1 - Abra dentro de Discount.API/ Program.cs

using Discount.API.Extensions;

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Hosting;

namespace Discount.API

{

public class Program

{

public static void Main(string[] args)

{

var host = CreateHostBuilder(args).Build();

host.MigrateDatabase<Program>();

host.Run();

}

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

}

2 - Em Discount.API Adcione a pasta Extensions e dentro da pasta a Classe statica

2.1 - public static class HostExtensions...

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using Npgsql;

namespace Discount.API.Extensions

{

public static class HostExtensions

{

public static IHost MigrateDatabase<TContext>(this IHost host, int? retry = 0)

{

int retryForAvailability = retry.Value;

using (var scope = host.Services.CreateScope())

{

var services = scope.ServiceProvider;

var configuration = services.GetRequiredService<IConfiguration>();

var logger = services.GetRequiredService<ILogger<TContext>>();

try

{

logger.LogInformation("Migrating postresql database.");

using var connection = new NpgsqlConnection

(configuration.GetValue<string>("DatabaseSettings:ConnectionString"));

connection.Open();

using var command = new NpgsqlCommand

{

Connection = connection

};

command.CommandText = "DROP TABLE IF EXISTS Coupon";

command.ExecuteNonQuery();

command.CommandText = @"CREATE TABLE Coupon(Id SERIAL PRIMARY KEY,

ProductName VARCHAR(24) NOT NULL,

Description TEXT,

Amount INT)";

command.ExecuteNonQuery();

command.CommandText = "INSERT INTO Coupon(ProductName, Description, Amount) VALUES('IPhone X', 'IPhone Discount', 150);";

command.ExecuteNonQuery();

command.CommandText = "INSERT INTO Coupon(ProductName, Description, Amount) VALUES('Samsung 10', 'Samsung Discount', 100);";

command.ExecuteNonQuery();

logger.LogInformation("Migrated postresql database.");

}

catch (NpgsqlException ex)

{

logger.LogError(ex, "An error occurred while migrating the postresql database");

if (retryForAvailability < 50)

{

retryForAvailability++;

System.Threading.Thread.Sleep(2000);

MigrateDatabase<TContext>(host, retryForAvailability);

}

}

}

return host;

}

}

}

docker-compose -f docker-compose.yml -f docker-compose.override.yml up -d