Name: Tay Hui Chun

Matric No: A0170109N

Task B

GitHub Repository Link: https://github.com/huichun66/cs3219otot-taskB

Referenced Tutorial Links: https://medium.com/@dinyangetoh/how-to-build-simple-restful-api-with-nodejs-expressjs-and-mongodb-99348012925d

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

https://medium.com/swlh/continous-integration-using-travis-nodejs-8c608662a0cd

Instructions:

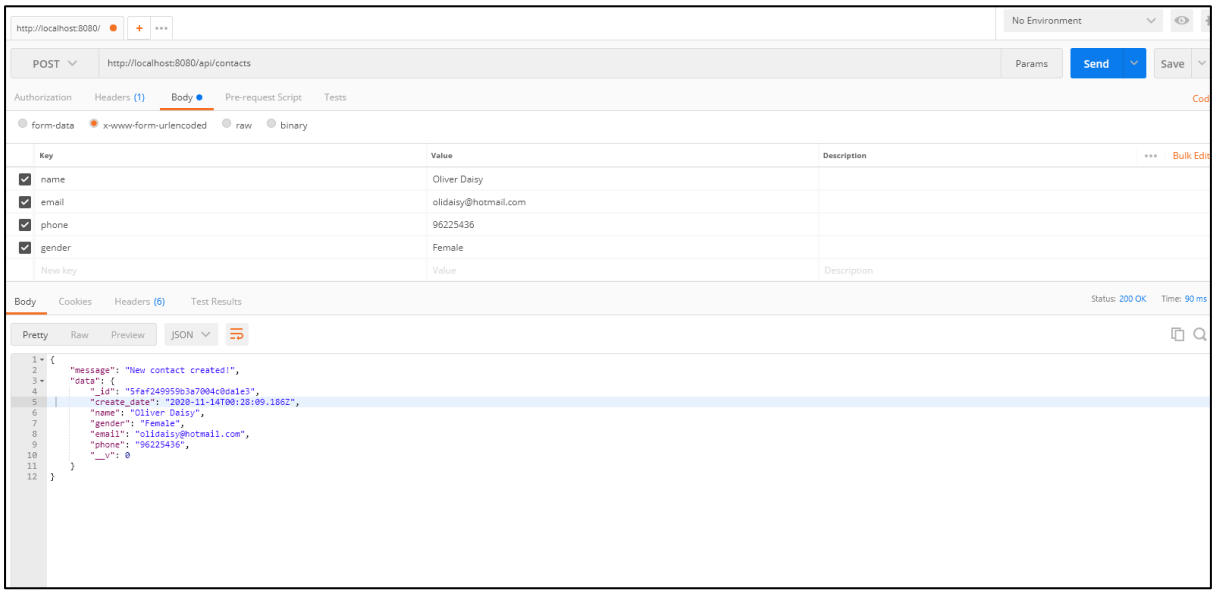
Running the APIs locally (B1):

```
cd resthub
node index
```

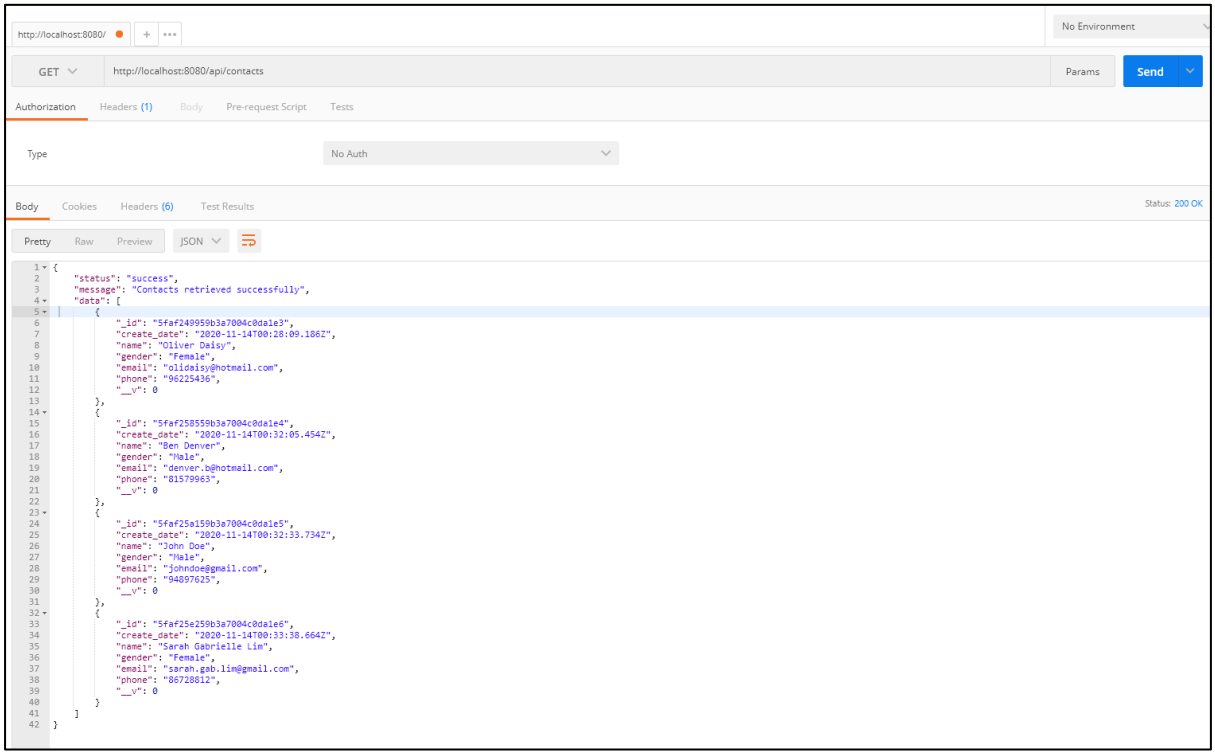
Access deployed APIs (B1):

Download the Postman app from https://www.postman.com/downloads/

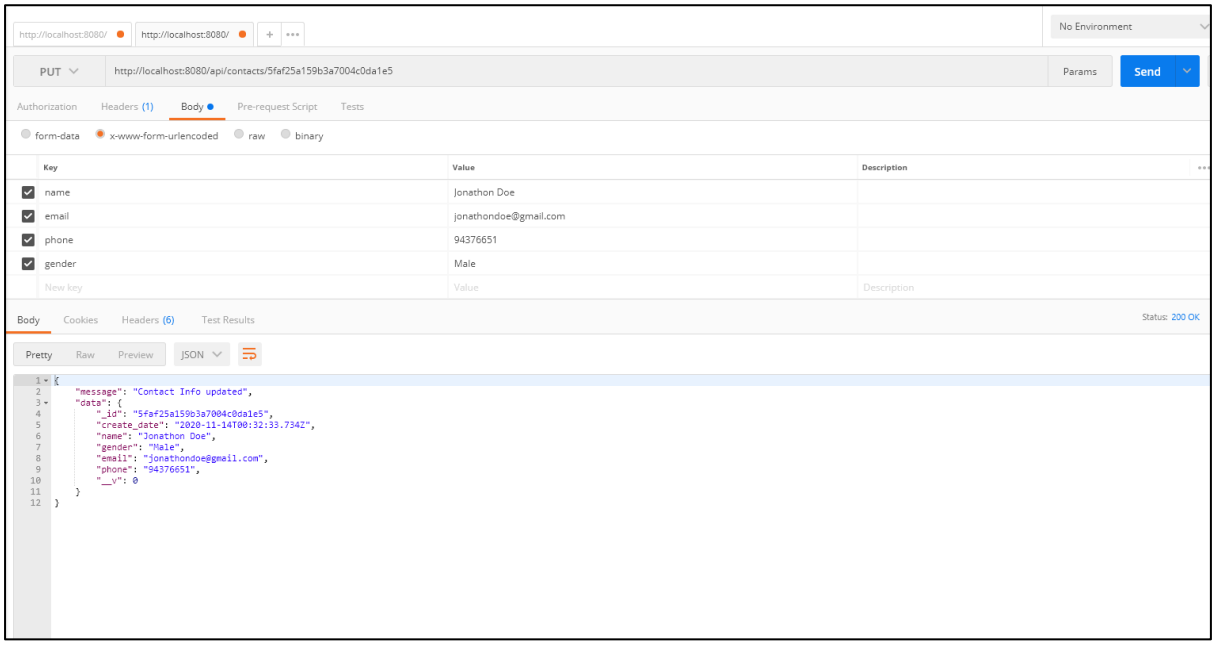
First, before we demo GET calls on Postman to retrieve list of contacts, we have to create some contacts in the database. To do that, we use POST api call (`/api/contacts`) on Postman to create some contacts. This is the sample screenshot shown for a successfully created contact (POST request).



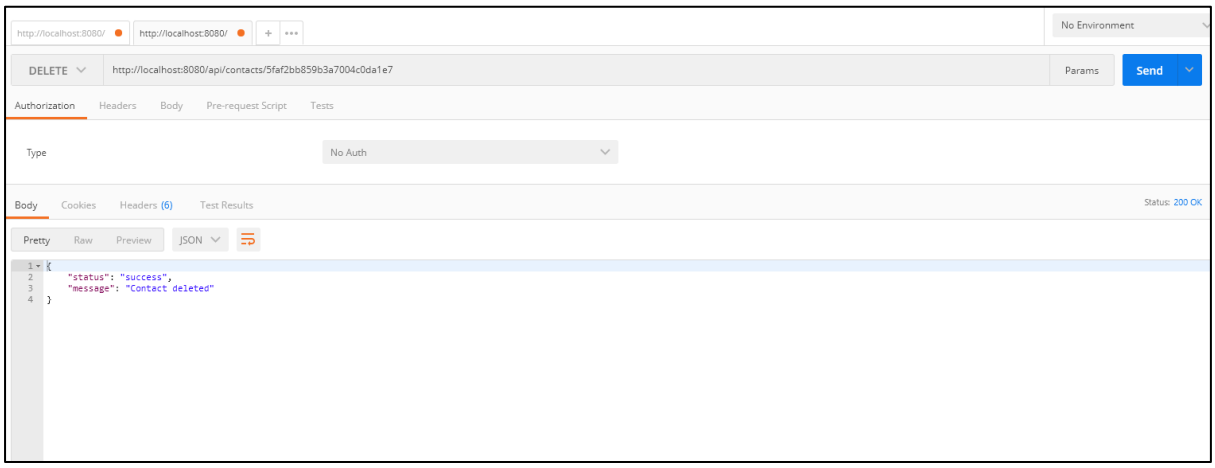
After creating some Contact records, we can now demo the GET api call on Postman with the following url shown in the screenshot below. Upon successful retrieval, a list of contacts from the database should be retrieved.



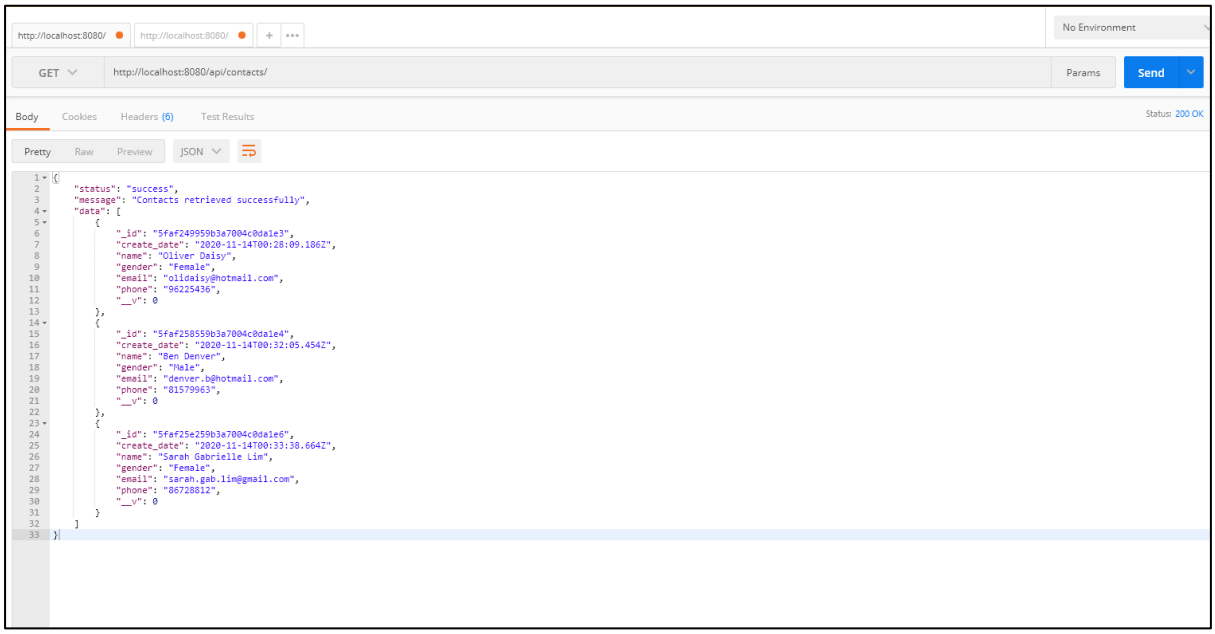
To update a particular Contact record, we call the (`/api/contacts/:id`) api with a PUT request as shown below. In this example, we are updating John Doe's contact details to change his name, email and phone number. To view a particular Contact record, we call (`/api/contacts/:id`) api with a GET request.



To delete a particular Contact record, we call (`/api/contacts/:id`) api with a DELETE request. In this example, we have deleted the record that we have just updated.



So, if you call the GET request on the list of contacts again, there should be one less record now as shown below.

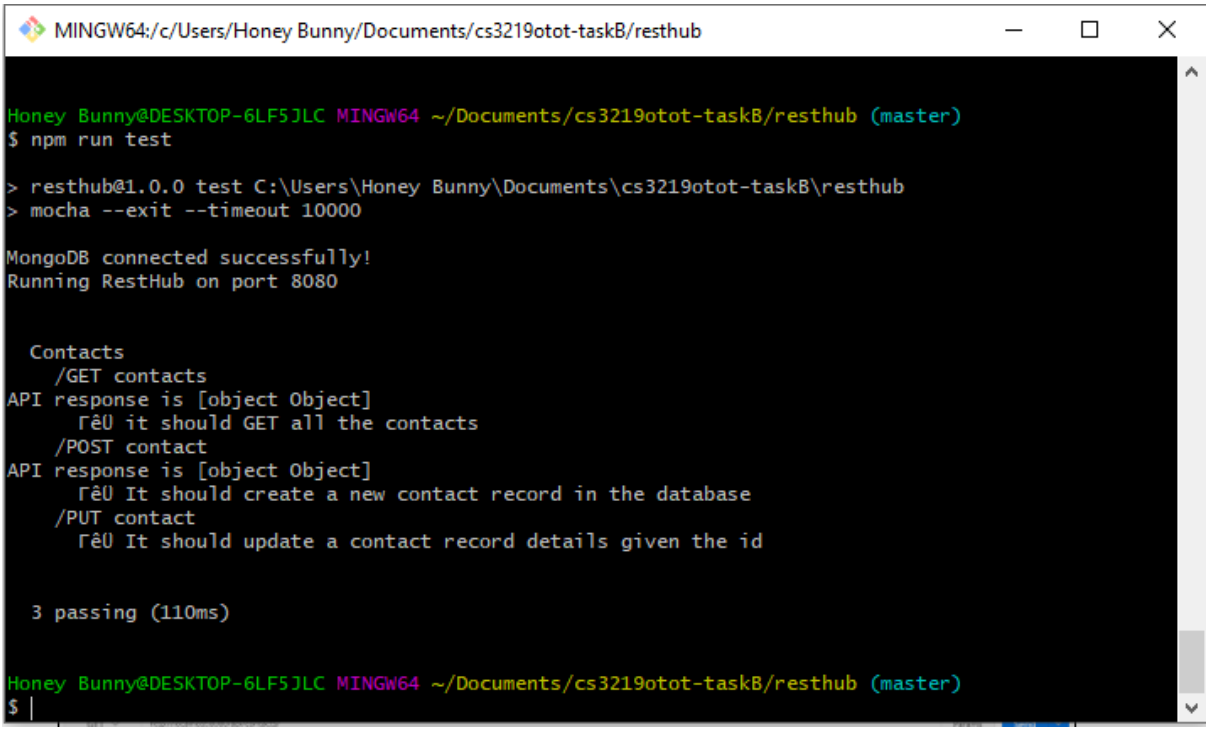


Run tests locally and via Travis (B2):

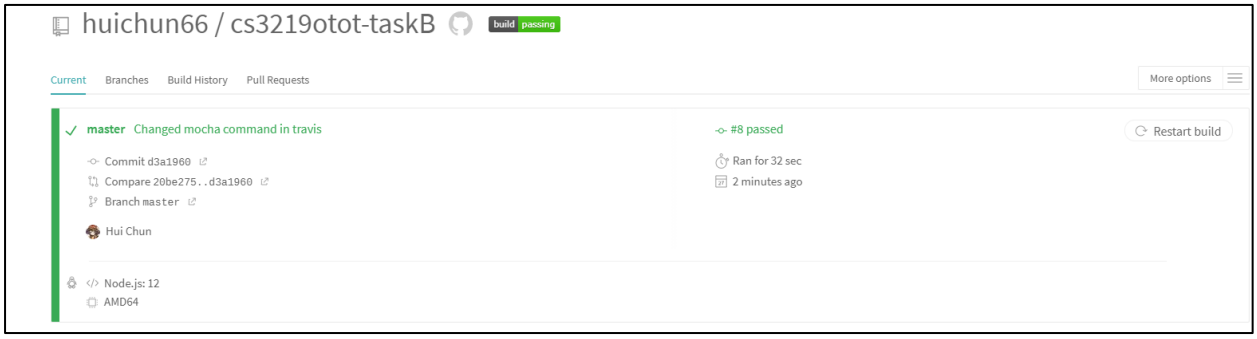
```
cd resthub

npm run test
```

Run the above commands to run the tests locally for testing the GET/POST/PUT api calls. You should see a similar output as below:



The following screenshot shows the same tests being run remotely on Travis CI.



```
53 npm WARN resthub@1.0.0 No repository field.
54 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/fsevents):
55 npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
56
57 audited 202 packages in 1.081s
58
59 17 packages are looking for funding
60   run `npm fund` for details
61
62 found 0 vulnerabilities
63
64 The command "npm install" exited with 0.
65 $ echo 'Running Tests'                                                    0.00s
66 Running Tests
67 The command "echo 'Running Tests'" exited with 0.
68 $ npm run test                                                           0.73s
69
70 > resthub@1.0.0 test /home/travis/build/huichun66/cs3219otot-taskB/resthub
71 > mocha --exit --timeout 10000
72
73 MongoDB connected successfully!
74 Running RestHub on port 8080
75
76
77   Contacts
78     /GET contacts
79 API response is [object Object]
80       ✓ it should GET all the contacts
81     /POST contact
82 API response is [object Object]
83       ✓ It should create a new contact record in the database
84     /PUT contact
85       ✓ It should update a contact record details given the id
86
87
88   3 passing (83ms)
89
90 The command "npm run test" exited with 0.
91 $ echo '******Build Successful!******'                                   0.00s
92 ******Build Successful!******
93 The command "echo '******Build Successful!******'" exited with 0.
94 store build cache                                                        cache.2
96
97
98 Done. Your build exited with 0.
                                                                            Top
```

Access deployed APIs (B3):

After deploying the application to AWS Lambda via Travis CI, this is what the Travis Build log will look like:

```
102                                                                         4.35s
103 Installing deploy dependencies                                          dpl.1
104 Fetching dpl-lambda-1.10.15.gem
105 Fetching jmespath-1.4.0.gem
106 Fetching aws-eventstream-1.1.0.gem
107 Fetching aws-sigv4-1.2.2.gem
108 Fetching aws-sdk-core-2.11.627.gem
109 Fetching aws-sdk-resources-2.11.627.gem
110 Fetching aws-sdk-2.11.627.gem
111 Fetching rubyzip-1.3.0.gem
112 Successfully installed jmespath-1.4.0
113 Successfully installed aws-eventstream-1.1.0
114 Successfully installed aws-sigv4-1.2.2
115 AWS SDK For Ruby V2 has been marked as deprecated. Please upgrade to AWS SDK For Ruby V3.
116 Successfully installed aws-sdk-core-2.11.627
117 AWS SDK For Ruby V2 has been marked as deprecated. Please upgrade to AWS SDK For Ruby V3.
118 Successfully installed aws-sdk-resources-2.11.627
119 AWS SDK For Ruby V2 has been marked as deprecated. Please upgrade to AWS SDK For Ruby V3.
120 Successfully installed aws-sdk-2.11.627
121 Successfully installed rubyzip-1.3.0
122 Successfully installed dpl-lambda-1.10.15
123 8 gems installed
124
125 Preparing deploy                                                        dpl.2
126 Using Access Key: ****************PIPV
127
128 Deploying application                                                   dpl.3
129 Version 2 of the Ruby SDK will enter maintenance mode as of November 20, 2020. To continue receiving service updates and new features, please upgrade to Version 3. More information can be found
    here: https://aws.amazon.com/blogs/developer/deprecation-schedule-for-aws-sdk-for-ruby-v2/
130 Function helloworld already exists, updating.
131 Updated configuration of function: helloworld.
132 Updated code of function: helloworld.
133
134 Done. Your build exited with 0.
                                                                            Top
```

Similarly, this is what my AWS Lambda Functions console panel will look like once it is deployed successfully: