Curso .NET

Unidade Curricular: Laboratório Web **Docente:** David Jardim

---

FICHA DE TRABALHO 14

Exercícios:

1. Projeto (*API with controllers*)
    a. Crie um projeto do tipo *ASP.NET Core Web App (MVC)* denominado por Ficha14
2. Modelos
    a. Crie uma pasta denominada por Models
    b. Crie uma classe *User* para representar um utilizador e adicione as seguintes propriedades (tenha em atenção os tipos) respeitando os nomes
        i. ID
        ii. UserName
        iii. Password
        iv. Role
        v. Email
    c. Crie uma classe *UserContext* para criar a sessão com a base de dados e que será usada para realizar as operações necessárias à BD
        i. Adicione uma propriedade do tipo DbSet para realizar queries à base de dados utilizando instâncias do tipo User
        ii. Implemente o construtor por parâmetros (consulte o material disponibilizado)
        iii. Sobreponha o método *OnConfiguring* tendo em conta a *connection string* do seu servidor de MySQL (consulte o material disponibilizado)
        iv. Sobreponha o método *OnModelCreating* tendo em conta as propriedades das colunas das tabelas na DB(consulte o material disponibilizado)
3. Serviços
    a. Crie uma pasta denominada por Services
    b. Crie uma interface denominada por *IUserService* e defina os seguintes métodos abstratos tendo em conta os seus parâmetros e tipo a devolver
        i. Get(userName, password)
        ii. Create
    c. Crie uma classe denominada por *UserService* que implementa a interface anterior
        i. Implemente o construtor por parâmetros que receberá como parâmetro uma referência para a classe *UsersContext*

ii. Implemente os métodos abstratos que utilizarão a referência para a classe *UsersContext* de forma a realizar as operações de seleção e modificação na base de dados (consulte o material disponibilizado)

4. API Controllers
    a. Na pasta *Controllers* crie um controller Users*Controller*
        i. Adicione um atributo privado do tipo *IUserService*
        ii. Implemente o construtor por parâmetros, onde o parâmetro é do tipo *IBookService*
        iii. Implemente os seguintes endpoints tendo em conta as validações necessárias e respostas adequadas e utilize o serviço para :
            1. Adicionar um novo utilizador, o ID do novo utilizador deve ser devolvido na resposta
            2. Selecionar apenas um utilizador pelo seu *UserName* e *Password*
5. Dados
    a. Crie uma pasta denominada por Data
    b. Crie uma classe estática denominada por *UsersDBInitializer*
    c. Implemente um método estático denominado por *InsertData*, este método deverá receber uma referência para a classe *UsersContext,* e essa referência deverá ser utilizada para adicionar alguns utilizadores iniciais à base de dados
    d. Crie uma estática classe denominada por UsersExtension que será usada como um middleware para invocar o método para inserir os dados na base de dados
    e. Na classe LibraryExtension implemente um método estático chamado *CreateDbIfNotExists* para criar a base de dados senão existir e inserir os dados (consulte o material disponibilizado)

6. Program
    a. Aplicando *dependency injection* adicione à lista de serviços da aplicação um contexto para a base de dados do tipo *UsersContext* (consulte o material disponibilizado)
    b. Aplicando *dependency injection* adicione à lista de serviços da aplicação um serviço com tempo de vida scoped para a base de dados do tipo *IUserService* com implementação *UserService* (consulte o material disponibilizado)
    c. Aplicando *dependency injection* adicione à lista de serviços da aplicação um serviço com tempo de vida transient para a base de dados do tipo *IJWTService* com implementação *JWTService* (consulte o material disponibilizado)
    d. Invoque a extensão *CreateDbIfNotExists* através da instância para a sua aplicação de forma a criar a base de dados

7. JWT
    a. Adicione as configurações necessárias para o JWT ao ficheiro **appsettings.json**
    b. Inspecione a interface IJWTService e a classe concreta JWTService
    c. Adicione os serviços necessários para:
        i. Ativar a Sessão
        ii. Ativar a Autenticação
    d. Adicione os middlewares necessários para:
        i. Utilizar a sessão
        ii. Utilizar a autenticação
        iii. Utilizar a autorização
    e. Implemente o inline middleware para adicionar o cabeçalho de Bearer Token

8. Views
    a. Adicione as seguintes views e implemente o código HTML e Razor necessário para:
        i. SignUp.cshtm – Funcionalidade de SignUp
        ii. Index.cshtm – Funcionalidade de Login
        iii. UserDetails.cshtm – Funcionalidade para inspecionar os dados do utilizador
        iv. Logout.cshtm – Funcionalidade de Logout
        v. ApiTest.cshtm – Funcionalidade para testar a invocação de um endpoint da API
    b.
9. View Controller
    a. Implementar os endpoints que vão devolver as Views correspondentes, tenha em atenção a sua visibilidade para o exterior
        i. SignUp
        ii. UserDetails
        iii. Logout
        iv. ApiTest
    b. Implementar os endpoints que serão invocados nos formulários
        i. SignUpUser
        ii. LoginUser
        iii. LogoutUser
        iv. ApiTest