API Token Management

Database

1. Execute the SQL script *SetupDatabase.sql* to create a new *TokenManager* database, create user roles and login for command & query access, and to create tables and stored procedures used in this application.
2. The database consists of two tables, *sec.Users*  where the users of this application is stored and *dbo. ApiToken* where the API tokens are created.
3. An admin user is created with username: *adminuser* and password *123456*.
4. The *dbo. ApiToken* table is populated with around 100 records to display on home page.
5. The stored procedures are added for auth, token operations.

.Net Core API

1. Restore NuGet packages if required.
2. The *TM.Api* is the Web API project to expose the end points.
3. The *TM.Configuration* is used to populate configuration values from *appsettings.json*.
4. The *TM.Data.\** projects are for database related entity access.
5. The *TM.Models* project is used to define models.
6. The TM.Security.\* projects are used for authentication/authorization related operations.
7. Modify *appSettings.json* for database connection string and Jwt issuer and audience.
8. *Swagger* is enabled in the root of Web API to test end points.
9. *Validate token* can be tested using *swagger*  with the end point */api/token/validate/{token}*.

Angular

1. Angular application is in the *WebNg* folder in the root.
2. *Angular 12* is used to develop the front-end.
3. *Angular Material* is used for the UX design.
4. Execute *npm install* to install packages*.*
5. Web API url is saved in *config.service.ts* and *app.module.ts.* Ideally those should come from server configuration, not implemented in this version. Update the URL in both places before testing.
6. Components are created in the *component* folder.
7. Dialogsare created in the *component/dialogs* folder*.*
8. Models are crated in the *models* folder.
9. Pipes are created in the *pipes* folder.
10. Services are created in the *services* folder.
11. Security (Auth/Jwt) related services are in *services/security* folder.
12. Routing is implemented in the *app-routing.module.ts*.
13. Logout feature is implemented.

Features

1. Authentication is implemented.
2. Jwt token is used for authentication.
3. Infinite scroll is implemented in the home page.
4. Generate token is implemented.
5. Disable token is implemented.
6. Ended tokens are displayed in red colour.
7. Validate token public API is available at */api/token/validate/{token}*
8. Logout is implemented.
9. Search is implemented on token list on change event.