



rtshkmr Add headers to writeups ...

✓ 9 minutes ago [History](#)

..

backend	10 hours ago
frontend	25 minutes ago
resources	32 minutes ago
README.md	9 minutes ago

README.md



Student Name: Ritesh Kumar

Matriculation Number: A0201829H

[GitHub Repository](#)

B: A simple CRUD Application

B1: A simple backend

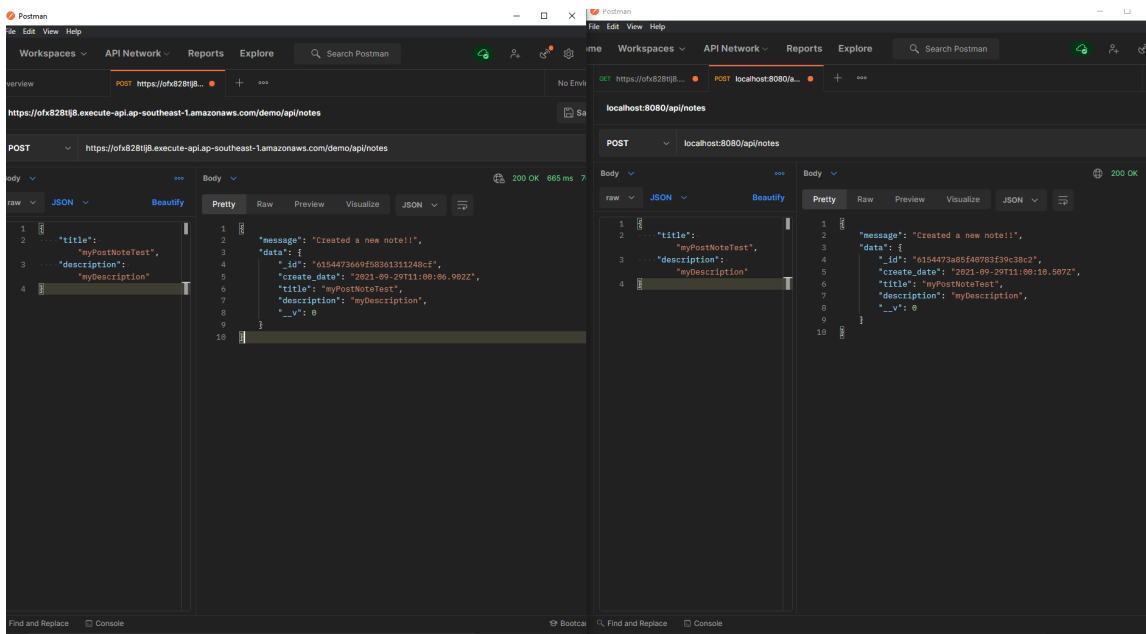
Reference: using this [notes tutorial](#)

CRUD Actions via API endpoints

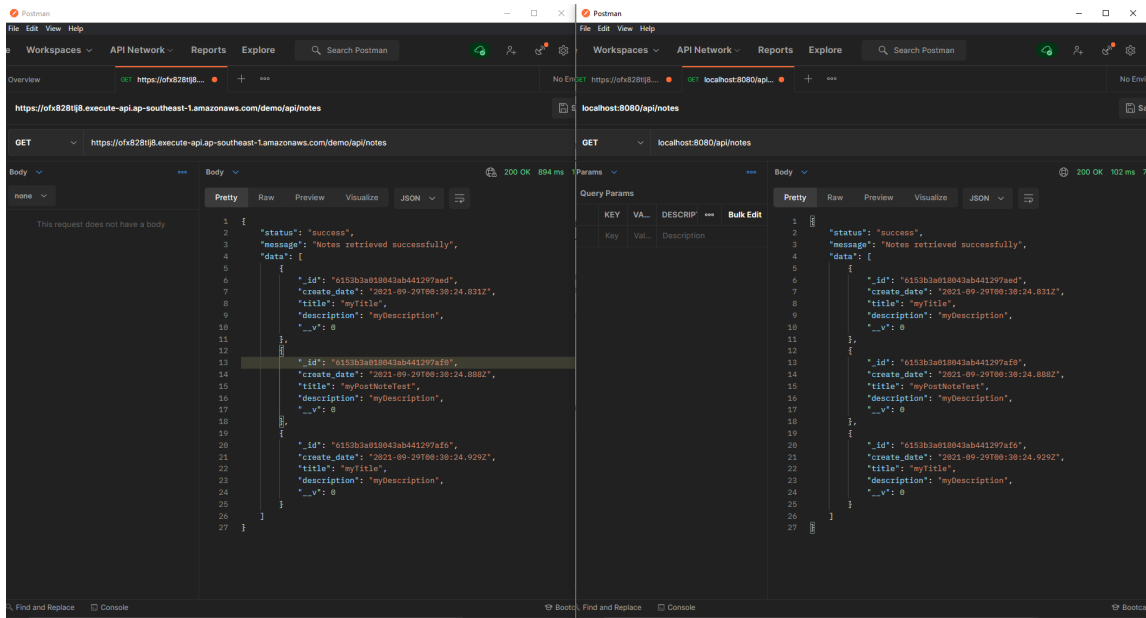
Note: db is hosted at MongoDB Cloud Atlas, API gateway is over at [here](#)

The following screenshots show behaviour on deployed api (left) and localhost(right)

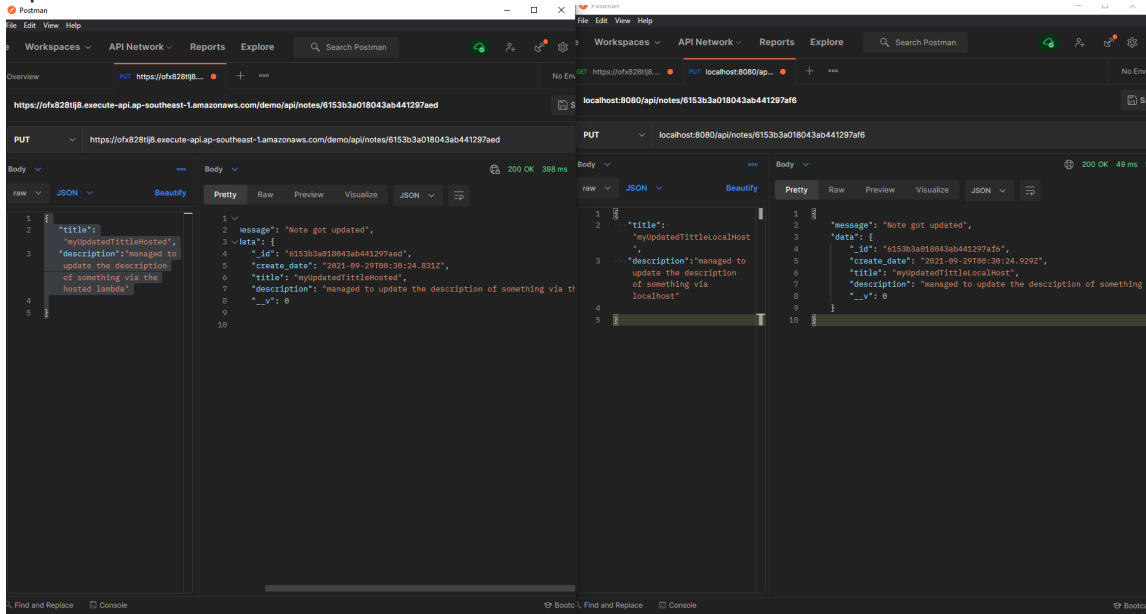
1. Create Notes:



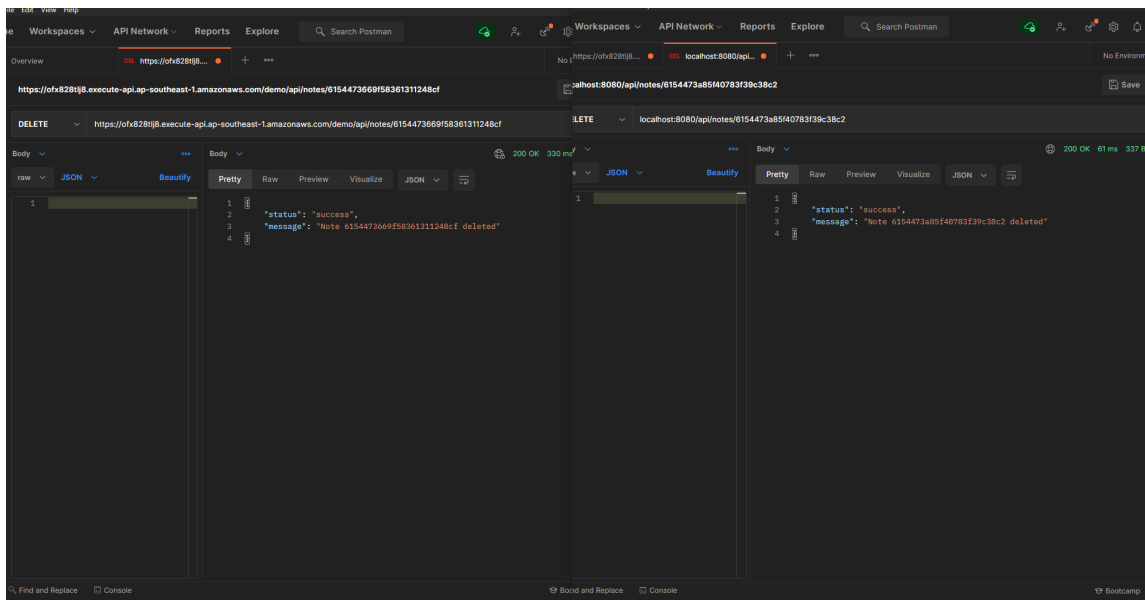
2. Read Notes:



3. Update Notes:

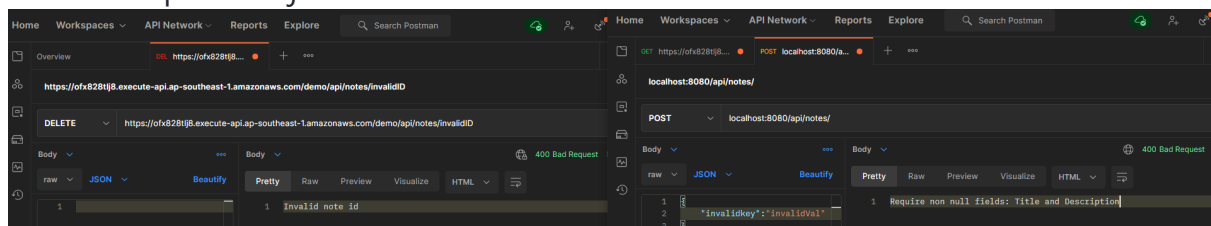


4. Delete Notes:



Basic edge cases:

On the left, deleting something that isn't there, on the right, posting something without required object structure.



****Takeaways:**

1. How express uses middleware via [the .use function](#)
 2. [Mongodb installation](#). hate windows for this.
- To connect to your local MongoDB, you set Hostname to localhost and Port to 27017.

B2: Testing via CI

MochaJS

pointers on usage:

1. Mocha discovers test files; when given no files or directories, it finds files with extensions .js, .mjs or .cjs in the test directory (but not its children), relative to the current working directory ==> can't store it in subdirs and need to be in working dir

2. try not to use lambda functions because of environment scoping issues [see this](#)

Chai

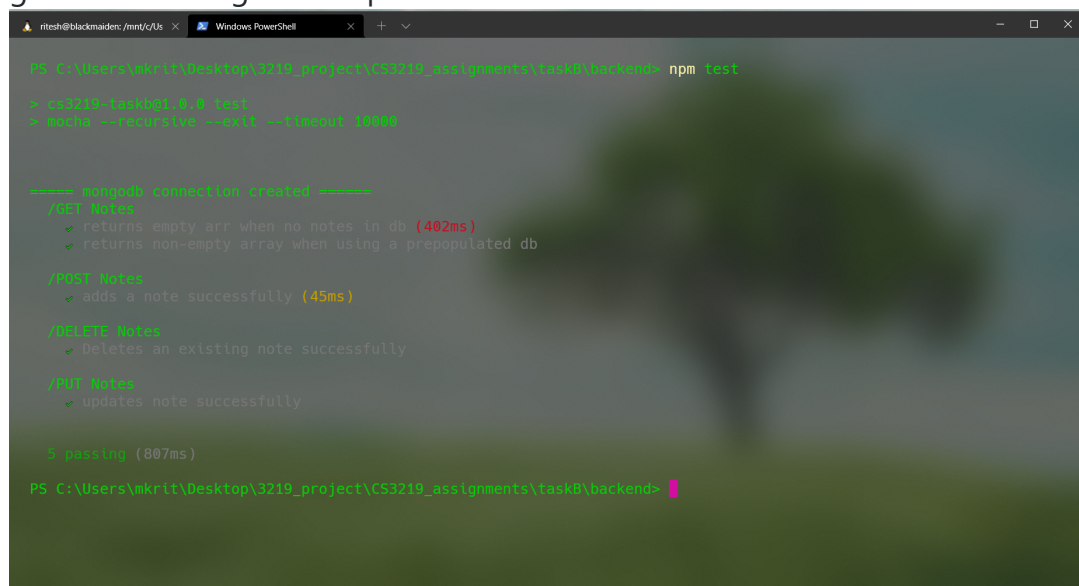
- an assertion library to work together with your testing framework
- has three assertion styles: as usual assert style, as an expect style or should style of defining assertions

GitHub Actions:

1. Uses doesn't use the working directory
2. Uses and run can't be tgt in the same step
3. Zipping is a pain, have to set the destination as the same directory as the contents (and don't zip the entire folder, zip its contents instead). Can check by exporting the lambda deployment files!

Deliverables:

1. [How to test](#) when there's a mongo client being applied to your server application (and hence the server application would complain if you just pass in an export keyword)
2. How to conduct testing:
 - i. **Locally**, navigate to [taskB/backend](#) directory and via cli, run `npm test` to get the following test output:



```
PS C:\Users\mkrit\Desktop\3219_project\CS3219_assignments\taskB\backend> npm test

> cs3219-taskB@1.0.0 test
> mocha --recursive --exit --timeout 10000

===== mongodb connection created =====
/GET Notes
  ✓ returns empty arr when no notes in db (402ms)
  ✓ returns non-empty array when using a prepopulated db

/POST Notes
  ✓ adds a note successfully (45ms)

/DELETE Notes
  ✓ Deletes an existing note successfully

/PUT Notes
  ✓ updates note successfully

3 passing (807ms)
PS C:\Users\mkrit\Desktop\3219_project\CS3219_assignments\taskB\backend>
```

- ii. Via CI, following the deployment script [here](#), gives us a similar github runner output

```
deploy (14.x, 5.0)
succeeded 5 hours ago in 53s

> ☒ print list and stuff

v ☒ Run all chai tests

1 ▶ Run npm run test_local
6
7 > cs3219-taskb@1.0.0 test_local /home/runner/work/CS3219_assignments/CS3219_assignments/taskB/backend
8 > cross-env MONGODB_URL=mongodb://localhost/notesTest mocha --recursive --exit --timeout 10000
9
10
11
12 ===== mongodb connection created =====
13 /GET Notes
14   ✓ returns empty arr when no notes in db (95ms)
15   ✓ returns non-empty array when using a prepopulated db
16
17 /POST Notes
18   ✓ adds a note successfully
19
20 /DELETE Notes
21   ✓ Deletes an existing note successfully
22
23 /PUT Notes
24   ✓ updates note successfully
25
26
27 5 passing (229ms)
28
```

. Here's a [link to the job](#)

3.

Issues Faced:

1. for some reason, the http server persisted after tests (i.e. mocha doesn't cause the server to kill itself once done). [Here's a workaround](#)

Followed the adding of `mocha --exit` to the test script instead, the other solutions didn't work :(

2. Was homicidal after trying out Travis. Eventually got it working but somehow couldn't get the API gateway ==> Lambda right
 - i. ended up just redoing everything and using github actions (and a simpler example tutorial)

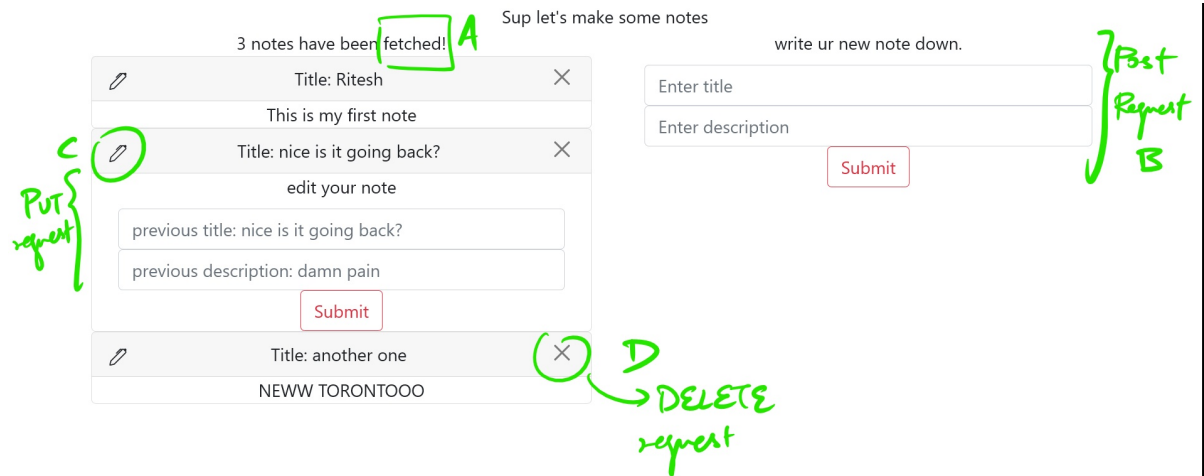
B3 - Deployment through Continuous Deployment

Figured out travis [over at this branch](#), but couldn't get the API gateway working. Even managed to setup the Lambda and see logs that imply it works. Anyway, ended up purging the whole thing and restarting.

Backend API Endpoint

B4 - Frontend Implementation

A simple react app that calls all the API endpoints. It uses ReactBootstrap for styled components. This annotated screenshot shows the interactions, the app is basic and intuitive.



Running Instructions

In the [project directory](#), you can run:

npm start

Runs the app in the development mode.

Open <http://localhost:3000> to view it in the browser.

The page will reload if you make edits.

You will also see any lint errors in the console.