

Tax Ease Application

Problem Statement:

←

 Classroom

OAuth and OIDC
TaxEase Application

Problem

Submissions

Solutions

Doubts

Copy

Share Code

Spring IDE Notes

Problem statement

You've been provided with a **tax management application** where you're required to incorporate real users and establish the login with the Okta feature using OAuth 2.0.

▼ **Tasks:-**

- Complete the UserController class by adding the following APIs:
 - GET "/user/all": It fetches the list of all users and can only be accessed by the role "ADMIN".
 - GET "/user/{userid}" (@PathVariable Long userid): It fetches the user from the given id and can only be accessed by the role "NORMAL".
 - POST "/user/signup" (@RequestBody UserDto userDto): It saves the user object and can be accessed by anyone.
 - PUT "/user/update/{id}" (@PathVariable Long id,@RequestBody UserDto userDto): It updates a user object and can only be accessed by the role "NORMAL".
- Add the following attributes in the UserDto:
 - username: String
 - password: String
 - age: int
 - role: String
- Complete the UserRepository by extending it with the JpaRepository and add the appropriate annotation.
- Implement logic for all methods in the UserService class:
 - createUser(UserDto userDto): This function saves a user in the database using UserRepository. If the role field in userDto is empty, it defaults to 'NORMAL'.
 - getAllUsers(): It returns the list of all users present in the database.
 - updateUser(UserDto userDto, Long id): This method checks if the user exists by the given ID in the database. If found, it updates the user's information via UserRepository, excluding modifications to the role field.

Problem

Submissions

Solutions

Doubts

- username: String
- password: String
- age: int
- role: String

- Complete the UserRepository by extending it with the JpaRepository and add the appropriate annotation.
- Implement logic for all methods in the UserService class:
 - createUser(UserDto userDto): This function saves a user in the database using UserRepository. If the role field in userDto is empty, it defaults to 'NORMAL'.
 - getAllUsers(): It returns the list of all users present in the database.
 - updateUser(UserDto userDto, Long id): This method checks if the user exists by the given ID in the database. If found, it updates the user's information via UserRepository, excluding modifications to the role field.
 - getUserById(Long userid): This method retrieves the user from the UserRepository in the database. If the user is found, it returns the user; otherwise, it throws a 'UserNotFoundException' with a custom message.
- Complete the UserNotFoundException by extending it with the appropriate class.
- Create a LoginController and a login.html file as described below:
 - Create a 'login.html' file within the templates folder. Set up a login form with a "login with Okta" button that redirects to Okta's redirect URL.
 - Create a method login() and use mapping GET "/login": It returns login as a string.
- Lastly, add the required configuration for oauth2 in the application.yml and TaxSecurityConfig file.
- Finally, test your application.

► **Special Instructions for submitting the solution:**

► **Note:-**