Passo a Passo Criação BlogAPI

**Será Utilizado: CRUD / Angular Forms / Image Selector / DDD / Entity Framework Core / EF Core Relationships / Authentication and Authorization – Identity (JSON Web Token) / Auth Guards**

**Blog Class:**

|  |  |  |
| --- | --- | --- |
| Category | Type | Null? |
| Id | Guid | No |
| Name | String | No |
| UrlHandle | String | No |

|  |  |  |
| --- | --- | --- |
| Category | Type | Null? |
| Id | Guid | No |
| Title | String | No |
| ShortDescription | String | No |
| Content | String | No |
| UrlHandle | String | No |
| FeatureImageUrl | String | No |
| DateCreated | DateTime | No |
| Author | String | No |
| IsVisible | Bool | No |

**Criação da Arquitetura (DDD):**

Models -> Domain (Classes, Serviços, Validações e Regras) ->

**Instalações dos pacotes:**

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

Microsoft.EntityFrameworkCore.Desing

**Db Context Class:**

Forma

Descrição gerada automaticamente

Criação da pasta Data -> Onde ficará o Contexto do Banco de Dados

A classe criada herdará de: DbContext

Depois inserir o construtor: public ApplicationDbContext(DbContextOptions options): base(options)

{}

Depois criar o DbSet<> que **será r**esponsável pela coleção dos dados a ser criado

Ex: public DbSet<BlogPost> BlogPosts { get; set; }

public DbSet<Category> Categories { get; set; }

**Criação do Connection String:**

"ConnectionStrings": {

"CodeConnectionString"(pode ser outro nome): "Data Source=DESKTOP-L8175GI\\SQLEXPRESS;Initial Catalog=BlogApiDb(Nome do banco que vai ser criado);Integrated Security=False;User ID=sa;Password=123456;Connect Timeout=15;Encrypt=False;TrustServerCertificate=False;"

}

**Injeção de Dependência:**

No arquivo Program.cs -> injetar antes do .Build

builder.Services.AddDbContext<ApplicationDbContext>(options =>

{

options.UseSqlServer(builder.Configuration.GetConnectionString("CodeConnectionString"));

});

Dessa maneira chamamos o AddDbContext para injetar em nossa aplicação, depois passamos a classe responsável pelo tratamento do banco (ApplicationDbContext) -> Para se comunicar com o BD do controlador ou os repositórios, pegamos o Options que foi passado como parâmetro e chamamos o UseSqlServer Vindo do pacote que foi instalado, em seguida chamamos o configuration e passamos a string de conexão responsável.

**Rodar o EF CORE MIGRATION**

1. Add-Migration “Name of Migration”
2. Update-Database

**Create Controller and Actions**

1. Criar a controladora
2. Injeção de Dependência no construtor da controladora

(Private readonly (nome do contexto) \_context)

Explicando DTOs -> Usado para transferir dados entre diferentes camadas.

O Cliente não precisa ter acesso ao Modelo de Domínio, mas apenas as propriedades necessárias. Dessa maneira ele não tem o mapeamento entre as tabelas do BD

Vantagens:

* Segurança
* Performance
* Versionamento

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Repository pattern**

É um padrão de Design que serve para separar o acesso aos dados camada do restante da aplicação, ele oferece uma interface padrão, para acessar e gerenciar dados em um armazenamento de dados.

Criação da Pasta -> Repositories onde ficará as Interfaces(Pasta) e as Implementações(Pasta) dessas interfaces

**Injetar no Program.Cs**

(nome da interface, Implementação da interface)

Ex: builder.Services.AddScoped<ICategoryRepository, CategoryRepository>();

**Criação Do Front**

1. Ng new (nome do projeto) – Angular
2. Depois da criação do projeto, cria-se os componentes seja ele (compartilhado) como o “Navbar” por exemplo e os componentes próprios de cada função.
3. Criação dos Services
4. Declarações dos imports e Components, declarar o bootstrap no index.html
5. Declaração das rotas de cada componente

**Exemplo abaixo:**

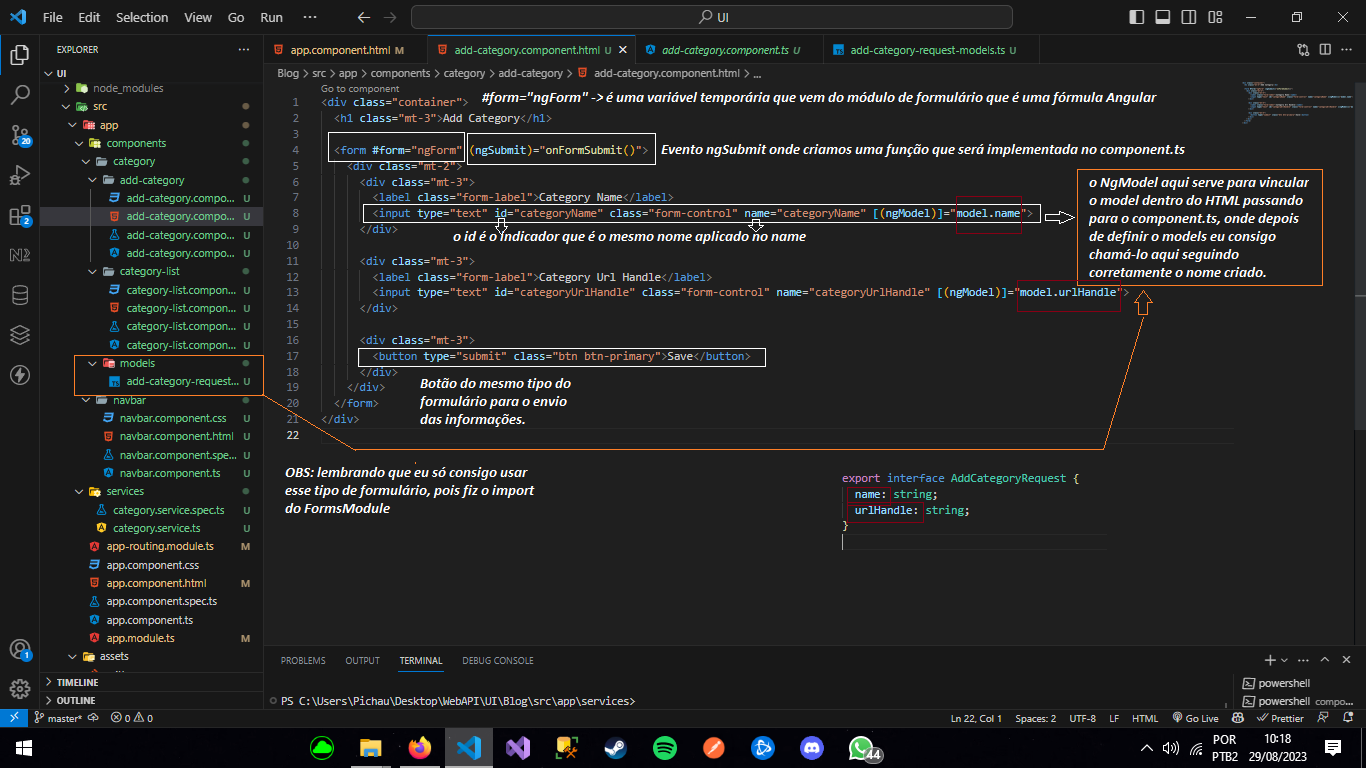
Tela de computador

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente