

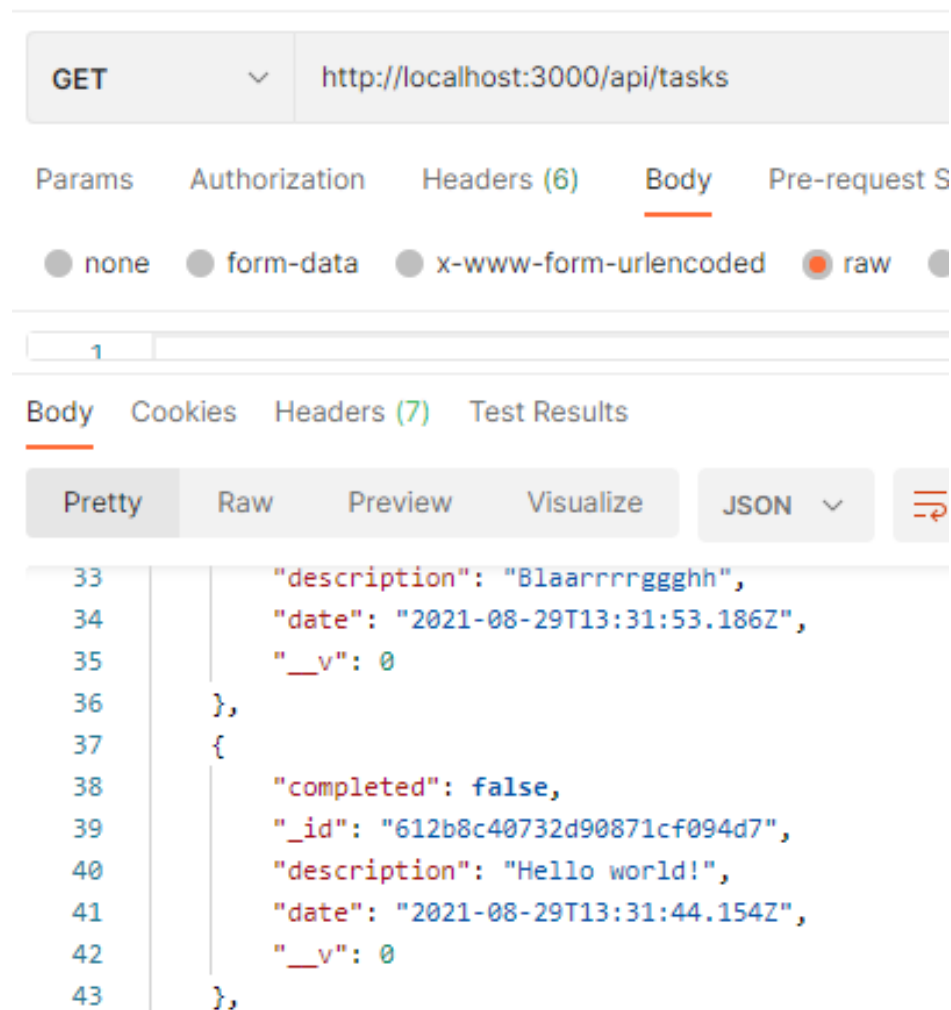
1a. Ambrose Liew Cheng Yuan, A0204750N

1b. [https://github.com/MorningLit/OTOT\\_Task\\_B\\_Backend](https://github.com/MorningLit/OTOT_Task_B_Backend)

1ci. To run the API locally, download a copy of my repository to your local machine. Next, change directory into the root folder(not inside the folder "frontend"! ) with my package.json and run "npm install", to install all the needed node module dependencies. Next, once you are done installing, run "npm run start", this starts the API locally, or you can run "npm run server" to start the server with nodemon.

Postman calls are demonstrated with the help of the following screenshots. To localhost,

**http://localhost:3000/api/tasks**



http://localhost:3000/api/tasks

POST

http://localhost:3000/api/tasks

Params

Authorization

Headers (8)

Body ●

Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {  
2   ... "description": "post to local using postman!"  
3 }
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "completed": false,  
3   "_id": "614c2847a135c143907d3433",  
4   "description": "post to local using postman!",  
5   "date": "2021-09-23T07:09:59.589Z",  
6   "__v": 0  
7 }
```

http://localhost:3000/api/tasks/614c2847a135c143907d3433

PUT

http://localhost:3000/api/tasks/614c2847a135c143907d3433

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

Settings

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

```
1 {  
2   ... "description": "put to local using postman!"  
3 }
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "completed": false,  
3   "_id": "614c2847a135c143907d3433",  
4   "description": "put to local using postman!",  
5   "date": "2021-09-23T07:09:59.589Z",  
6   "__v": 0  
7 }
```

DELETE ▼ http://localhost:3000/api/tasks/614c2847a135c143907d3433

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ Gr

1

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "completed": false,
3   "_id": "614c2847a135c143907d3433",
4   "description": "put to local using postman!",
5   "date": "2021-09-23T07:09:59.589Z",
6   "__v": 0
7 }
```

To deployed API,

<https://cs3219taskb.herokuapp.com/api/tasks>

GET ▼ https://cs3219taskb.herokuapp.com/api/tasks

Params Authorization Headers (6) Body Pre-request S

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐


1


Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ▼

```
1 [
2   {
3     "completed": false,
4     "_id": "614c25e920652200168d74d6",
5     "description": "hello!",
6     "date": "2021-09-23T06:59:53.343Z",
7     "__v": 0
8   },
9   ...
10 ]
```

https://cs3219taskb.herokuapp.com/api/tasks



POST  https://cs3219taskb.herokuapp.com/api/tasks

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

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


```
1 {  
2   ... "description": "posted to deployed api with postman!"  
3 }
```


**Body** Cookies Headers (9) Test Results


Pretty Raw Preview Visualize JSON  

```
1 {  
2   "completed": false,  
3   "_id": "614c293d20652200168d74da",  
4   "description": "posted to deployed api with postman!",  
5   "date": "2021-09-23T07:14:05.164Z",  
6   "__v": 0  
7 }
```

https://cs3219taskb.herokuapp.com/api/tasks/614c293d20652200168d74da



PUT  https://cs3219taskb.herokuapp.com/api/tasks/614c293d20652200168d74da

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** 


```
1 {  
2   ... "description": "put to deployed api with postman!"  
3 }
```

**Body** Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON  

```
1 {  
2   "completed": false,  
3   "_id": "614c293d20652200168d74da",  
4   "description": "put to deployed api with postman!",  
5   "date": "2021-09-23T07:14:05.164Z",  
6   "__v": 0  
7 }
```

https://cs3219taskb.herokuapp.com/api/tasks/614c293d20652200168d74da



DELETE  https://cs3219taskb.herokuapp.com/api/tasks/614c293d20652200168d74da

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

1

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON  

```
1 {
2   "completed": false,
3   "_id": "614c293d20652200168d74da",
4   "description": "put to deployed api with postman!",
5   "date": "2021-09-23T07:14:05.164Z",
6   "__v": 0
7 }
```

## Testing Edgecases

POST but with no body.

POST  https://cs3219taskb.herokuapp.com/api/tasks/

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** 

1

Body Cookies Headers (9) Test Results  Status: 500 Internal Server Error

Pretty Raw Preview Visualize JSON  

```
1 {
2   "message": "task validation failed: description: Path `description` is required."
3 }
```

PUT but with invalid id.

PUT  https://cs3219taskb.herokuapp.com/api/tasks/1

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** 

1

Body Cookies Headers (9) Test Results  Status: 500 Internal Server Error

Pretty Raw Preview Visualize JSON  

```
1 {
2   "message": "Cast to ObjectId failed for value \"1\" (type string) at path \"_id\" for model \"task\""
3 }
```

DELETE but with invalid id.

DELETE  https://cs3219taskb.herokuapp.com/api/tasks/1

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** 

1

Body Cookies Headers (9) Test Results  Status: 500 Internal Server Error

Pretty Raw Preview Visualize JSON  

```
1 {
2   "message": "Cast to ObjectId failed for value \"1\" (type string) at path \"_id\" for model \"task\""
3 }
```

1cii. To Access the deployed API, it is found hosted on Heroku. You can perform the CRUD operations on this weblink <https://cs3219taskb.herokuapp.com/api/tasks> .

You can perform a GET request on <https://cs3219taskb.herokuapp.com/api/tasks> .

You can perform a POST request on <https://cs3219taskb.herokuapp.com/api/tasks> . However, you must send a json body as an object containing “description” as the key and a value for it, to perform a successful POST request to the API.

You can perform a PUT request on

[<<YourObjectID>>](https://cs3219taskb.herokuapp.com/api/tasks/) . However, you must send a json body as an object containing “description” as the key and an edited value for it and the object you are trying to edit’s ID must be found in the URL where you replace “<<YourObjectID>>” with your object’s ID, to perform a successful PUT request to the API.

You can perform a DELETE request on

[<<YourObjectID>>](https://cs3219taskb.herokuapp.com/api/tasks/) . However, you must have the object ID you are trying to delete in the URL, where you replace the object ID you are trying to delete with “<<YourObjectID>>”, to perform a successful DELETE request to the API.

1ciii.

To run our tests locally, download a copy of my repository to your local machine. Next, change directory into the root folder(not inside the folder “frontend”!) with my package.json and next, run “npm install”, if you haven’t installed the required node modules. Once done, run “npm run test”, to run the tests locally. Hypothetically speaking, after a few seconds, you should be able to see the test’s output. However, you do not have my .env file which contains my MONGO\_URI environment variable as it is private and not supposed to be shared. Thus, you will see an error message.

To run our tests via travis, one way we could do is to push any changes of the repository to the GitHub repository and travis would automatically build and test for our backend application.

Another way we could do is to login to <https://app.travis-ci.com/> with my credentials and force a “Restart Build” to trigger our travis to build again and run the testcases. However, again, you do not have my travis credentials and access to my GitHub account, so you will not be able to test any of my tests via travis.

1civ. To set up the frontend, change directory into the folder, “frontend”. Next, run “npm install” to install all the needed node module dependencies. Next, once you are done installing, run “npm run serve” to start up our frontend locally. You can view the frontend on <http://localhost:8080/> on your browser.

1d. Starting the local backend server’s output

```
ambrc@LAPTOP-JULD6R3U MINGW64 ~/Downloads/nus/~y3s1/cs3219/assignments/otot/OTOT_Task_B/backend (master)
$ npm run start

> otot_task_b@1.0.0 start
> node server.js

listening on 3000
mongodb connected
```

Starting the local frontend server's output

```
DONE Compiled successfully in 4751ms

App running at:
- Local:   http://localhost:8080/
- Network: http://192.168.10.125:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

The frontend website

