

Tokenize

We have been introduced to token-based authentication and we just can't get enough of it. We want to create a new API using token-based authentication to test out our newly found skills. Let's dive into the world of tokens.

Assignment description

In order to complete the assignment, the steps below must be followed and implemented.

1. Setup three-layer structured project with:
 - a. API (Tokenize.API)
 - b. Models (Tokenize.API.Models)
 - c. Services (Tokenize.API.Services)
 - d. Repositories (Tokenize.API.Repositories)
2. Register JWT Bearer authentication in Program.cs and using values from appsettings.json
3. Set up a database context in the repository layer and register the database context in Program.cs using SQLite and values from appsettings.json
4. Set up two controllers
 - a. AccountController
 - i. Register
 - ii. Login
 - b. BookController (all routes should be authorized)
 - i. GetAllBooks()
 - ii. GetBookById(int id)
 - iii. CreateBook(BookInputModel inputModel)
 - iv. UpdateBookById(int id, BookInputModel inputModel)
5. Set up three services
 - a. AccountService
 - i. This service must make use of same **Hasher** as seen in **Authentication Soup** and available in GitHub (<https://github.com/arnarleifs/WebServicesRU/blob/master/authentication/authentication-soup/AuthenticationSoup.Api.Services/Utilities/Hasher.cs>)
 - ii. Register: void
 - iii. Login: string
 - b. BookService
 - i. GetAllBooks: BookDto[]
 - ii. GetBookById: BookDetailsDto
 - iii. CreateBook: int
 - iv. UpdateBookById: void

- c. JwtTokenService
 - i. GenerateJWT: string
 - 1. Signs the token using a signing key (stored in appsettings.json)
 - 2. Attaches the claims: Full name, email address and national id (kennitala)
 - 3. Validates the issuer
 - 4. Validates the audience
- 6. Set up two repositories
 - a. AccountRepository
 - i. Register: void
 - ii. GetUser: UserDto?
 - b. BookRepository
 - i. GetAllBookes: BookDto[]
 - ii. GetBookById: BookDetailsDto
 - iii. CreateBook: int
 - iv. UpdateBookById: void

Models

- Entities
 - Book
 - Id: int
 - Name: string
 - Description: string?
 - NumberOfPages: int
 - CreatedDate: DateTime
 - ModifiedDate: DateTime?
 - User
 - Id: int
 - FullName: string
 - EmailAddress: string
 - HashedPassword: string
 - RegistrationDate: DateTime
- DTOs
 - UserDto
 - Id: int
 - FullName: string
 - EmailAddress: string
 - BookDto
 - Id: int
 - Name: string
 - NumberOfPages: int
 - BookDetailsDto
 - Id: int
 - Name: string
 - Description: string?
 - NumberOfPages: int
- Input models
 - RegisterInputModel
 - FullName: string
 - EmailAddress: string
 - Password: string
 - ConfirmPassword: string
 - LoginInputModel
 - EmailAddress: string
 - Password: string
 - BookInputModel
 - Name: string
 - Description: string?
 - NumberOfPages: int