

4) DPF text: user can select some number of copies and some number of pages to print. If user selects 2 copies and 2 pages, it will print 4 pages.

4) DPF text: user can select some number of copies and some number of pages to print. If user selects 2 copies and 2 pages, it will print 4 pages.

Left margin (EFC)

Criando string de conexão com o banco

as tabelas serão criadas de forma automática pelo código
de migrations, mas para isso precisamos configurar a string
de conexão

String de conexão → expressão com os parâmetros para conexão do BD

Us temos 2 tipos, um com autenticação do SQL server e a
com autenticação Windows
(trusted connection)

→ Na string consta a instância, nome do DB, detalhes
de autenticação e outras configurações de acordo
com o ambiente

Primeiro obtemos o nome do servidor e o nome da
instância, podemos fazer isso abrindo o Manager do SQL server →
Serviços do SQL → SQL server que estiver em execução → M2 → propriedades
Serviços, com isso conseguimos o nome da instância e do
servidor local.

Que podemos achar no VS22 em Exibir → Server explorer.

Para criar uma string a partir do VS22, clicamos em conexões de dados →
adicionadas informamos o nome do servidor e/ou instância,
da lista clicamos (M2) no banco → propriedades e iremos achar a string
de conexão.

Configurar o EF Core

↳ Baixar pacotes Nuget (gerenciador de pacotes)

Mic... Ent. Ent. C... SQL Server

(-- -- -- --) . Tools

(-- -- -- --) . Design

(Podemos ver os pacotes dentro do projeto ou nas dependências/pacotes)

↳ Criar uma classe de contexto que herda de DbContext (EFc)

↳ Criar um construtor de options do tipo `DbContextOptions<meucontext>`
que faz referência ao upper



Carrega as informações necessárias para configurar o DbContext

↳ Definir o meu DbSet para mostrar quais classes queremos mapear no BD

↳ Public DbSet<classe que serão mapeadas nome > set;

↳ Definir o string de conexão no appsetting.json

Basicamente essa classe que criamos carrega informações de configuração do DbContext e define quais classes serão mapeadas no banco

TD Definir o service no startup.cs no ConfigureServices

Is services.AddDbContext< arquivo de contexto>(x => x.name ->
do provedor do banco(Configuration.GetConnectionString("name ->
do string de conexão que definimos no appsettings"));

O serviço é criado com tempo de vida scoped, ou
seja, o cada request será criado novo no instâncias.

Data Annotations

São atributos que podem ser aplicados nos membros de sua classe, eles servem para válios, o modelo de funcionamento como os dados devem ser exibidos, especificar o localizamento das entidades e sobreescriver as convenções do EFC.

Podemos definir se algo é requerido no largo, max e min de caracteres, se é uma chave primária e não outros tipos, como válidos números e emails.

(~~Ex:~~) Esses atributos/regras devem ser aplicados nos model

Quando preenchemos nossa classe através de um request HTTP, ele verifica os atributos da classe para saber se os dados do request são válidos, assim a verificação ocorre no lado servidor.

(aplicado no projeto)

comunicação — somos a maneira
de dizer — aplicações e maneira
de dizer — como dizer maneira
de dizer — como dizer maneira

COMANDOS

aplicações e maneira
de dizer — somos a maneira
de dizer — como dizer maneira
de dizer — como dizer maneira

172 271
e no processo de aprendizagem de forma a que os

processos e de EFC e passo de comando de um trajeto
diferente ou seja DBA/DBAF/DB comando ..

— quando se tem a necessidade de execução.
maneira, seção maneira, para executar A01 e aplicações
de tipo: através de Leste, para ação, numerosas

— para serem a maneira de dizer.
Sua parte ativa é sua maneira de dizer.

Maryana

Quando digitamos o comando para criá-lo, só geraremos pasto Migration com a configuração das tabelas, para que ele não crie os dados da tabela sem as configurações (~~as~~) adequadas, precisamos usar o data anno-tations.