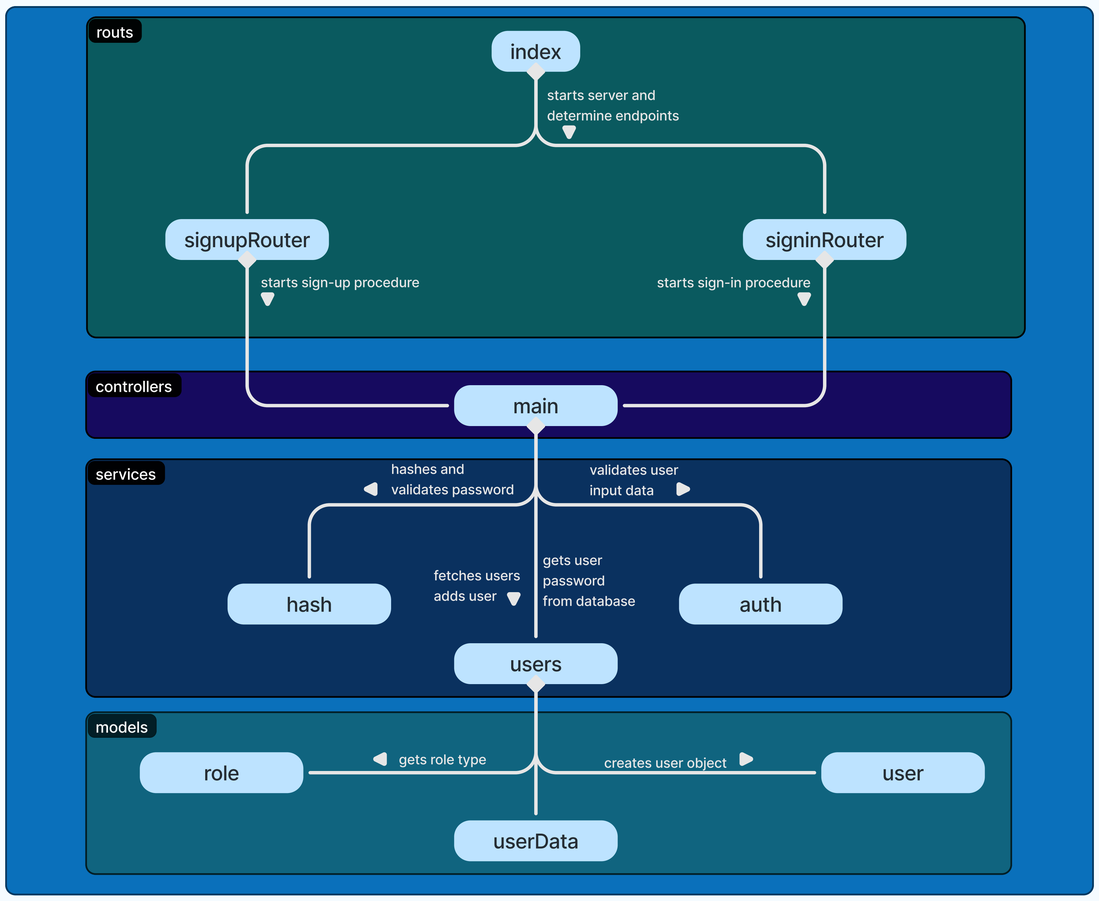
Authentication System

Description

The program receives user input data and validates them. If the data is valid, then the program proceeds.



Scripts & Classes

1. **Routes**
   a. **index.js:** Starts the server and determines the API endpoints.
   b. **signupRouter.js:** Router for signing up.
   c. **signinRouter.js:** Router for signing in.
2. **Controllers**
   a. **main.js:** Starts the procedures.
3. **Services**
   a. **hash.js:** Hashes and validates a given password.
   b. **auth.js:** Authenticates user data.
   c. **users.js:** Contains methods related to the users from database.
4. **Models**
   a. **role.js:** Contains constant strings for user role.
   b. **userData.json:** Contains users from the database.
   c. **user.js:** Contains a class to represent a User with constructor, getters and setters

Procedures

- **Sign-up:** The program first fetches users from the database and starts the authentication of the client's input data. If the validation is successful, it will proceed to hash the password. Finally, the program adds the new user to the database.

- **Sign-in:** The program first fetches users from the database and starts the authentication of the client's input data. If the validation is successful, it will proceed to get the hashed password of that user from the database to validate the password. Afterwards, the program checks whether or not the account has been confirmed by the user. Finally, the program will be ready to fetch user data from the database and send them to the client and proceed to the user dashboard.

How to use?

1. Go to the directory of the router: **.\backend\auth-service\src\routs**
2. Use this command to start the server: **node .\index.js**
3. Use these API endpoints:
   a. Sign up ⇒ **/auth/signup**
   b. Sign in ⇒ **/auth/signin**
4. For both procedures, the client must send all the necessary data in a JSON format.

```
{
    "firstName": "John",
    "lastName": "Doe",
    "email": "John@gmail.com",
    "password": "J0hn&Doe"
}
```

Sign up data

```
{
    "email": "John@gmail.com",
    "password": "J0hn&Doe"
}
```

Sign in data