

Backend Assignment

You need to create Backend RESTful APIs in **NodeJS** with any framework like **ExpressJS/NestJS** and **MongoDB** Database.

This service should have a user who will be able to signup and login. Only The authenticated (logged in) user will be able to perform actions like creating an Article, Retrieving all Articles and updating their own user profile. You can set the expiry of the user session according to you.

- User Resource will have email, password, name and age. Email will be unique.
- Article Resource will have a title and description. One Article can have only one author (user)

1. Api to signup a user with email and password.

Endpoint : api/signup

NOTE:

- a. Apis should have basic validations on the request inputs, ensuring no Duplicate Users and Proper error responses.
- b. Password should be stored in an encrypted format. Use any encryption libraries like bcrypt

2. API to login a user with email and password.

Endpoint: api/login

NOTE:

- a. Apis should have basic validations on the request inputs and Proper error responses.
- b. API should return JWT auth token in the response.
- c. Use JWT token to maintain for user authentication of resources and apis in next features.

3. API to create an Article.

Endpoint: api/users/:userId/articles

NOTE:

- a. This protected Api should be authenticated using the JWT in the authorization header of the request.
4. API to get all articles

Endpoint: api/articles

NOTE:

- b. Every article in the response should contain the user info who created them, respectively
 - c. This protected Api should be authenticated using the JWT in the header of request.
5. API to update user profile. Only name and age are editable.

Endpoint: api/users/:userId

Points to Remember:

- a. Standard response format of the APIs

```
{
  statusCode:
  data: {
    data: //response data
  },
  error: //if any exists
  message:
}
```
- b. All requests and responses are in JSON format

Evaluation of the Assignment will be based on:

- 1. Following REST Principles like proper request methods, headers, etc.
- 2. Schema Design (Scalability, maintainability)
- 3. Implementation of features acc. to the Instructions

Note: You need to publish your codebase to a GitHub repository and share the link to krishan@sploot.tech. Also, please host your server on any platform like netlify/vercel.

In case of any queries, please contact krishan@sploot.tech

