Machine Test for Node.js Developer || Code Brew Labs||

**Problem Statement:**

**Create a server instance and create a route for the following functions**  
A.) There is a user who can create a booking for any type of filling like gas, diesel, petrol. As per the booking you have to provide the nearest pump details who is suitable for the booking.(For example, like Zomato and Uber eats they provide the nearest restaurants according to user location)  
 Note\*\* Booking can have multiple vehicles with different types of filling.  
  
**B.)**Pump can view the bookings as per the data with user and person details who did the filling. Here user details would be shown only when the user is not deleted or blocked. So make database schema accordingly to manage user statuses.  
  
**C.)**A route For file upload with a limit of 3mb.The file could be of service station where filling is happening or for a user profile pic.  
  
**D.)** You Should Have an AUTH Function/strategy which is required to authenticate. here you should have both token and API Key for Authorization.

      The API key must be placed in .eve file and accessed when required.

**Solution:**

**Github URL:** [**https://github.com/Iaakashgarg/StationBookingApp**](https://github.com/Iaakashgarg/StationBookingApp)

**Postman Collection Link:** [**https://www.getpostman.com/collections/b2d0871a52cc2949d6e8**](https://www.getpostman.com/collections/b2d0871a52cc2949d6e8)

**How to run the application?**

**To install dependencies:**

Go to root folder and execute below command:

* **npm install**

To start the application, go to the “**CodeBewLabs\_**” folder and execute the ‘server.js’ script which is the entry file of the application using below command (execute in root path).

* **node server.js**

Note: You need to change database configuration or JWT token in .env file, if required.

The given folder contains three sub-folders

1. **CodeBewLabs\_:** This folder contains the actual Nodejs code for starting the application.
2. **codebrewDB**: This folder contains the dummy databases which can be used to check the available functionality of the application.

Note: You need to have postgresql downloaded and make sure you have database

with name “**codebrewDB**”

After making the database import the given dummy database.

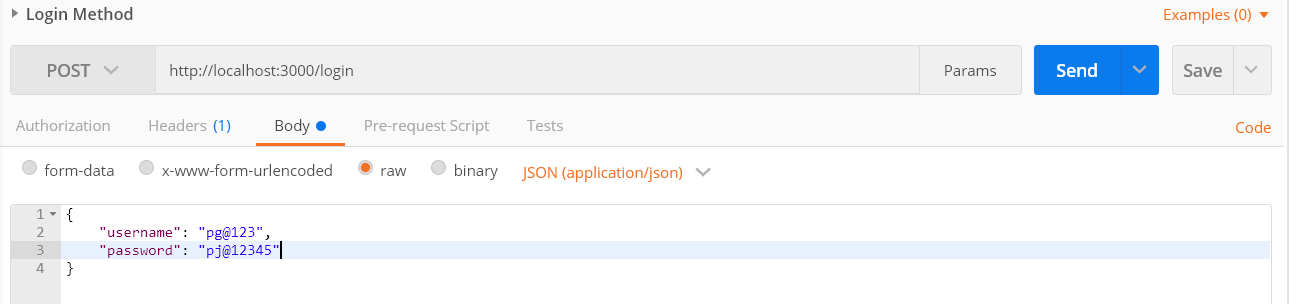
1. Postman Collection

**Relevant Assumptions:**

* Express is used as server.
* Database: Postgresql
* PostGIS Extension is used for location functionality.
* Dotenv is used for environment variables.
* Jsonwebtoken for JWT Authentication.
* Multer middleware is used for file management
* File is stored in project folder and link of file is stored in database. (file location will be FTP location or cloud deployment service location for real life scenario).
* Each station is considered to provide all three facilities.
* User can book any number of services for different vehicles.

**Login Method**

API Call:

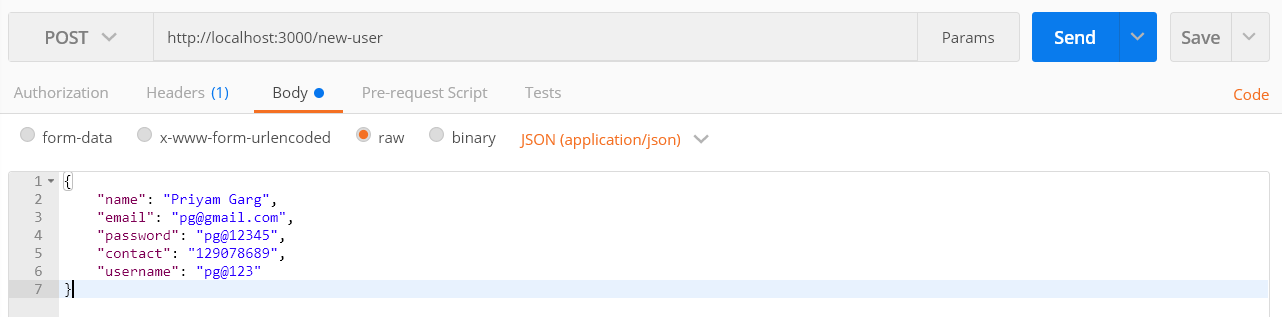


Response:

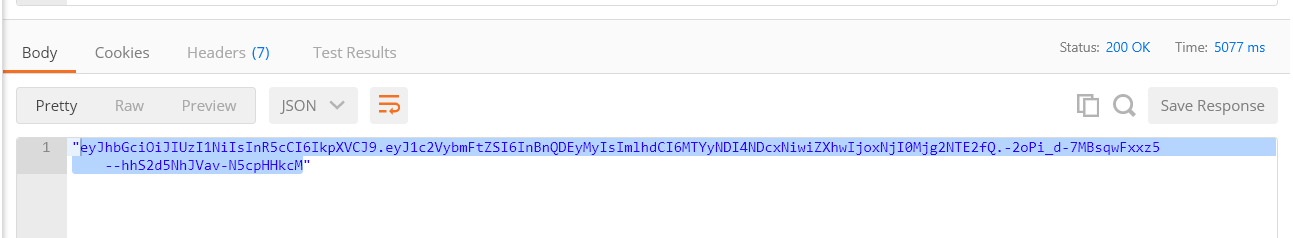


**Add new user Method**

API Call:

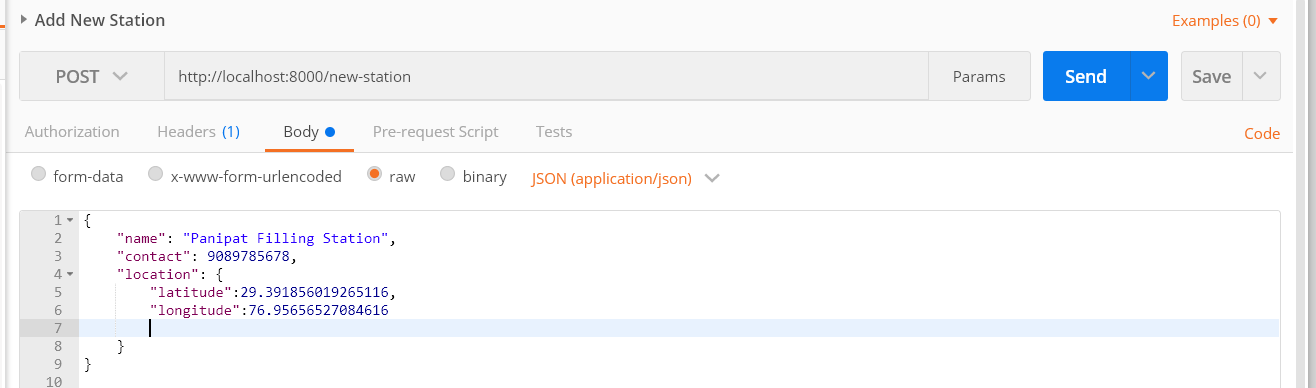


Response:

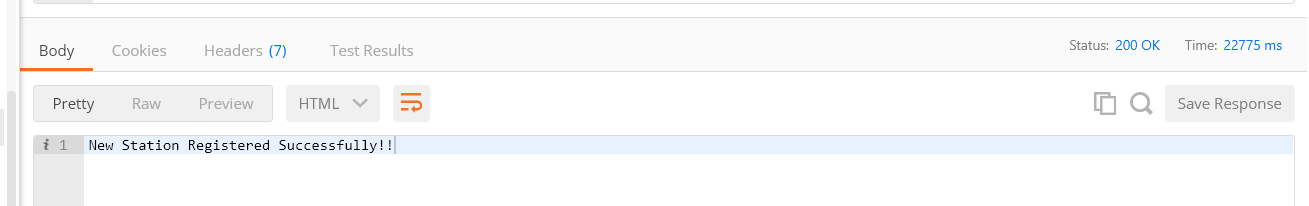


**Add new station Method**

API Call:

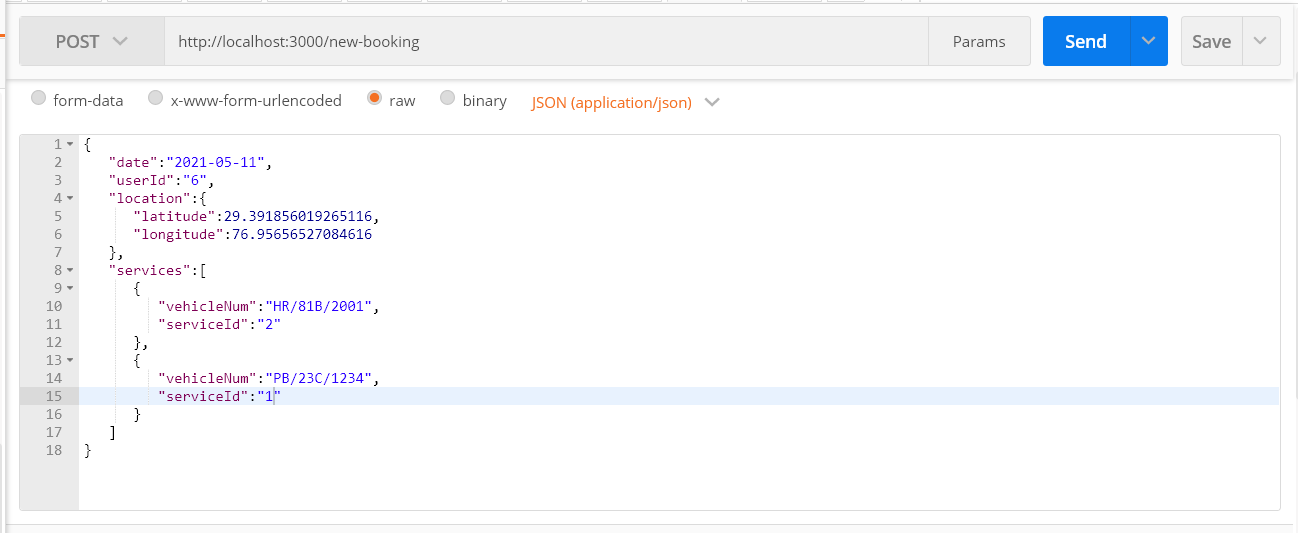


Response:

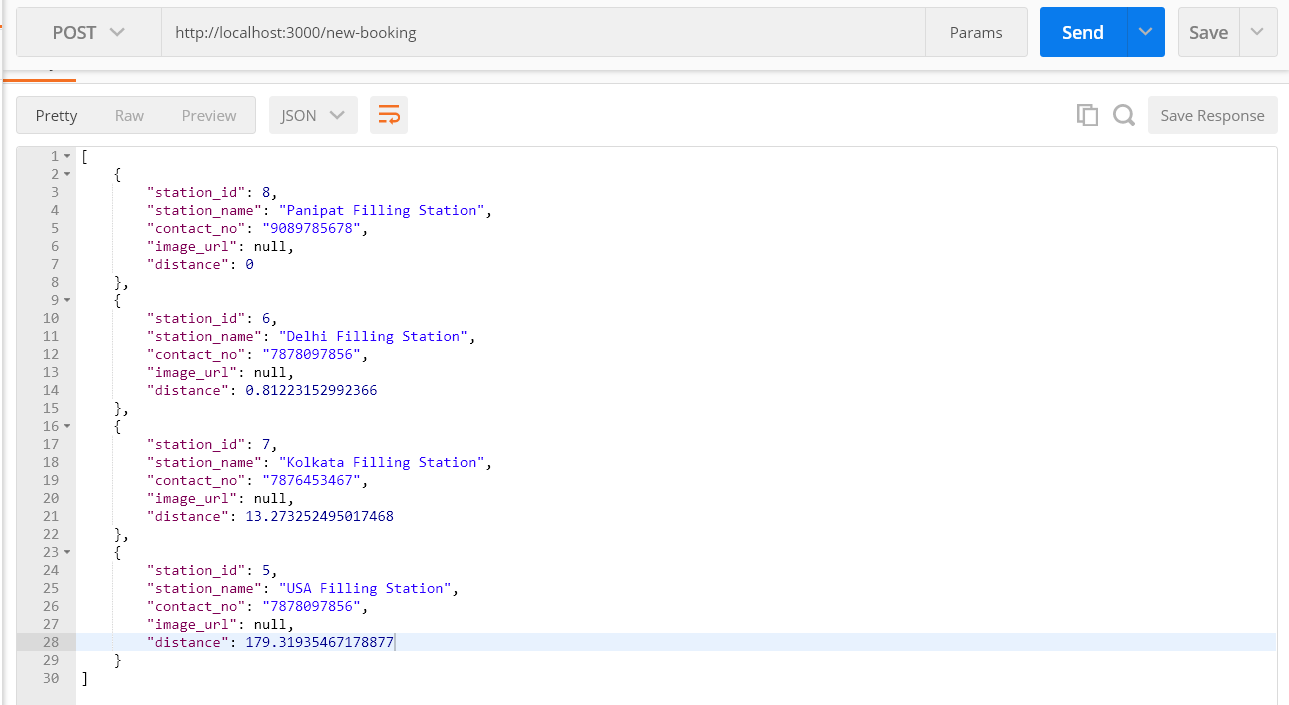


**New Booking Method:**

API Call:

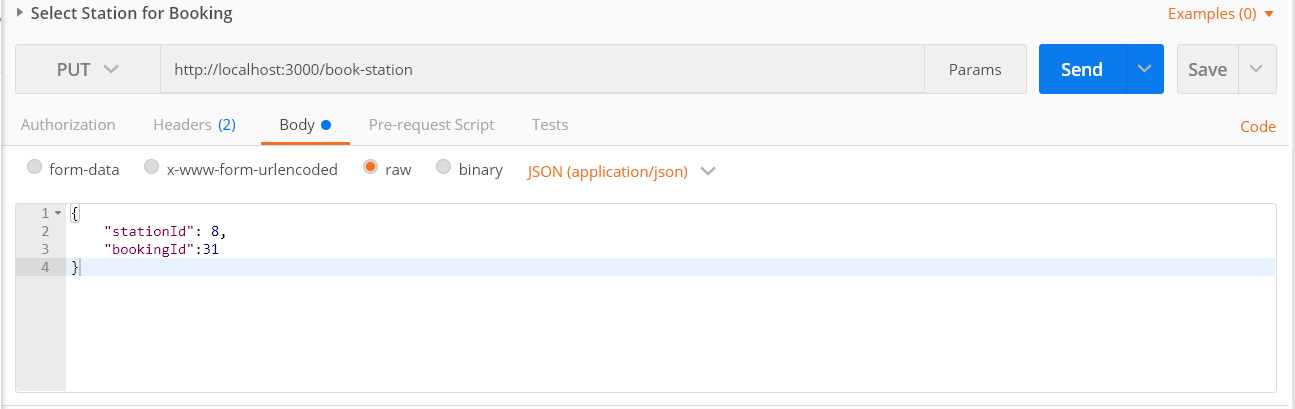


Response:

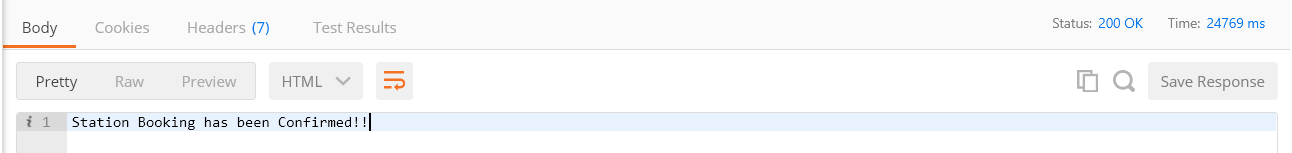


**Select Station Method:**

API Call:

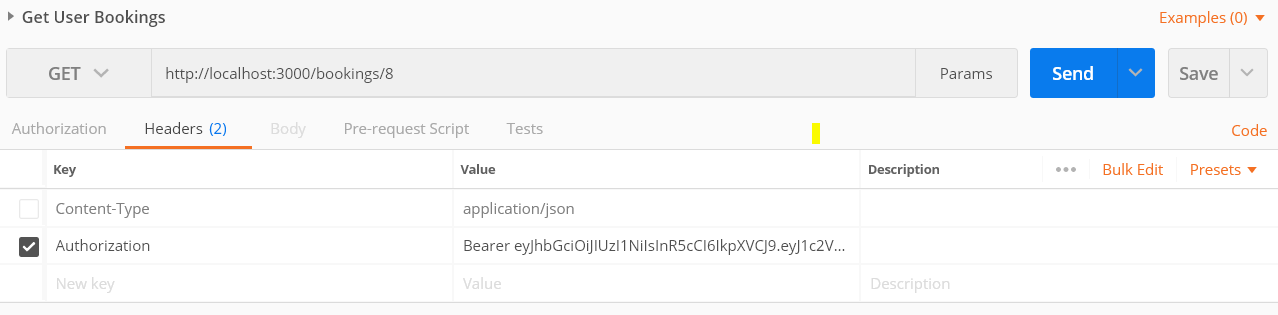


Response:

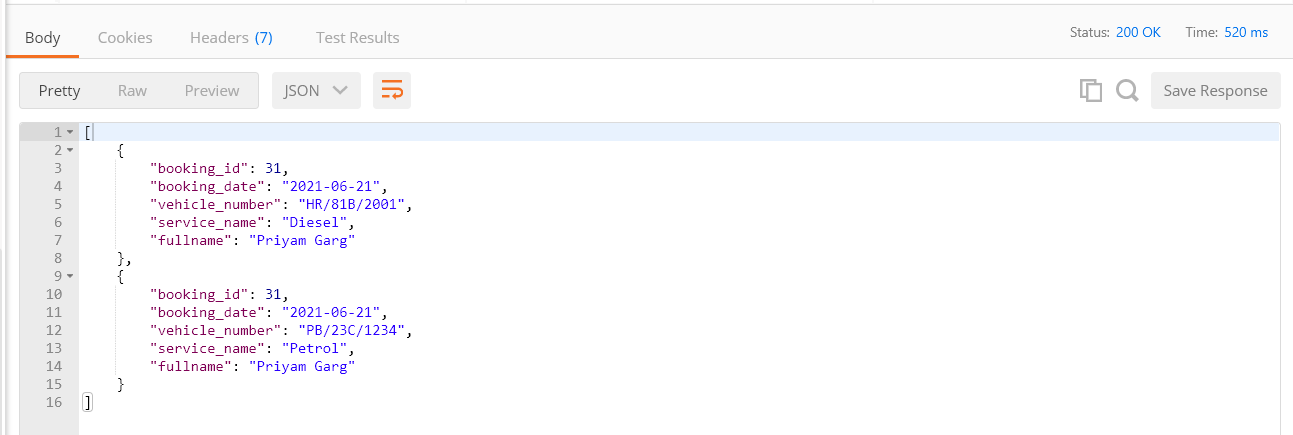


**Get User Bookings:**

API Call:

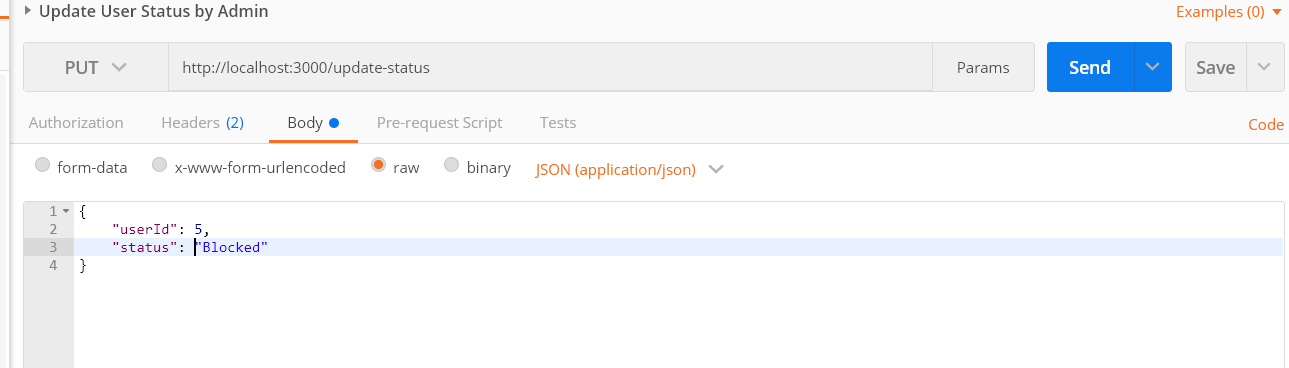


Response:

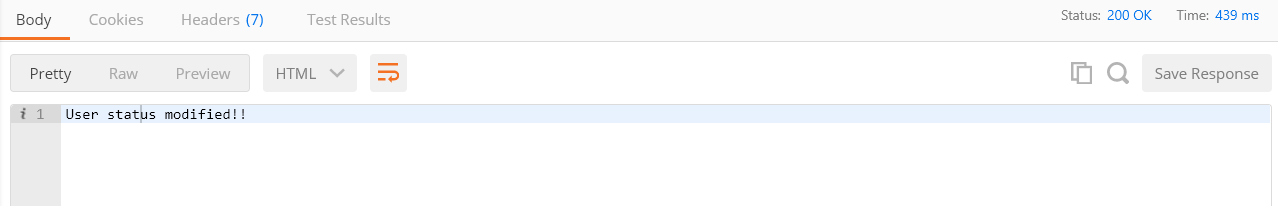


**Update User Status:**

API Call:

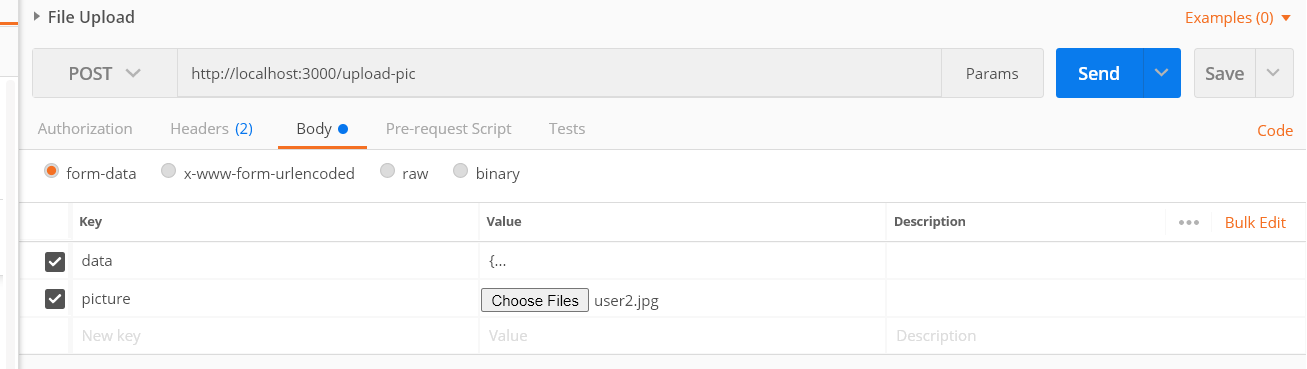


Response:

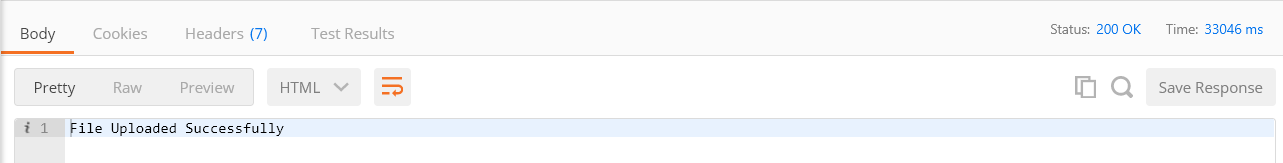


**File Upload Method:**

API Call:

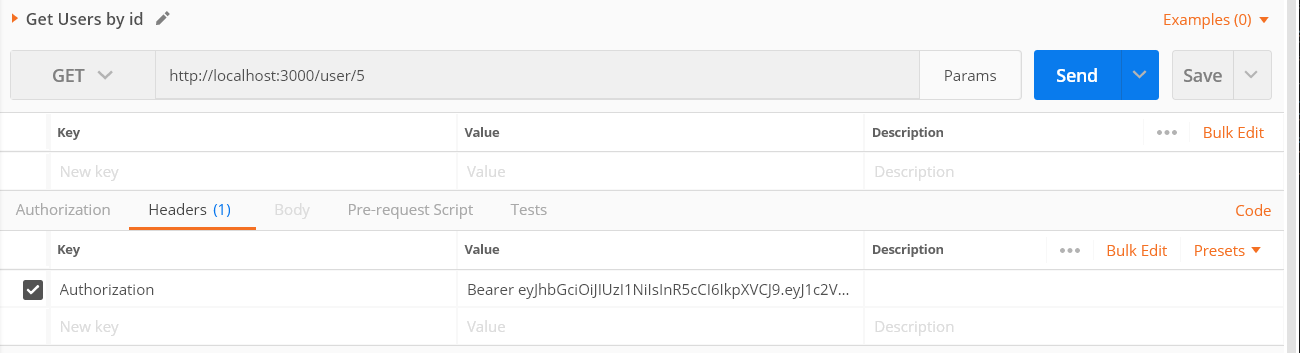


Response:

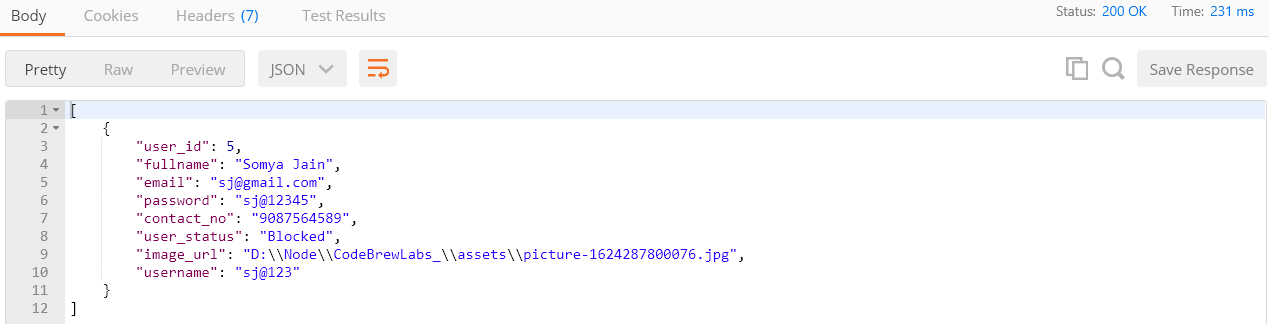


**Get User By Id:**

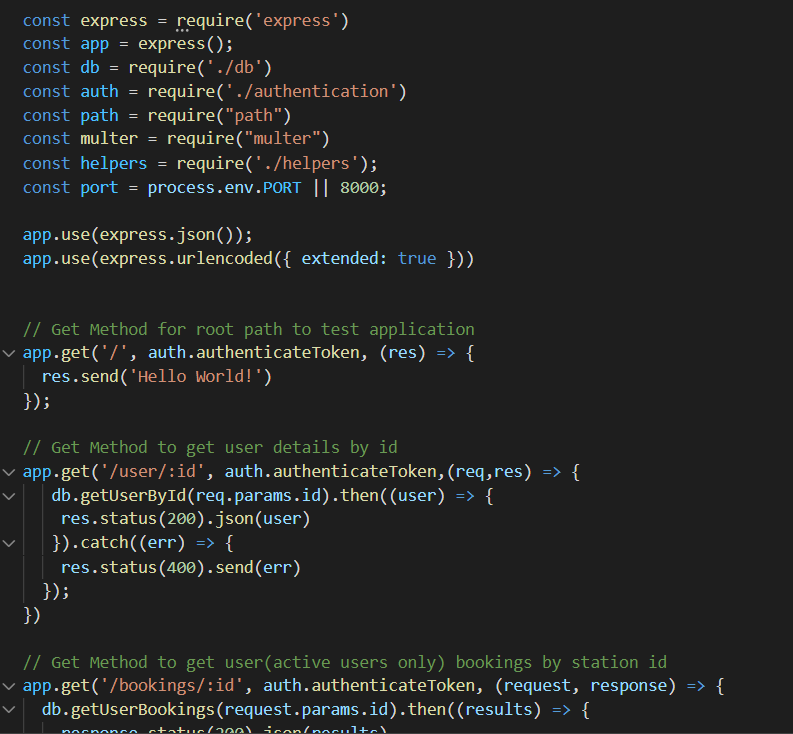
API Call:

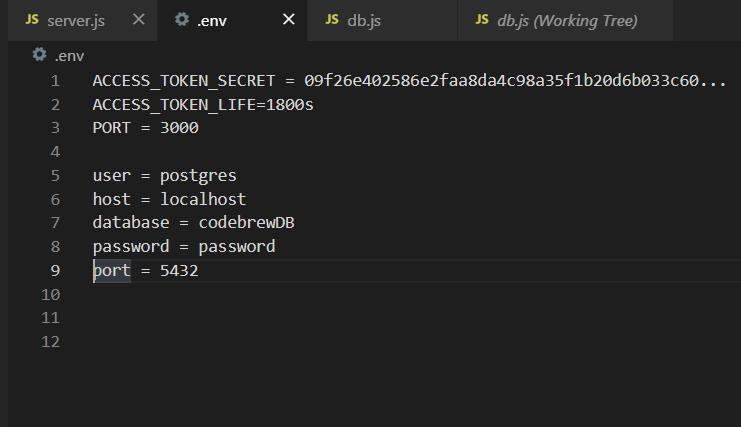


Response:



**Snapshots:**







**Folder structure:**

