Back End Development 1

Sesi 23

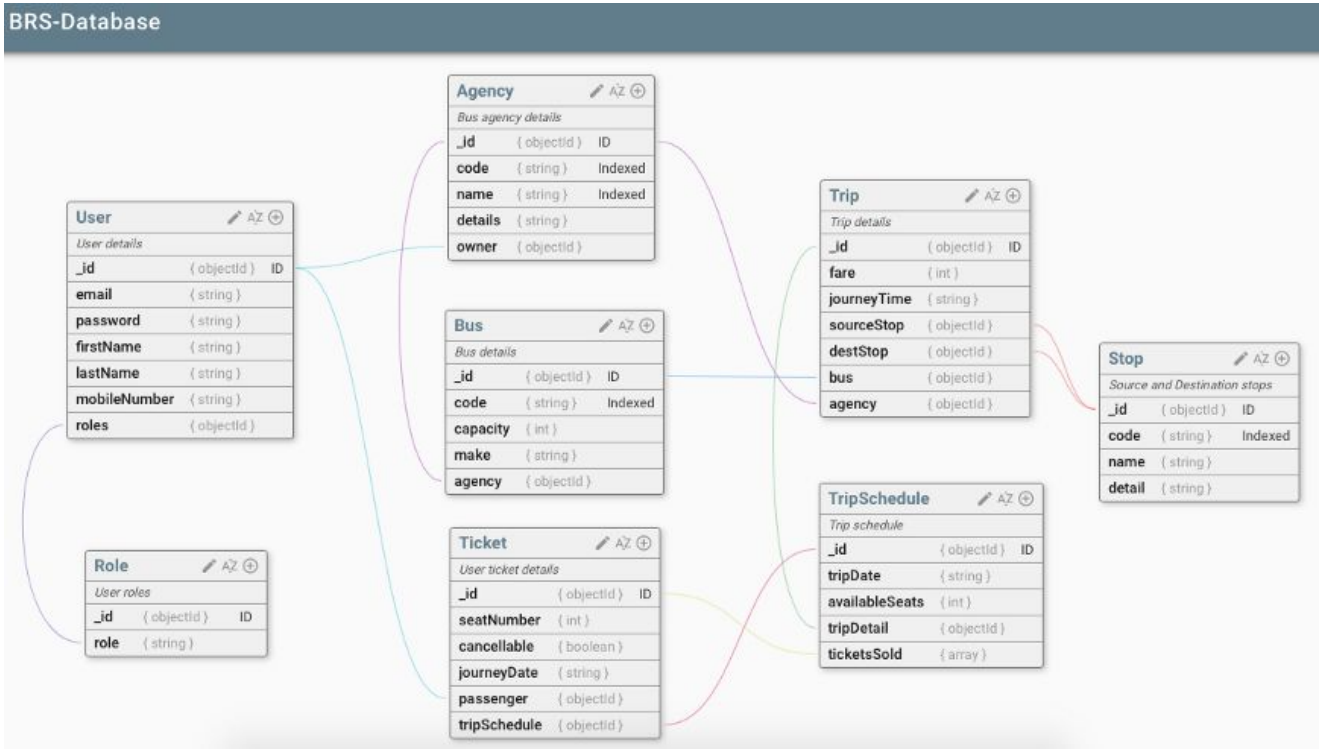
FINAL PROJECTS +

Introduction

The Admin user can access this application on browser (laptop or mobile/tablet, doesn't really matter as this is built using bootstrap, material design and is completely responsive) and can perform the following actions :

1. Signup
2. Login (and get a JWT token)
3. List all available stops
4. Search a trip between any two stops
5. Filter search results with a date option
6. Book a ticket for a given trip schedule

Database Schema



- The authentication and authorization is governed by User and Role collection.
- The Agency collection keeps the details of agency along with who owns it.
- The Stop collection keeps the data about all the stops in the system.
- The Bus collection has the data of all the buses for various agencies along with their capacity and make years.
- The Trip and TripSchedule collections keep the information about all the trips that agency owners create within the system. Information like source and destination stops, journey time, data of travel, tickets sold so far and the available seats is collected in them.
- Last, the Ticket collection keeps information about all the tickets issued in the application for various trips.

Library/Tools Used

Following libraries were used during the development of this Final Project as your starter kit :

- Spring Boot : Server side framework
- MySQL : mySQL database
- JWT : Authentication mechanism for REST APIs
- Thymeleaf : Templating Engine
- Material : UI Themeing / design
- Bootstrap : CSS Frameworks
- JWT : Authentication mechanism for REST APIs

API

http://localhost:8080

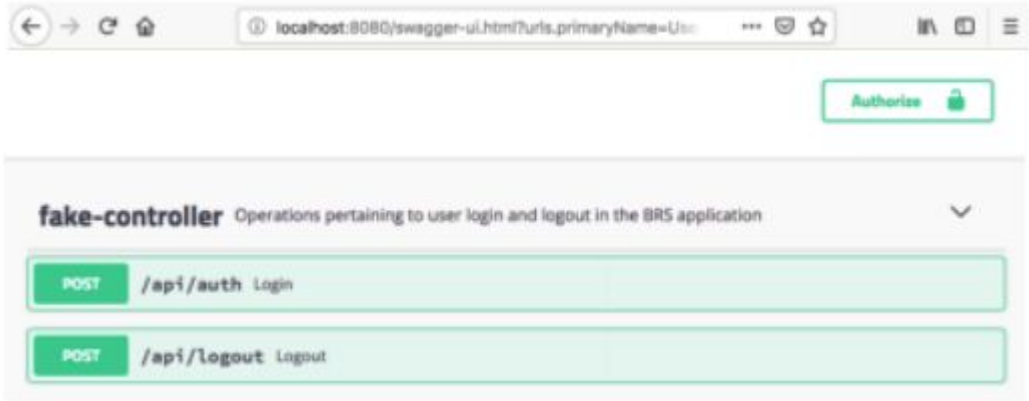
And the REST APIs can be accessed over the following base-path :

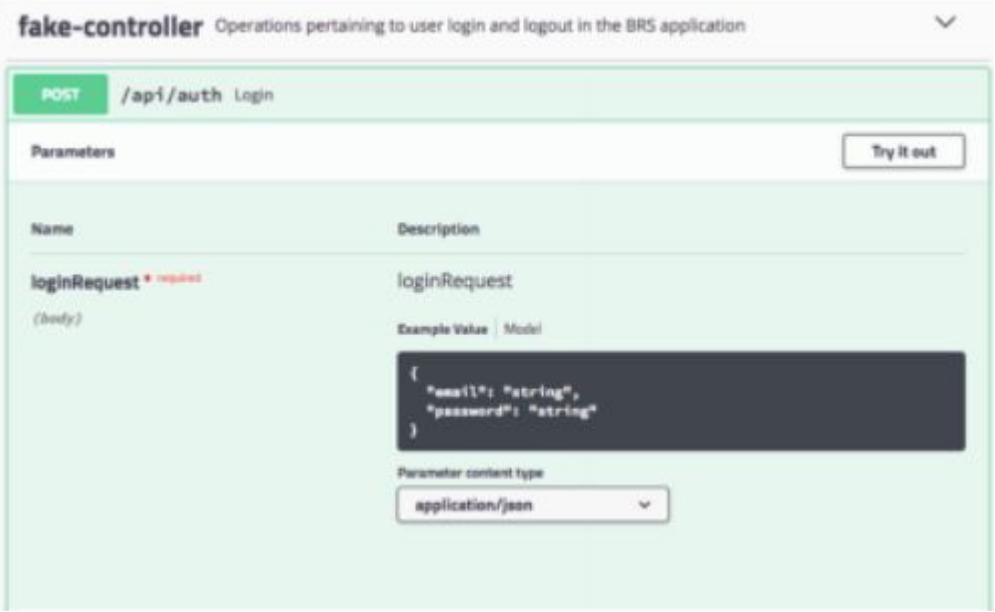
http://localhost:8080/api/

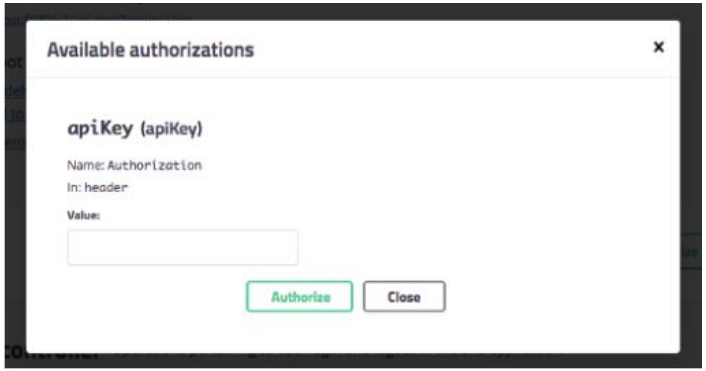
Some of the important api endpoints are as follows :

- http://localhost:8080/api/v1/user/signup (HTTP:POST)
- http://localhost:8080/api/auth (HTTP:POST)
- http://localhost:8080/api/v1/reservation/stops (HTTP:GET)
- http://localhost:8080/api/v1/reservation/tripsbystops (HTTP:GET)
- http://localhost:8080/api/v1/reservation/tripschedules (HTTP:GET)
- http://localhost:8080/api/v1/reservation/bookticket (HTTP:POST)

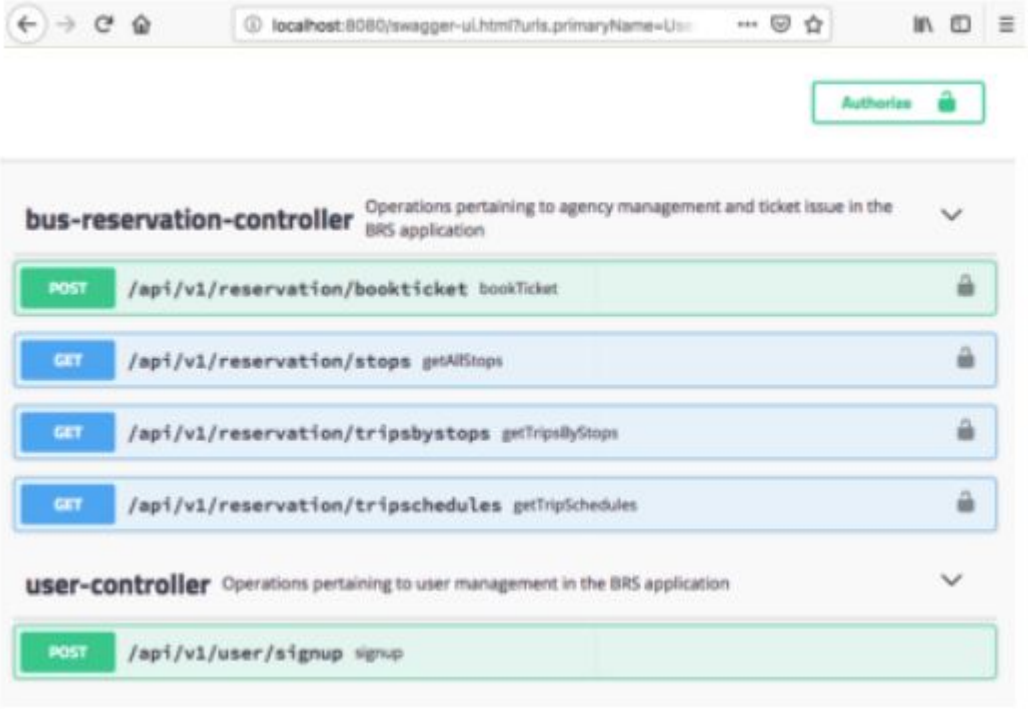
Output


localhost:8080/swagger-ui.html?urls.primaryName=Use
Authorize
fake-controller
Operations pertaining to user login and logout in the BRS application
POST
/api/auth
Login
POST
/api/logout
Logout


fake-controller
Operations pertaining to user login and logout in the BRS application
POST
/api/auth
Login
Parameters
Try it out
Name
Description
loginRequest
* required
(body)
loginRequest
Example Value
Model
{
"email": "string",
"password": "string"
}
Parameter content type
application/json


Available authorizations
apiKey (apiKey)
Name: Authorization
In: header
Value:
Authorize
Close

HACKTIV8


localhost:8080/swagger-ui.html?urls.primaryName=Us
Authorize
bus-reservation-controller
Operations pertaining to agency management and ticket issue in the BRS application
POST /api/v1/reservation/bookticket bookTicket
GET /api/v1/reservation/stops getAllStops
GET /api/v1/reservation/tripsbystops getTripsByStops
GET /api/v1/reservation/tripschedules getTripSchedules
user-controller
Operations pertaining to user management in the BRS application
POST /api/v1/user/signup signup

Output

Please note this one :

Nb.
1. Make Sure your code not error and your admin authentification is match with our mandatory requirements
2. Push to your Github & Do Deployment With Heroku
3. Copy your link repository Github & Heroku on txt/docs files.
4. Then Upload to this section for complete your 1st Journey on this course(s)