

Create a RBAC system

Details

A Django application which will have a web interface, API's and a database(PostgreSQL). The web application will have views which will show the different sections as per the roles the user has access to. The apis should have a check as well on the basis of the roles.

A user should be able to login using the username and password on the web application. Once the user is authenticated the user will be getting a token which has to be saved in the database. The token has to be a JWT token.

The expiry of the token will be for 1 day.

The system should have different roles.

Details of the role

1. Admin

a. Actions

- i. Can add users.
- ii. Add API's
- iii. Remove Users.
- iv. Update Users.
- v. Remove API
- vi. Update API
- vii. View API

2. User

a. Actions

- i. Add API's
- ii. Update API's.
- iii. Map API's to users.
- iv. View API's.

3. Viewer

a. Actions

- i. View API's.

1. Add user: Add user and map the role for the user and the API's the user can access.
2. Remove user: Remove user and the mapping of the role and API's the user can access.
3. Update user: Update the role and the mapping of the API's the user can access.
4. Add API: Create new API. A database entry in the Django app and the access to it.
5. Remove API: Remove the API and if there are any users mapped to the API an error should be thrown.
6. Update API: Update the API in the database.
7. View API: List the API's mapped against the user.

For all the above 7 there has to be different API's created in the Django Application and the web interface should be displaying and taking the input for it.

If the user is not authorized for the role to access the API then **401 Unauthorized Exception** should be returned.

A user will be mapped against a certain role.

For example:

- John Doe if mapped to an admin role can carry out the admin actions.
- Sue Smith if mapped to the user role will carry out the user actions.
- Jonnathan Green if mapped to the viewer role will carry out the view actions.