

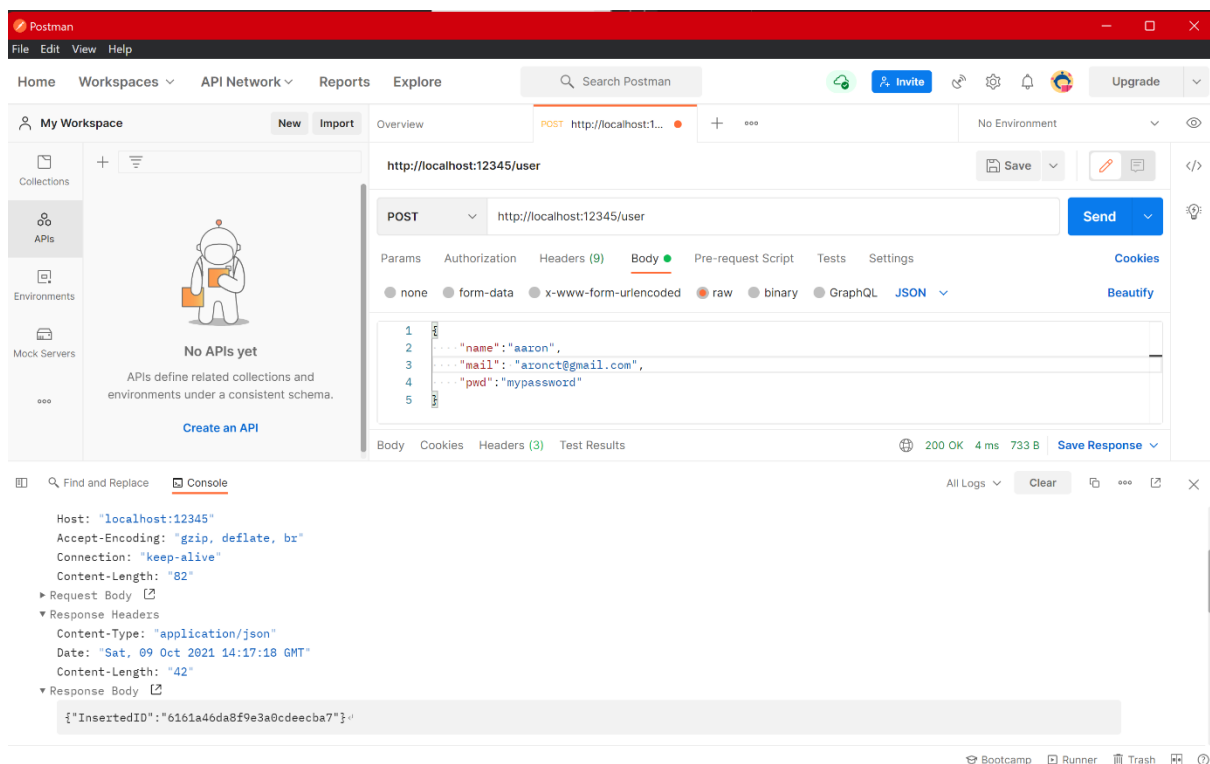
Appointy 2 month internship Tech Task Documentation

Buddhi Rohan Sai

19BCE2041

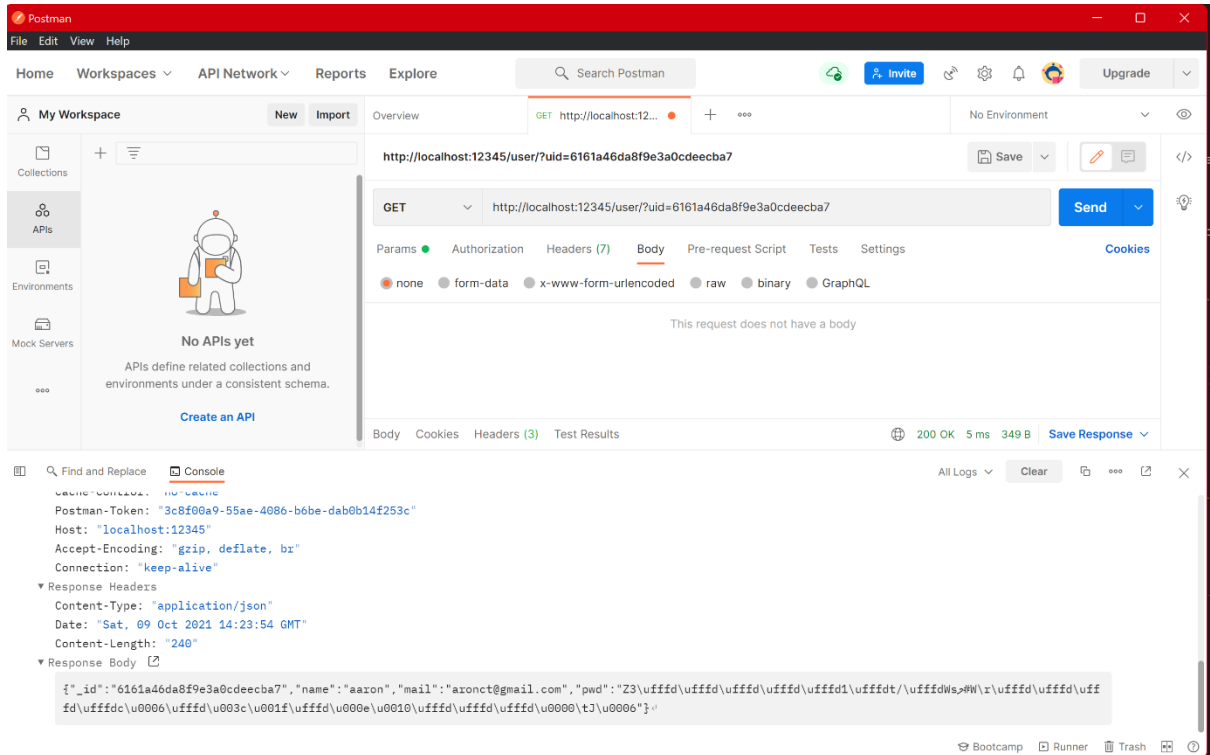
Create a user

The user is created by the function `CreateUserEndpoint` which takes in the http response writer object and the http request pointer. The request is checked to be a POST request. The JSON request body is decoded into the user type object and then the password obtained from the user is encrypted using the AES encryption algorithm found in the `crypt` library of the `GOLANG` std library and then the user data is stored in the mongo db in database `appointy` in collection called “users” this way the user password is safe.



Getting user using ID

We use the url parameter as “`?uid =`” to get the user required. The uid is obtained from URL using the `request.URL.query()`. The user is then matched by using the `collection.FindOne` by comparing `User{ID:id}` in the `appointy` database in the `users` collection.

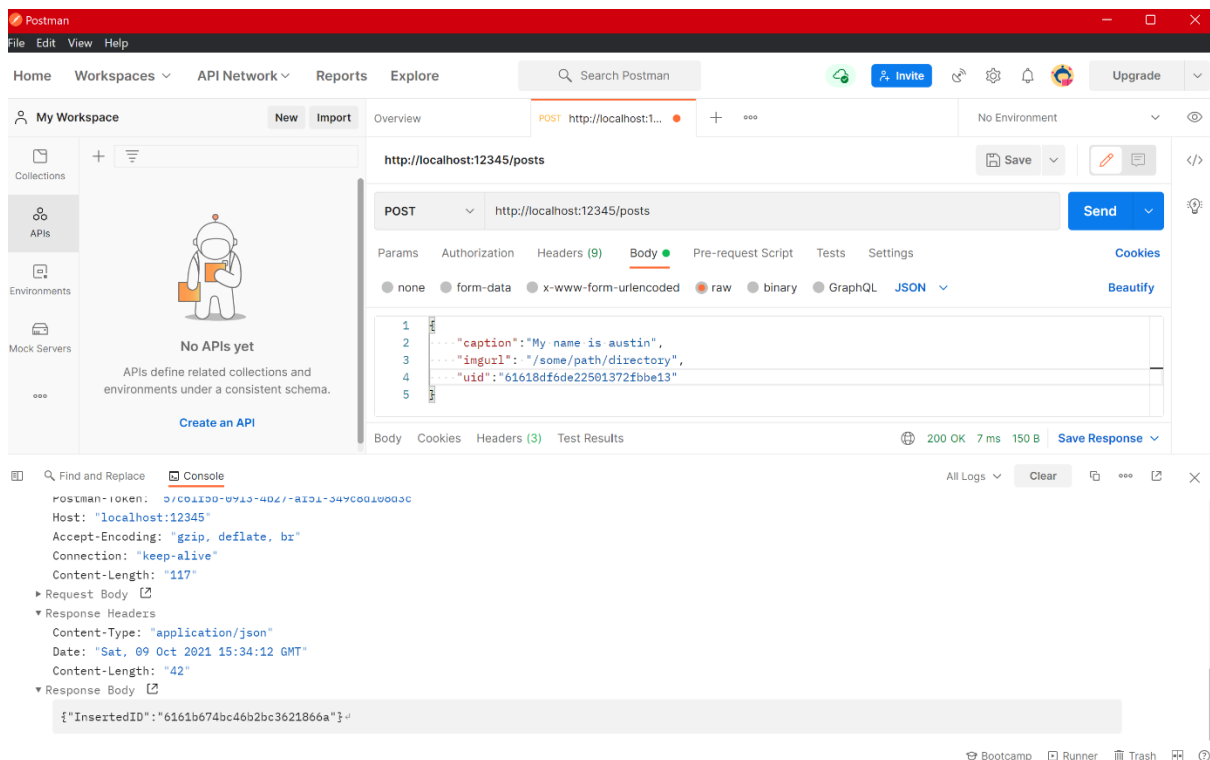


Create a post

The post function is created in a similar way to the createuser function. The posts also have the user id of the one who posted the post as a string to enable us to query all the posts of a user with some userid later. Each Post has a postid, caption, url and userid associated with it.

Get Post using id

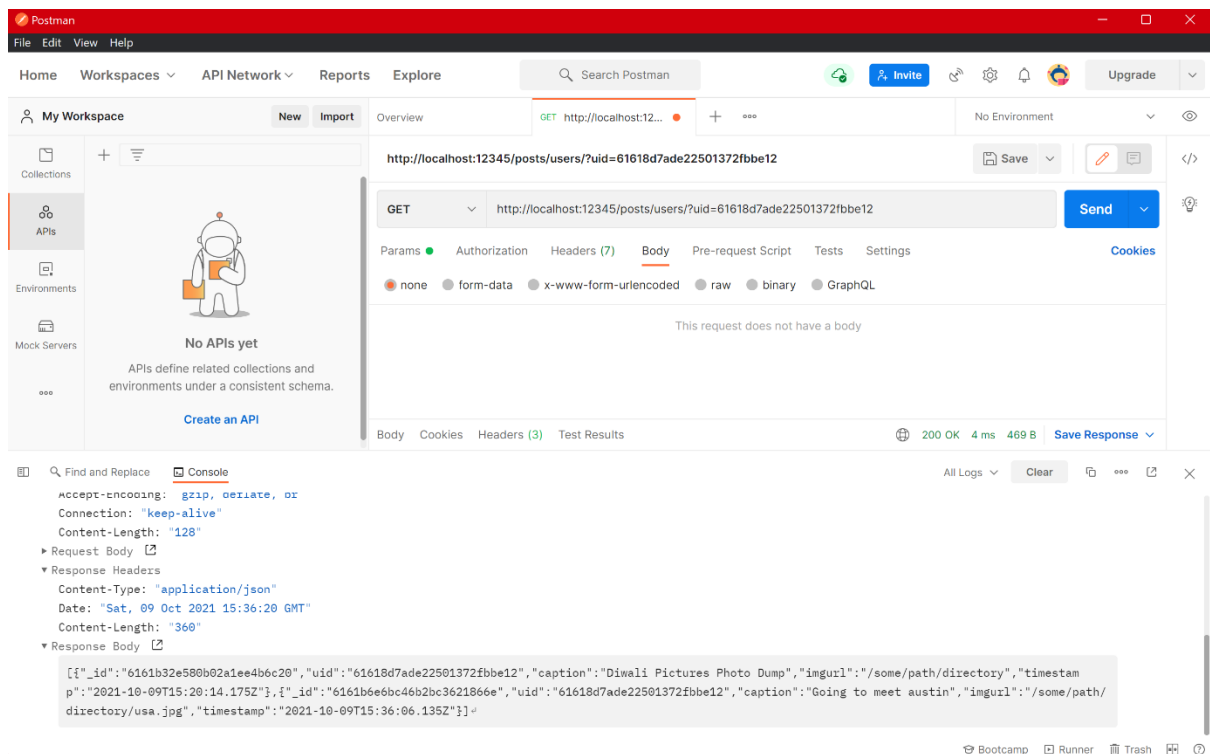
Again a similar approach to getting a user with userid was followed here by using “?pid=...” As the url parameter to specify post id in the api.



List all posts of a user

The user id of the user is taken through the url using “?uid=...”

Here, we query all the posts there are in the database and then we decode and check which of the posts have user id which match the user id specified in the url parameter (uid). These are appended to a list which contains objects of type Post and then this is encoded to the JSON



The API was tested by using the POSTMAN desktop app. All the 5 APIs successfully ran with the local mongodb server at `mongodb://127.0.0.1:27017/`.

The postman screenshots are also attached for reference.