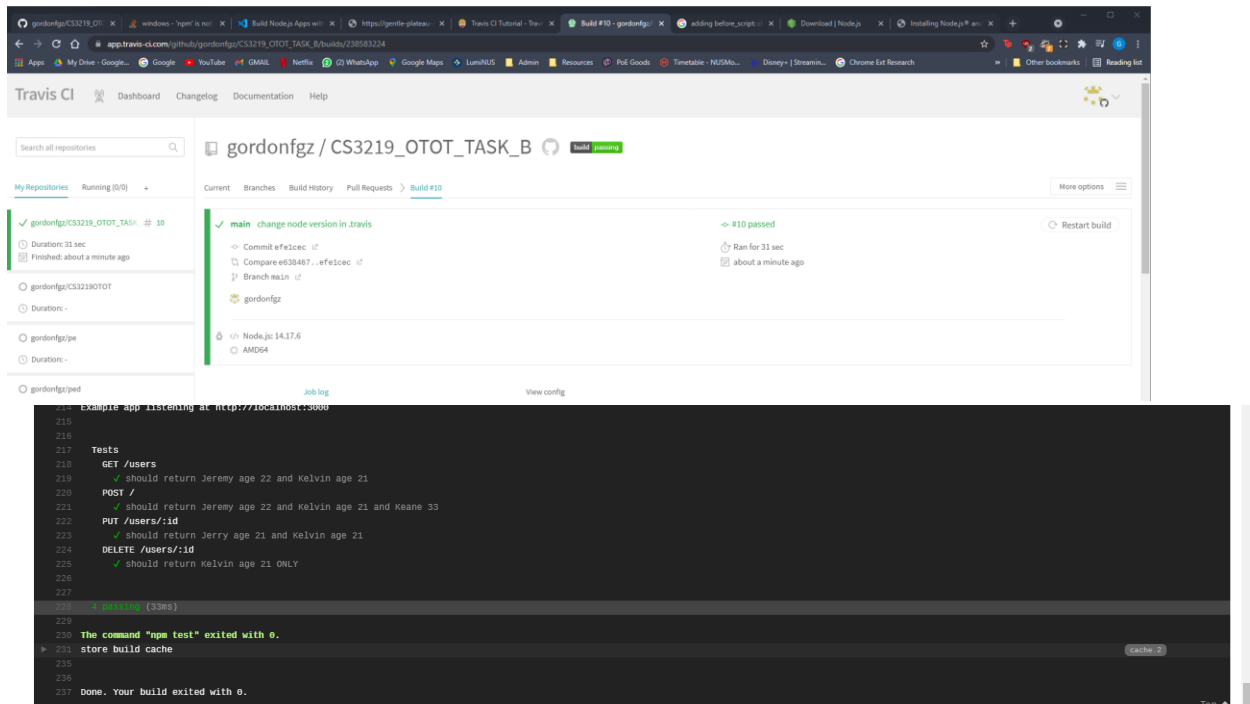
DELETE /users/:id

✓ should return Kelvin age 21 ONLY

4 passing (33ms)

```
PS C:\Users\gordo\OneDrive\Desktop\School\CS3219\OTOT_TASK_B> █
```

Using Travis to automate testing and building



Note:

The code has since been changed after completing task B4. However, the change is very minor. There is an addition of an Id field in the array of users. I have since updated the test.js in light of the change. However, since nothing conceptually/fundamentally changed (What I did in postman is the same as what I did in the testcases anyway), I kept the above screenshots as they are still able to show that the different requests work and the error handling considerations are the same.

Special thanks to:

<https://www.digitalocean.com/community/tutorials/test-a-node-restful-api-with-mocha-and-chai>

<https://stackoverflow.com/questions/24364361/permission-denied-when-running-npm-test-via-mocha-on-travis-ci/30141673>