Restaurant

Restaurant web applicaion.

Built With

- Spring Boot - REST APIs backend
- AngularJS - Frontend
- Spring Data - Data access
- MySQL - DataBase

Spring boot

REST APIs backend using JWT for Authentication. App run on port :8080. User should use Email to login to the system.

- version 2.3.1

- Maven - Dependency Management.

  - Available Routes:

  - User routes (for sign up and login) :

    ~~~
    ```
    Post /auth/sign-up
    ```
    ```
    Post /auth
    ```
    ~~~

  - Tables routes: ** to get all tables from db **

    ~~~
    ```
    Get /tables/
    ```
    ** to add table to db **
    ```
    Post /tables/
    ```
    ** to get available tables for certain date **
    ```
     Get /tables/available/{date}
    ```
    ~~~

  - Reservations routes:

    ~~~
    ** to get all reservations from db **
    ```
    Get /books/all
    ```
    ** to add reservation to db **
    ```
    Post /books/
    ```
    ~~~

~~~
** to get user reservations for certain date **
```
 Get /book/me
```
~~~

When the app run for the first time:

- the initailizer class do the following:
  - create admin user 'email = admin@admin.com , password = "admin"'
  - create user 'email = user@user.com , password = "user"'
  - create number of tables:
    - Four tables that can accommodate a max of 2 persons.
    - Seven tables that can accommodate a max of 5 persons.
    - Two tables that can accommodate a max of 10 persons.

AngularJS

The web application frontend run on port:4200.

- version 8.3.28

- **App Compoments:**
  - Home Component - web home page.
  - Sign-up Component - sign up form.
  - Sign-in Component - login form.
  - User-Home Component - reservation form to book table, user own reservations.
  - Available-tables Component - user can search for availabe tables for specific date.
  - Admin-Home Component - add table form, view all restaurant tables.
  - Reservation Component - admin see and search for specific books by date.

  App Services:
  - Server service : deal with server side attach token to the header each request if user is logged in.
  - Auth service : responsible for user login, save token in local storage and decode token to check user role.
  - GuardAuth service : guard routes according user role.
  - Tables sevice : responsible for tables and resevation "add and fetch".

  App routes:

  --

Route	Compnent
'' or /home	HomeCompnent
/sign-in	SignInComponent
/sign-up	SignUpComponent
/user	UserHomeComponent
/available	AvailableTablesComponent
/admin	AdminHomeComponent

/admin-books	ReservationComponent

MySQL

Web application DataBase

- database name: orange_restaurant
- **db tables:**
  - **USER:**
  - **TABLES:**
  - **RESERVATIONS**
-
  -