

Teste Prático

Dev .NET - Leanwork Group

Este documento tem como finalidade detalhar a criação da API de gestão de carrinho de compras, explicando alguns pontos e orientando o usuário de forma simples e objetiva com instruções detalhadas para configuração do ambiente e execução do projeto.

Configurações do Ambiente

Primeiro vamos configurar nosso banco de dados, rodando a query que está dentro da pasta SQL.

- 1 - Acesse: LeanWork/src/SQL/Scripts
- 2 - Copie a query dentro do documento, abra uma nova consulta no sql server, cole e execute a query.
- 3 - Resultado esperado: será criado um banco chamado **carrinho_de_compras** junto com duas tabelas chamadas: **Carrinho** e **Carrinho_Itens**.
- 4 - A tabela Carrinho será populada com 5 itens.

Agora vamos rodar nosso projeto e configurar nossa string de conexão.

- 1 – Acesse: LeanWork/src/CarrinhoDeCompras/CarrinhoDeCompras.sln
- 2 – Execute a solution para abrir o projeto
- 3 – Dentro do projeto, acesse appsettings.Development.json para configurar a string de conexão
- 4 – Cole sua string dentro de ConnectionStrings
- 5- Exemplo de uma string:

"ConnectionStrings":{

"DefaultConnection":

"Server=localhost;Database=carrinho_de_compras;User
Id=sa;Password=@Password123;TrustServerCertificate=True;"}

Com o ambiente configurado já podemos rodar o projeto, ao rodar podemos ver que o projeto consiste em 3 páginas: página de início, página do carrinho para listar os itens dos carrinhos e página para cadastrar um item em um determinado carrinho.

Página de Cadastro

Na página de cadastro podemos cadastrar um item, inserindo quantidade, preço unitário, produto e o id do carrinho de 1 a 5 para relacionar o item a um dos carrinhos que foram criados junto com a base de dados, seguindo as validações que foram impostas para cada campo, também temos a opção de voltar a página inicial.

Página do carrinho

A página do carrinho basicamente lista os itens dos carrinhos, mostrando o nome do produto, quantidade, preço unitário e preço total.

Na mesma página temos a opção de atualizar ou excluir um item, também temos a opção de voltar a página inicial.

Back-End

Optei por separar o projeto por pastas e não por camadas, por ser algo mais simples.

Criei os métodos no “CarrinhoRepository” e chamo cada um deles através do “CarrinhoService”.

Inseri algumas validações de erro nos métodos quando necessário e utilizei o Dapper para acessar a base de dados.

Criei um método interno chamado “GetCarrinhoItem” para acessar a base de dados e recuperar o item desejado de acordo com o id que eu passar, para atualizar o item.

Criei uma lógica no projeto para o campo “PrecoTotal”, sempre que eu inserir um item ou atualizar a quantidade de um item, o Preço total será calculado de acordo com a regra: $\text{Preço Total} = \text{Quantidade} * \text{Preço Unitário}$

