**User Details Service**

Path: com.opus.backend.features.UserDetailsService.class

This class implements the **UserDetailsService** interface from Spring Security. This class provides authentication and user management services for a backend web application that uses JSON Web Tokens (JWT) for user authentication.

Here's a breakdown of the class and its methods:

1. Imports: The class imports necessary packages and classes for user management and Spring Security.
2. Class Declaration: The **@Service** annotation indicates that this class is a Spring service component. The class implements the **UserDetailsService** interface for custom user authentication and management.
3. Autowired Components:
   * **UserDao**: A data access object (DAO) for interacting with the user repository.
   * **PasswordEncoder**: A password encoder for hashing user passwords.
4. **loadUserByUsername()**: This method overrides the method from the **UserDetailsService** interface. It takes a username as input and returns a **UserDetails** object if the user is found in the database, otherwise throws a **UsernameNotFoundException**.
5. **user\_status()**: This method checks if a user with the given email or username already exists in the database. It returns:
   * **-2**: If the email already exists in the database.
   * **-1**: If the username already exists in the database.
   * **1**: If the user does not exist in the database and has been successfully saved.
6. **save()**: This method creates a new **DAOUser** object from the input **UserDTO** and saves it to the database using the **userDao** repository. The user's password is hashed using the **bcryptEncoder** before saving.
7. **updateUser()**: This method updates an existing **DAOUser** object in the database.
8. **getUser()**: This method retrieves a **DAOUser** object from the database by username.
9. **setPassword()**: This method sets the password for a **DAOUser** object, hashes it using the **bcryptEncoder**, and saves the updated object to the database.

In summary, the **JwtUserDetailsService** class provides essential user management and authentication features, including user retrieval, checking for existing users, saving and updating users, and setting user passwords. The class is integrated with Spring Security to ensure secure user authentication using JWT.