

Incluindo muitos livros

Transcrição

Até então, inserimos os livros individualmente na biblioteca. Veremos agora como é o processo para os adicionarmos em lote.

O primeiro passo será abrirmos a classe "Livro.cs", e criar uma função `incluiValoresLivro` :

```
public class valoresLivro
{
    public static Livro incluiValoresLivro(string Titulo, string Autor, int Ano, int Paginas, string Assunto)
    {
        Livro livro = new Livro ();
        livro.Titulo = Titulo;
        livro.Autor = Autor;
        livro.Ano = Ano;
        livro.Paginas = Paginas;
        string[] vetAssunto = assunto.Split(',');
        List<string> vetAssunto2= new List<string>();
        for (int i = 0; i <= vetAssunto.Length - 1; i++)
        {
            vetAssunto2.Add(vetAssunto[i].Trim())
        }
        livro.Assunto = vetAssunto2;
        return livro;
    }
}
```

Ela conterá os dados de título, autor, ano, número de página e assuntos. Estes últimos serão listados e separados por uma vírgula.

Criaremos então uma classe `Livro` , contendo todas as características listadas acima e, na variável `varAssunto` indicaremos o `split` por vírgulas (',') .

Inseriremos um `array` de assuntos, onde serão inseridos valores e, uma vez populado, haverá a valoração `Assunto` , que é uma lista de strings, e o valor do `array`, que são os assuntos separados por vírgulas.

Faremos uma nova classe, clicando com o botão direito do mouse sobre o nome do nosso projeto, e selecionando "Adicionar > Novo Item". Daremos o nome de "usandoValoresLivros.cs".

Incluiremos os livros por meio da classe `valoresLivro` . Lembrando sempre de fazer referência ao `MongoDB.Bson` e `MongoDB.Driver` .

Utilizaremos o código com o qual trabalhamos anteriormente e, para simplificar, removeremos os comentários.

Incluiremos referência à classe abaixo da conexão.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class usandovaloresLivros
    {
        static void Main(string[] args)
        {
            Task T = MainAsync(args);
            Console.WriteLine();
            Console.WriteLine("Pressione Enter");
            Console.ReadLine();
        }
        static async Task MainAsync(string[] args)
        {

            var conexaoBiblioteca = new conectandoMongoDB();

            Livro livro = new Livro();
            livro.Titulo = "Star Wars Legends";
            livro.Autor = "Timothy Zahn";
            livro.Ano = 2010;
            livro.Paginas = 245;
            List<string> listaAssuntos = new List<string>();
            listaAssuntos.Add("Ficção Científica");
            listaAssuntos.Add("Ação");
            livro.Assunto = listaAssuntos;

            await conexaoBiblioteca.Livros.InsertOneAsync(livro);

            Console.WriteLine("Documento Incluído");

        }
    }
}

```

Ao incluirmos um novo livro, em vez de utilizarmos a classe inserindo os dados dos livros individualmente, criaremos uma nova `Livro livro = new Livro();` .

Onde daremos a valoração e incluiremos os parâmetros, com o comando `livro = valoresLivro.incluiValoresLivro()` .

Incluiremos um novo livro da seguinte forma:

```

var conexaoBiblioteca = new conectandoMongoDB();

Livro livro = new Livro();
livro = valoresLivro.incluiValoresLivro("Dom Casmurro", "Machado de Assis", 1923, 188, "Romance,

await conexaoBiblioteca.Livros.InsertOneAsync(livro);

```

```
Console.WriteLine("Documento Incluído");
```

Para inserir um segundo livro, criaremos uma segunda variável, Livro2 :

```
var conexaoBiblioteca = new conectandoMongoDB();

Livro Livro = new Livro();
Livro = valoresLivro.incluiValoresLivro("Dom Casmurro", "Machado de Assis", 1923, 188, "Romance");
await conexaoBiblioteca.Livros.InsertOneAsync(Livro);
Livro Livro2 = new Livro();
Livro2 = valoresLivro.incluiValoresLivro("A Arte da Ficção", "David Lodge", 2002, 230, "Didático");
await conexaoBiblioteca.Livros.InsertOneAsync(Livro2);

Console.WriteLine("Documento Incluído");
```

Iremos salvar e executar nosso programa. Para isso clicaremos com o botão direito do mouse sobre o nome do nosso projeto, selecionando "Propriedades", definiremos o "Objeto de Inicialização" como "exemplosMongoDBusandoValoresLivros".

Feito isso, clicaremos em "Iniciar" para executarmos. Surgirá a mensagem "Documento Incluído".

Abriremos o Robomongo, lembrando de atualizá-lo, e, em "Biblioteca > Collections > Livros" será possível visualizar os novos documentos.

Isso não significa que deverão ser criadas novas classes sempre que quisermos adicionar um novo item, há uma maneira de fazermos isso em lote, ou seja, incluir vários livros de uma só vez.

Clicando sobre o nome do arquivo com o botão direito do mouse, selecionaremos "Adicionar > Novo Item", daremos o nome de "incluindoMuitosLivros.cs".

Este novo item será:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class incluindoMuitosLivros
```

Copiaremos o código do último exemplo e o utilizaremos, colando logo abaixo do nome da classe:

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class incluindoMuitosLivros
    {
        static void Main(string[] args)
        {
            Task T = MainAsync(args);
            Console.WriteLine();
            Console.WriteLine("Pressione Enter");
            Console.ReadLine();
        }
        static async Task MainAsync(string[] args)
        {

            var conexaoBiblioteca = new conectandoMongoDB();

            Livro Livro = new Livro();
            Livro.Titulo = "Star Wars Legends";
            Livro.Autor = "Timothy Zahn";
            Livro.Ano = 2010;
            Livro.Paginas = 245;
            List<string> Lista_Assuntos = new List<string>();
            Lista_Assuntos.Add("Ficção Científica");
            Lista_Assuntos.Add("Ação");
            Livro.Assunto = Lista_Assuntos;

            await conexaoBiblioteca.Livros.InsertOneAsync(Livro);

            Console.WriteLine("Documento Incluido");

        }
    }
}

```

Para inserirmos muitos livros ao mesmo tempo, em vez de utilizarmos a classe como parâmetro, será utilizada uma lista de classes.

Faremos a conexão criando a variável `List<Livro> Livros = new List<Livro>();` . No link para downloads do curso, estará disponível um arquivo intitulado "incluirMuitosLivros".

Copiaremos para nossa máquina e extrairemos, já que é um arquivo compactado.

Abrindo o arquivo com o editor de textos, teremos um trecho de programa com os dados de diversos livros.

Copiaremos o referido trecho e colaremos abaixo da nova variável criada.

Em cada livro é feita referência à função `valoresLivro` .

Abaixo, a função que utilizaremos é a `await conexaoBiblioteca.Livros.InsertManyAsync()` e, como parâmetro, incluiremos a lista de livros.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class incluindoMuitosLivros
    {
        static void Main(string[] args)
        {
            Task T = MainAsync(args);
            Console.WriteLine();
            Console.WriteLine("Pressione Enter");
            Console.ReadLine();
        }
        static async Task MainAsync(string[] args)
        {
            var conexaoBiblioteca = new conectandoMongoDB();

            List<Livro> Livros = new List<Livro>();
            Livros.Add(valoresLivro.incluiValoresLivro("A Dança com os Dragões", "George R R Martin"));
            Livros.Add(valoresLivro.incluiValoresLivro("A Tormenta das Espadas", "George R R Martin"));
            Livros.Add(valoresLivro.incluiValoresLivro("Memórias Póstumas de Brás Cubas", "Machado de Assis"));
            Livros.Add(valoresLivro.incluiValoresLivro("Star Trek Portal do Tempo", "Crispin Hahn"));
            Livros.Add(valoresLivro.incluiValoresLivro("Star Trek Enigmas", "Dedopolus Tim"));
            Livros.Add(valoresLivro.incluiValoresLivro("Emília no País da Gramática", "Monty Python"));
            Livros.Add(valoresLivro.incluiValoresLivro("Chapelzinho Amarelo", "Chico Buarque"));
            Livros.Add(valoresLivro.incluiValoresLivro("20000 Léguas Submarinas", "Julio Verne"));
            Livros.Add(valoresLivro.incluiValoresLivro("Primeiros Passos na Matemática", "Mário de Sá-Carneiro"));
            Livros.Add(valoresLivro.incluiValoresLivro("Saúde e Sabor", "Yeomans Matthew", 1956, 2019));
            Livros.Add(valoresLivro.incluiValoresLivro("Goldfinger", "Iam Fleming", 1956, 2019));
            Livros.Add(valoresLivro.incluiValoresLivro("Da Rússia com Amor", "Iam Fleming", 1956, 2019));
            Livros.Add(valoresLivro.incluiValoresLivro("O Senhor dos Aneis", "J R R Token", 1956, 2019));

            await conexaoBiblioteca.Livros.InsertManyAsync(Livros);

            Console.WriteLine("Documento Incluido");
        }
    }
}

```

Salvaremos o programa e clicaremos com o botão direito do mouse sobre o nome do nosso projeto, selecionando "Propriedades", definiremos o "Objeto de Inicialização" como "exemplosMongoDB.incluindoMuitosLivros". Agora clicamos em "Iniciar" para executar.

Surgirá a mensagem "Documento Incluido".

Abriremos o Robomongo e acessaremos a pasta "Livros", onde será possível observar todos os arquivos que acabamos de adicionar.

