

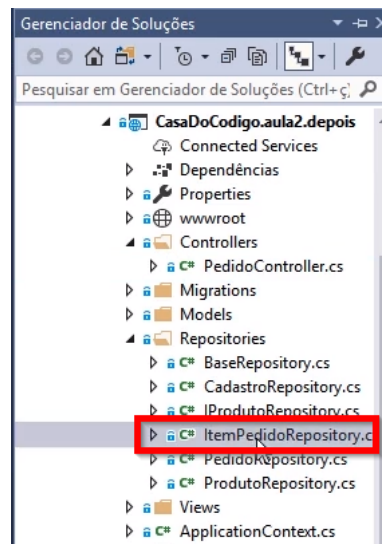
Chamando Repositório de Itens a Partir do Controller

Transcrição

Iniciamos a nossa aula 2 com um novo projeto para a solução. Seguiremos esse padrão de projeto por aula até o final do curso, dessa forma podemos comparar isoladamente o que foi produzido em cada atividade.

Em `PedidoController` nós temos o método `UpdateQuantidade()` que é acessado com sucesso a partir do código JavaScript. A próxima etapa é coletar as informações de `Id`, `ItemPedido` e `Quantidade` e fazer as alterações necessárias no banco de dados. Para isso, precisaremos utilizar como meio o **Entity Framework** e, como já aprendemos na primeira parte do curso, para acessar o Entity utilizamos o conceito de repositório, portanto adicionaremos um novo método para alterar a quantidade que irá cair na classe repositório de `ItemPedido`.

No gerenciador de soluções, abriremos o repositório `ItemPedidoRepository`, localizado na pasta `Repositories`.



Teremos acesso à interface que usaremos para declarar o novo método.

```
namespace CasaDoCodigo.Repositories
{
    public interface ItemPedidoRepository
    {
    }

    public class ItemPedidoRepository: BaseRepository<ItemPedido>, IItemPedidoRepository
    {
        public ItemPedidoRepository(ApplicatonContext) : base(contexto)
        {
        }
    }
}
```

Nada será retornado, então adicionaremos `void` e o nome o mesmo nome proveniente de `PedidoController`, isto é, `UpdateQuantidade` que receberá o parâmetro `ItemPedido`.

```
namespace CasaDoCodigo.Repositories
{
    public interface ItemPedidoRepository
    {
        void UpdateQuantidade(ItemPedido itemPedido);
    }

    public class ItemPedidoRepository: BaseRepository<ItemPedido>, IItemPedidoRepository
    {
        public ItemPedidoRepository(ApplicatonContext) : base(contexto)
        {
        }
    }
}
```

Faremos essa classe que implementa a interface, também implementar o método `UpdateQuantidade()` . Clicaremos sobre o nome da interface `IItemPedidoRepository` , digitamos o atalho "Ctrl + ." e selecionamos a opção "Implementar Interface".

Dessa forma, será criado um método novo, o `NotImplementedException()`

```
namespace CasaDoCodigo.Repositories
{
    public interface ItemPedidoRepository
    {
        void UpdateQuantidade(ItemPedido itemPedido);
    }

    public class ItemPedidoRepository: BaseRepository<ItemPedido>, IItemPedidoRepository
    {
        public ItemPedidoRepository(ApplicatonContext) : base(contexto)
        {
        }

        public void UpdateQuantidade(ItemPedido itemPedido)
        {
            throw new NotImplementedException();
        }
    }
}
```

Mesmo não existindo nenhum código dentro desse novo método, nós iremos realizar sua chamada a partir de `PedidoController` . Para fazer isso, **utilizaremos a injeção de dependências**.

Declararemos um novo campo privado, chamado `itemPedidoRepository` .

```
namespace CasaDoCodigo.Controllers
{
    public class PedidoController : Controller
    {
        private readonly IProdutoRepository produtoRepository;
        private readonly IPedidoRepository pedidoRepository;
        private readonly IItemPedidoRepository itemPedidoRepository;

        public PedidoController(IPedidoRepository itemPedidoRepository, IProdutoRepository produtoRepository)
```

```

    public PedidoController(IProdutoRepository produtoRepository,
        IPedidoRepository pedidoRepository)
    {
        this.produtoRepository = produtoRepository;
        this.pedidoRepository = pedidoRepository;
    }

```

O novo campo privado que criamos receberá o valor da injeção de dependência. Iremos incluir um novo parâmetro no construtor da classe, que será `IItemPedidoRepository itemPedidoRepository`. Em seguida faremos a atribuição, de modo que o campo privado receba o valor do parâmetro que foi injetado, portanto escreveremos `this.itemPedidoRepository = itemPedidoRepository`

```

namespace CasaDoCodigo.Controllers
{
    public class PedidoController : Controller
    {
        private readonly IProdutoRepository produtoRepository;
        private readonly IPedidoRepository pedidoRepository;
        private readonly IItemPedidoRepository itemPedidoRepository;

        public PedidoController(IProdutoRepository produtoRepository,
            IPedidoRepository pedidoRepository,
            IItemPedidoRepository itemPedidoRepository)
        {
            this.produtoRepository = produtoRepository;
            this.pedidoRepository = pedidoRepository;
            this.itemPedidoRepository = itemPedidoRepository;
        }
    }
}

```

Assim feito, no método `UpdateQuantidade()` acessaremos o novo método `UpdateQuantidade()` e passaremos o parâmetro `itemPedido`

```

namespace CasaDoCodigo.Controllers
{
    <****!****>

    [HttpPost]
    public void UpdateQuantidade([FromBody]ItemPedido itemPedido)
    {
        itemPedidoRepository.UpdateQuantidade(itemPedido);
    }
}

```

Acessaremos a página de produtos, selecionaremos qualquer um e clicaremos sobre o botão "+" quando estivermos na página de carrinho. Seremos redirecionados para o método `UpdateQuantidade()` do `PedidoController`. Usaremos o atalho "F11" e chegaremos à linha `throw new NotImplementedException()`.

```

public class ItemPedidoRepository : BaseRepository<ItemPedido>, IItemPedidoRepository
{
    public ItemPedidoRepository(ApplicationDbContext contexto) : base(contexto)
    {
    }
}

```

```
public void UpdateQuantidade(ItemPedido itemPedido)
{
    throw new NotImplementedException();
}
```

A partir desse ponto começaremos a implementar o novo método para gravar informações no banco de dados.