

## **Task B : CRUD Application Task**

### **Submission for Task B1 and B2**

**Name:** Murugesan Karthika

**Matriculation Number:** A0191135L

**Link to Github:** <https://github.com/Kzrthikz/resthub>

For my own reference:

<https://dev.to/vunderkind/your-heroku-app-keeps-crashing-after-deploy-tell-me-about-it-1a71>

Git push heroku main

The live app is on:

<https://protected-taiga-25269.herokuapp.com>

<https://protected-taiga-25269.herokuapp.com/api/quotes>

## B1 – Implementing a simple backend

### An API for Quotes

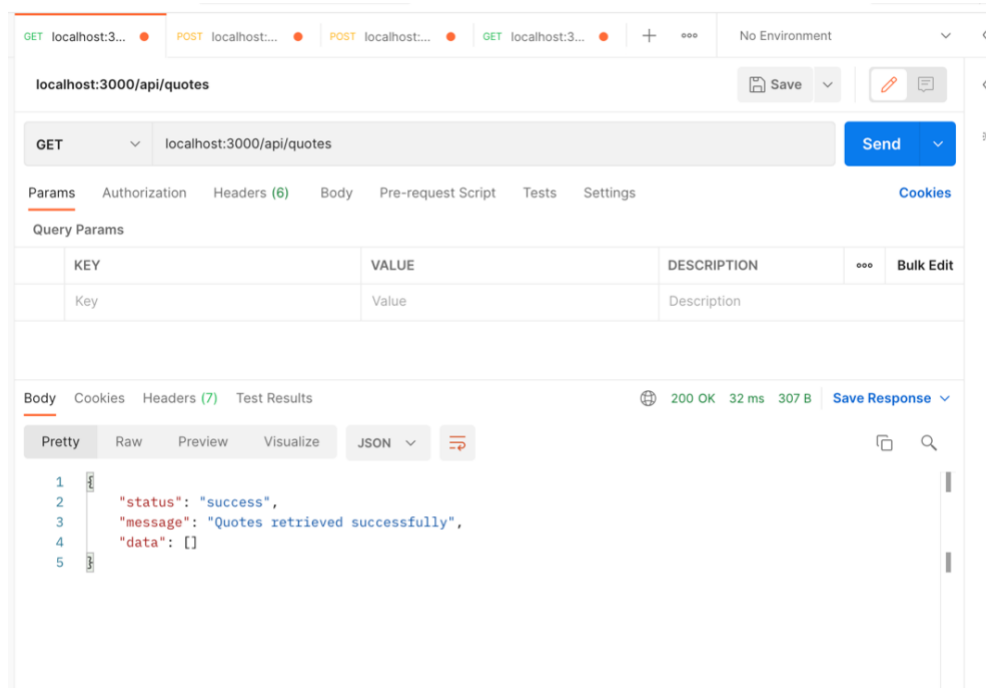
(Kindly note that in the code, I'm using localhost:8080 but in some of the screenshots, I'm using 3000 (as I was using this port initially). To test, please use 8080. Apologies for any confusion caused. Thanks.)

To run the API locally, use **npm start**

In postman kindly follow as demonstrated in these screenshots.

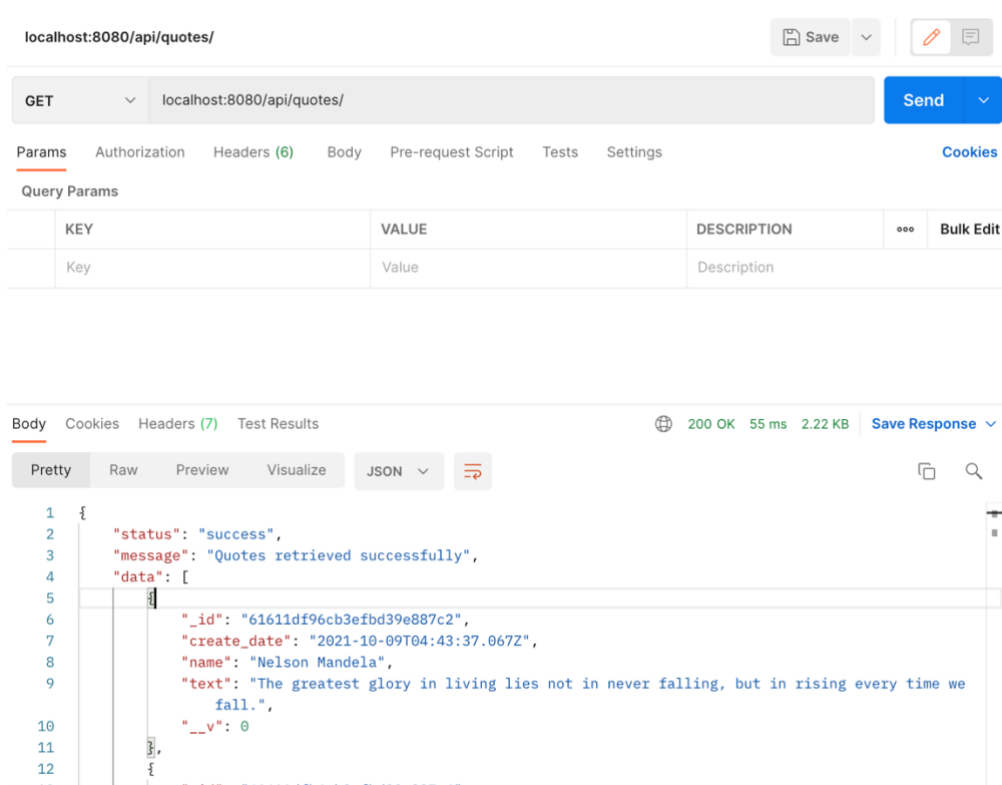
### GET localhost:8080/api/quotes

When no quote is available to show:



### GET localhost:8080/api/quotes

After quotes are added:



## GET localhost:8080/api/quotes/{quote\_id}

When we want to retrieve a specific quote using its quote\_id.

localhost:8080/api/quotes/61611df96cb3efbd39e887c2

GET localhost:8080/api/quotes/61611df96cb3efbd39e887c2

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 200 OK 26 ms 487 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Quote details loading..",
3   "data": {
4     "_id": "61611df96cb3efbd39e887c2",
5     "create_date": "2021-10-09T04:43:37.067Z",
6     "name": "Nelson Mandela",
7     "text": "The greatest glory in living lies not in never falling, but in rising every time we fall.",
8     "": "",
9     "": ""
10  }
11 }
```

[edge case]

## GET localhost:8080/api/quotes/{quote\_id}

When we want to retrieve a specific quote using an invalid quote\_id.

localhost:8080/api/quotes/aaa

GET localhost:8080/api/quotes/aaa

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 400 Bad Request 7 ms 274 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Quote not found."
3 }
```

## POST localhost:8080/api/quotes

To add a new quote. Kindly provide name and text of the quote in the request body.

The screenshot shows a REST client interface for a POST request to `localhost:3000/api/quotes`. The request body is in JSON format, containing a message and a data object with quote details.

**Request Body (JSON):**

```
{  "message": "New quote created!",  "data": {    "_id": "6150383e20f55ac21c05efb8",    "create_date": "2021-09-26T09:07:10.049Z",    "name": "Nelson Mandela",    "text": "The greatest glory in living lies not in never falling, but in rising every time we fall.",    "__v": 0  }}
```

**Response (200 OK):**

```
{  "status": "success",  "message": "Quotes retrieved successfully",  "data": [    {      "_id": "6150383e20f55ac21c05efb8",      "create_date": "2021-09-26T09:07:10.049Z",      "name": "Nelson Mandela",      "text": "The greatest glory in living lies not in never falling, but in rising every time we fall.",      "__v": 0    },    {      "_id": "6150388320f55ac21c05efbb",      "create_date": "2021-09-26T09:08:19.532Z",      "name": "Walt Disney",      "text": "The way to get started is to quit talking and begin doing.",      "__v": 0    }  ]}
```

## GET localhost:8080/api/quotes

To show the newly added quotes.

The screenshot shows a REST client interface for a GET request to `localhost:3000/api/quotes`. The response body is in JSON format, containing a status, message, and an array of quote data objects.

**Response (200 OK):**

```
{  "status": "success",  "message": "Quotes retrieved successfully",  "data": [    {      "_id": "6150383e20f55ac21c05efb8",      "create_date": "2021-09-26T09:07:10.049Z",      "name": "Nelson Mandela",      "text": "The greatest glory in living lies not in never falling, but in rising every time we fall.",      "__v": 0    },    {      "_id": "6150388320f55ac21c05efbb",      "create_date": "2021-09-26T09:08:19.532Z",      "name": "Walt Disney",      "text": "The way to get started is to quit talking and begin doing.",      "__v": 0    }  ]}
```

## PUT localhost:8080/api/quotes/{quote\_id}

For example, to update a quote id's text attribute to another quote of Nelson Mandela.

localhost:3000/api/quotes/6150383e20f55ac21c05efb8

PUT localhost:3000/api/quotes/6150383e20f55ac21c05efb8

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	Nelson Mandela			
<input checked="" type="checkbox"/>	text	A winner is a dreamer who never gives up.			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 53 ms 434 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Quote Info updated",
3   "data": {
4     "_id": "6150383e20f55ac21c05efb8",
5     "create_date": "2021-09-26T09:07:10.049Z",
6     "name": "Nelson Mandela",
7     "text": "A winner is a dreamer who never gives up.",
8     "__v": 0
9   }
10 }
```

## DELETE localhost:8080/api/quotes/{quote\_id}

localhost:3000/api/quotes/6150383e20f55ac21c05efb8

DELETE localhost:3000/api/quotes/6150383e20f55ac21c05efb8

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 200 OK 34 ms 281 B Save Response

Pretty Raw Preview Visualize JSON

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

## GET localhost:8080/api/quotes

Quote should no longer exist after deleting.

The screenshot shows a REST client interface with the URL `localhost:3000/api/quotes` and the method `GET`. The response status is `200 OK` with a response time of `6 ms` and a body size of `480 B`. The response body is displayed in JSON format:

```
{
  "status": "success",
  "message": "Quotes retrieved successfully",
  "data": [
    {
      "_id": "6150388320f55ac21c05efbb",
      "create_date": "2021-09-26T09:08:19.532Z",
      "name": "Walt Disney",
      "text": "The way to get started is to quit talking and begin doing.",
      "__v": 0
    }
  ]
}
```

[Edge case]

## DELETE localhost:8080/api/quotes/{quote\_id}

Delete an invalid quote\_id.

The screenshot shows a REST client interface with the URL `localhost:8080/api/quotes/aaa` and the method `DELETE`. The response status is `400 Bad Request` with a response time of `5 ms` and a body size of `293 B`. The response body is displayed in JSON format:

```
{
  "message": "Quote not found. Unable to delete. "
}
```

## Deployed endpoints on Heroku

Deployed API link: <https://protected-taiga-25269.herokuapp.com>

GET <https://protected-taiga-25269.herokuapp.com/api/quotes>

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

- URL:** `https://protected-taiga-25269.herokuapp.com/api/quotes/`
- Method:** GET
- Response Status:** 200 OK, 466 ms, 872 B
- Response Body (JSON):**

```
1 {
2   "status": "success",
3   "message": "Quotes retrieved successfully",
4   "data": [
5     {
6       "_id": "61611df96cb3efbd39e887c2",
7       "create_date": "2021-10-09T04:43:37.067Z",
8       "name": "Nelson Mandela",
9       "text": "The greatest glory in living lies not in never falling, but in rising every time we
10        fall.",
11       "__v": 0
12     },
13     {
14       "_id": "61611df96cb3efbd39e887c3",
15       "create_date": "2021-10-09T04:43:37.067Z",
16       "name": "Nelson Mandela",
17       "text": "The greatest glory in living lies not in never falling, but in rising every time we
18        fall.",
19       "__v": 0
20     }
21   ]
22 }
```

GET [https://protected-taiga-25269.herokuapp.com/api/quotes/{quote\\_id}](https://protected-taiga-25269.herokuapp.com/api/quotes/{quote_id})

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

- URL:** `https://protected-taiga-25269.herokuapp.com/api/quotes/61611df96cb3efbd39e887c2`
- Method:** GET
- Response Status:** 200 OK, 1020 ms, 496 B
- Response Body (JSON):**

```
1 {
2   "message": "Quote details loading..",
3   "data": {
4     "_id": "61611df96cb3efbd39e887c2",
5     "create_date": "2021-10-09T04:43:37.067Z",
6     "name": "Nelson Mandela",
7     "text": "The greatest glory in living lies not in never falling, but in rising every time we fall.",
8     "__v": 0
9   }
10 }
```

[Edge case]

GET [https://protected-taiga-25269.herokuapp.com/api/quotes/{quote\\_id}](https://protected-taiga-25269.herokuapp.com/api/quotes/{quote_id})

Where an invalid quote\_id is provided.

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

- URL:** `https://protected-taiga-25269.herokuapp.com/api/quotes/aaa`
- Method:** GET
- Response Status:** 400 Bad Request, 273 ms, 283 B
- Response Body (JSON):**

```
{  "message": "Quote not found."}
```

POST <https://protected-taiga-25269.herokuapp.com/api/quotes>

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

- URL:** `https://protected-taiga-25269.herokuapp.com/api/quotes/`
- Method:** POST
- Body:** `{ "name": "Dalai Lama", "text": "The purpose of our lives is to be happy." }`
- Response Status:** 200 OK, 1140 ms, 438 B
- Response Body (JSON):**

```
{  "message": "New quote created!",  "data": {    "_id": "616168a43d7f215e03c3155a",    "create_date": "2021-10-09T10:02:12.926Z",    "name": "Dalai Lama",    "text": "The purpose of our lives is to be happy.",    "__v": 0  }  }
```



PUT: [https://protected-taiga-25269.herokuapp.com/api/quotes/{quote\\_id}](https://protected-taiga-25269.herokuapp.com/api/quotes/{quote_id})

Update the quote id's text attribute

The screenshot shows a REST client interface with a PUT request to `https://protected-taiga-25269.herokuapp.com/api/quotes/616168a43d7f215e03c3155a`. The request body is set to `x-www-form-urlencoded`. The parameters table shows the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Dalai Lama	
<input checked="" type="checkbox"/> text	Be Kind whenever possible.	

The response body is displayed in JSON format:

```
1  {
2    "message": "Quote Info updated",
3    "data": {
4      "_id": "616168a43d7f215e03c3155a",
5      "create_date": "2021-10-09T10:02:12.926Z",
6      "name": "Dalai Lama",
7      "text": "Be Kind whenever possible. ",
8      "__v": 0
9    }
10 }
```

The screenshot shows a REST client interface with a GET request to `https://protected-taiga-25269.herokuapp.com/api/quotes/`. The response body is displayed in JSON format:

```
22  {
23    "name": "Henry Ford",
24    "text": "The only real mistake is the one from which we learn nothing.",
25    "__v": 0
26  },
27  {
28    "_id": "616168a43d7f215e03c3155a",
29    "create_date": "2021-10-09T10:02:12.926Z",
30    "name": "Dalai Lama",
31    "text": "Be Kind whenever possible. ",
32    "__v": 0
33  }
34 }
```

DELETE [https://protected-taiga-25269.herokuapp.com/api/quotes/{quote\\_id}](https://protected-taiga-25269.herokuapp.com/api/quotes/{quote_id})

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

- Method:** DELETE
- URL:** <https://protected-taiga-25269.herokuapp.com/api/quotes/616168a43d7f215e03c3155a>
- Status:** 200 OK
- Time:** 474 ms
- Size:** 290 B
- Body (JSON):**

```
{  "status": "success",  "message": "Quote deleted"}
```

[Edge Case]

DELETE [https://protected-taiga-25269.herokuapp.com/api/quotes/{quote\\_id}](https://protected-taiga-25269.herokuapp.com/api/quotes/{quote_id})

When an invalid quote\_id is provided.

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

- Method:** DELETE
- URL:** <https://protected-taiga-25269.herokuapp.com/api/quotes/aaa>
- Status:** 400 Bad Request
- Time:** 261 ms
- Size:** 302 B
- Body (JSON):**

```
{  "message": "Quote not found. Unable to delete. "}
```

The deployed endpoint can handle common edge cases such as retrieving invalid `quote_id` or deleting a `quote_id`. It does not crash.

## B2 – Testing through CI

I have added tests for `/GET/api/books`, `/POST`, `/PUT` and `/DELETE` endpoints.

To run the tests locally: **npm test**

In the `.travis.yml` file, **npm install** installs all the dependencies.

**mocha – timeout 10000** initiates the test.

Running the test locally:

```
Quotes
/GET quotes
  ✓ it should GET all the quotes (301ms)
/POST quotes
  ✓ it should POST a quote (38ms)
/PUT/:id quotes
  ✓ it should UPDATE a quote by the given id (41ms)
/DELETE/:id quotes
(node:81431) [MONGODB DRIVER] Warning: collection.remove is deprecated. Use deleteOne, deleteMany, or bulkWrite instead.
(Use `node --trace-warnings ...` to show where the warning was created)
  ✓ it should UPDATE a quote by the given id

4 passing (417ms)
```

Testing through Travis for one of the builds:

The screenshot shows the Travis CI interface for the repository 'Kzrthikz / resthub'. At the top, there's a header with the repository name, a GitHub icon, and a 'build unknown' status. Below this, there are tabs for 'Current', 'Branches', 'Build History', and 'Pull Requests'. The 'Current' tab is active, showing a build for the 'main' branch with the title 'main timeout for travis'. The build status is '#2 started' and it has been 'Running for 4 min 29 sec'. There are links for 'Commit 8f23f12', 'Compare 08f11f6..8f23f12', and 'Branch main'. The user 'Murugesan Karthika' is associated with the build. At the bottom, there are links for 'Job log' and 'View config'.

✕ Remove log ⌵ Raw log

▶ 1	Worker information	worker_info	0.07s
6			0.01s
▶ 7	Build system information	system_info	
161			
162			2.64s
		docker_mtu_and_registry_mirrors	
		resolvconf	
▶ 163	Installing SSH key from: default repository key	ssh_key	
165	Using /home/travis/.netrc to clone repository.		
166			
▶ 167	\$ git clone --depth=50 --branch=main https://github.com/Kzrthikz/resthub.git Kzrthikz/resthub	git_checkout	1.57s
177			
178			0.01s
▶ 179	\$ nvm install 12	nvm_install	2.82s
198			
▶ 199	Setting up build cache	cache.1	
207			
▶ 208		cache.npm	
210	\$ node --version		
211	v12.22.7		
212	\$ npm --version		
213	6.14.15		
214	\$ nvm --version		
215	0.38.0		
216			
▶ 217	\$ npm ci	install.npm	3.16s

216	\$ echo '***** Build Starts *****'	0.00s
217	***** Build Starts *****	
218	The command "echo '***** Build Starts *****' exited with 0.	
219	\$ echo 'Installing Dependencies'	0.00s
220	Installing Dependencies	
221	The command "echo 'Installing Dependencies'" exited with 0.	
222	\$ npm install	1.52s
223	npm WARN resthub@1.0.0 No repository field.	
224		
225	audited 185 packages in 1.06s	
226		
227	23 packages are looking for funding	
228	run 'npm fund' for details	
229		
230	found 0 vulnerabilities	
231		
232	The command "npm install" exited with 0.	
233	\$ echo 'Running tests'	0.00s
234	Running tests	
235	The command "echo 'Running tests'" exited with 0.	
236	\$ mocha --timeout 10000	
237	Db connected successfully	
238	Running RestHub on port 8080	
239		
240		
241	Quotes	
242	/GET quotes	
243	✓ it should GET all the quotes (3187ms)	
244	/POST quotes	
245	✓ it should POST a quote (226ms)	
246	/PUT/:id quotes	
247	✓ it should UPDATE a quote by the given id (619ms)	
248	/DELETE/:id quotes	
249	(node:5192) [MONGODB DRIVER] Warning: collection.remove is deprecated. Use deleteOne, deleteMany, or bulkWrite instead.	
250	✓ it should UPDATE a quote by the given id (421ms)	
251		