Author: Sragvi ZCC - ReadMe

How to Run:

1. Initial step would be to start the server which connects to the Zendesk api and helps us in retrieving the data.

For this to happen, go to the Server/api folder

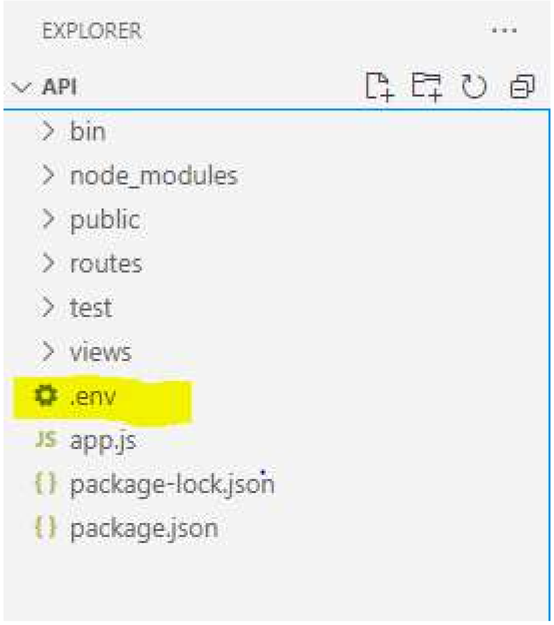
All the node modules are already present in the directory, but just to make sure that everything is fine, run the following command:

```
npm install
```

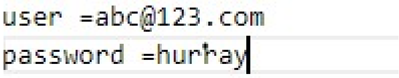
This will install all the required modules.

As we have to pass our username and password along with our get request, create a file named .env in the api folder.

A snippet of at which level the file should be created is share below:



After creating the file, the contents of the file must contain the username and password. A sample content is shared below:



```
user =abc@123.com
password =hurray
```

Note: Please make sure that no extra spaces are being added, either at the start or end of the credentials.

Now, to start the server run the following command:

```
npm run start
```

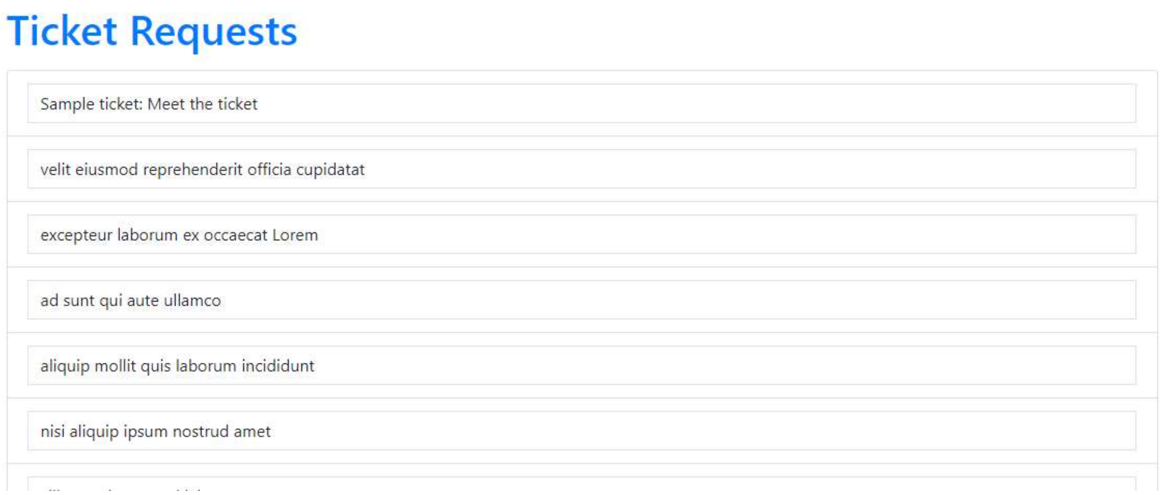
You can see a console log statement 'Listening at 9k' on the VSCode terminal or whatever is the terminal that is being used once the server is up and running.

2. Now we need a UI to showcase the data, so go to the Frontend/my-api folder and run the following commands:

```
npm install

npm run start
```

3. This will trigger a request to the 9000/server url which in turn sends a request to the api, requesting for the data. After a successful retrieval the page looks like this



Ticket Requests

Sample ticket: Meet the ticket

velit eiusmod reprehenderit officia cupidatat

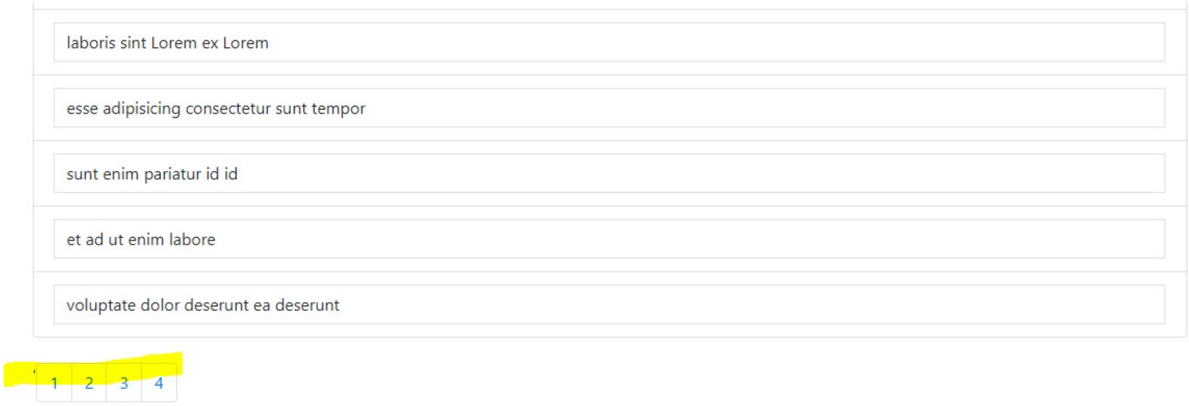
excepteur laborum ex occaecat Lorem

ad sunt qui aute ullamco

aliquip mollit quis laborum incididunt

nisi aliquip ipsum nostrud amet

At the end of the page, you can see a small navigation menu that helps us in going through all of the ticket requests.



laboris sint Lorem ex Lorem

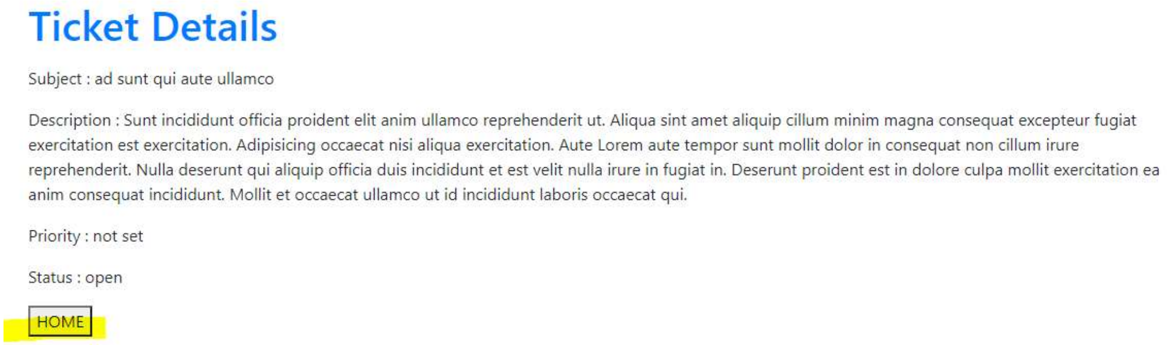
esse adipisicing consectetur sunt tempor

sunt enim pariatur id id

et ad ut enim labore

voluptate dolor deserunt ea deserunt

You can select on any of the tickets and then we show a detailed info about the ticket selected and, on that page, there is a button called Home, which takes you back to the list of ticket requests.



Ticket Details

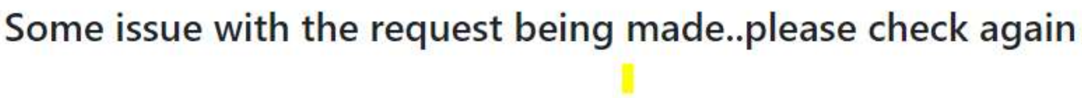
Subject : ad sunt qui aute ullamco

Description : Sunt incididunt officia proident elit anim ullamco reprehenderit ut. Aliqua sint amet aliquip cillum minim magna consequat excepteur fugiat exercitation est exercitation. Adipisicing occaecat nisi aliqua exercitation. Aute Lorem aute tempor sunt mollit dolor in consequat non cillum irure reprehenderit. Nulla deserunt qui aliquip officia duis incididunt et est velit nulla irure in fugiat in. Deserunt proident est in dolore culpa mollit exercitation ea anim consequat incididunt. Mollit et occaecat ullamco ut id incididunt laboris occaecat qui.

Priority : not set

Status : open

If there is an error in connecting to the server, we display few error messages based on the error type. Below is a snapshot of the page, when we come across an issue.



Some issue with the request being made..please check again

Unit Testing: For both Server/api and Frontend/my-api few basic good flows UT's have designed.

Run the following command in the respective folders to have the UT's run:

```
npm run test
```

What can be improved:

- We can handle the errors in a better way.
- Increase the UT coverage.
- Add functionality to cover more real-time scenarios.