Java framework used: NodeJS

Tools required to test: Postman

NPM dependencies used: Express, Postman-Request, Jest

End points available:

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. no | HTTP Method | Endpoint | Comments |
| 1 | POST | https://vinay-blog-application.herokuapp.com/user/login | In request body, set type to Json and add  { “admin”: “true” } to login as admin |
| 2 | POST | https://vinay-blog-application.herokuapp.com/user/logout |  |
| 3 | GET | https://vinay-blog-application.herokuapp.com/users |  |
| 4 | GET | https://vinay-blog-application.herokuapp.com/users/:id | :id is the user ID. Valid ID’s from 1 to 10 |

Code repository location:

The source code is available at: <https://github.com/vinscode/node-assignment>

To execute the code:

If Node is not installed: I have deployed code in Heroku server, please use this link <https://vinay-blog-application.herokuapp.com> in postman to run the endpoints.

To run the code manually use the below commands:

1. Install [Node.js](https://nodejs.org/)
2. Clone the github repository to somewhere on your hard drive
3. Open a command prompt and go to that directory
4. Type “npm install” to install all dependencies
5. Type “npm start”
6. Use https:localhost:3000/ in postman to run the endpoints

Assumptions and details:

All the below end points are implemented in /src/app.js file

1. POST - /user/login:

* This end point allows user to login to the application. I have created a simple function that sets admin variable to true.
* Setting admin = true in the request body is considered as user is an admin and any other value for admin is considered as non admin user e.g. admin = false, admin = dummy

Graphical user interface, text, application, email

Description automatically generated

1. POST - /user/logout:

* This will set the admin flag to false

Graphical user interface, text, application, email

Description automatically generated

1. GET - /users:

* This end point returns all the users using the blog app.
* Authenticates if the user is an admin
* Consumes <https://jsonplaceholder.typicode.com/users> API
* Returns Json data

Graphical user interface, text

Description automatically generated

1. GET - /users/:id

* This end point returns specific user details along with their posts by passing the user id variable “:id”
* E.g. <https://vinay-blog-application.herokuapp.com/users/1>, <https://vinay-blog-application.herokuapp.com/users/6>
* Authenticates if the user is an admin
* IDs are supposed to be numbers. Else, error message is thrown
* Consumes <https://jsonplaceholder.typicode.com/users> and <https://jsonplaceholder.typicode.com/posts> API’s
* Returns Json data

Text

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

Test cases:

* I have created /tests/user.test.js file that include all test cases.
* Used Jest framework and supertest library to test all the HTTP end points.
* To run test cases, run “**npm test”** command from the command prompt

Text

Description automatically generated