

Encapsulando o acesso ao MongoDB

Transcrição

Até o momento tivemos de executar três comandos para acessar a coleção `Livros`, os acessos: ao servidor do MongoDB; à base de dados `Biblioteca`; e à coleção `Livros`.

De modo que não seja necessária a repetição destes três comandos a cada conexão, criaremos uma nova classe que fará isto.

Clicaremos sobre o nome do nosso projeto, na lateral direita da tela, onde selecionaremos "Adicionar > Novo Item", selecionando a opção "Classe", daremos o nome de "conectandoMongoDB". Para concluir, clicaremos em "Adicionar".

Iremos agora adicionar as referências `using MongoDB.Driver` e `using Mongo.Bson`.

Em primeiro lugar, declararemos três constantes: o string de conexão, o nome da base de dados e o nome da coleção.

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

namespace exemplosMongoDB
{
    class conectandoMongoDB
    {
        public const string STRING_DE_CONEXAO = "mongodb://localhost:27017";
        public const string NOME_DA_BASE = "Biblioteca";
        public const string NOME_DA_COLECAO = "Livros";
    }
}
```

Temos a localização do servidor, em seguida o nome da base de dados e, por último, o nome da coleção.

Em seguida declararemos mais duas variáveis, uma que faz referência ao acesso do servidor ao MongoDB `private static readonly IMongoClient _cliente`, e outra ao acesso do banco de dados `private static readonly IMongoDatabase _BaseDeDados`.

Posteriormente incluiremos a função de criação da classe: `static conectandoMongoDB()` com a variável `_cliente = new MongoClient(STRING_DE_CONEXAO)`, atentando ao fato de que a string de conexão está em uma constante.

Em seguida faremos a inicialização do banco de dados, com o comando `_BaseDeDados = _cliente.GetDatabase(NOME_DA_BASE)`.

Agora é necessário criarmos mecanismos que nos permitam utilizar estas variáveis externamente, para acessarmos a base de dados e a coleção.

Utilizaremos:

```
public IMongoClient Cliente
{
    get { return _cliente; }
}

public IMongoCollection<Livro> Livro
{
    get { return _BaseDeDados.GetCollection<Livro>(NOME_DA_COLECAO); }
}
```

Assim temos nossa classe conectando ao MongoDB e acessando o servidor, a base de dados e a coleção.

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

namespace exemplosMongoDB
{
    class conectandoMongoDB
    {
        public const string STRING_DE_CONEXAO = "mongodb://localhost:27017";
        public const string NOME_DA_BASE = "Biblioteca";
        public const string NOME_DA_COLECAO = "Livros";

        private static readonly IMongoClient _cliente;
        private static readonly IMongoDatabase _BaseDeDados;
    }

    public IMongoClient Cliente
    {
        get { return _cliente; }
    }

    public IMongoCollection<Livro> Livros
    {
        get { return _BaseDeDados.GetCollection<Livro>(NOME_DA_COLEÇÃO); }
    }
}
}
```

Iremos utilizar agora em nosso exemplo o acesso por meio desta classe.

Criaremos uma nova classe, clicando com o botão direito do mouse sobre o nome do nosso projeto, e selecionando "Adicionar > Novo Item". A chamaremos de "manipulandoClassesExternas.cs".

Reaproveitando o código com que trabalhamos anteriormente, iremos copiar, e colar abaixo da classe que acabamos de criar.

Utilizaremos as informações de um novo livro:

```
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;
```

Substituiremos as informações que estão logo abaixo do comentário //inicializar uma variável do tipo objeto livro, deixando apenas a referência ao novo livro.

Adicionalmente, as linhas de código de //acesso ao servidor do MongoDB até //incluindo documento virarão comentários, onde será acrescentado o acesso por meio da classe de 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 manipulandoClasses
    {
        static void Main(string[] args)
        {
            Task T = MainSync(args);
            Console.WriteLine("Pressione ENTER");
            Console.ReadLine();
        }

        static async Task MainSync(string[] args);
        {
            {
                // "Titulo":"Guerra dos Tronos",
                // "Autor":"George R R Martin",
                // "Ano":1999,
                // "Páginas":856
                // "Assunto": [
                //     "Fantasia",
                //     "Ação"
                // ]
            }

            var doc = new BsonDocument
```

```
//      {
//          {"Título", "Guerra dos Tronos"}
//          {"Autor", "George R R Martin"}
//          {"Ano", "1999"}
//          {"Páginas", "856"}
//      };
//
//      var assuntoArray = new BsonArray ();
//      assuntoArray.Add ("Fantasia");
//      assuntoArray.Add ("Ação");
//      doc.Add("Assunto", assuntoArray);
//
//      Console.WriteLine(doc);
//
//      inicializar uma variável do tipo objeto livro
//
//      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;
//
//      //// acesso ao servidor do MongoDB
//
//      //string stringConexao = "mongodb://localhost:27017;
//      //IMongoClient cliente = new MongoClient(stringConexao);
//
//      //// acesso ao banco de dados
//
//      //IMongoDatabase bancoDados = cliente.GetDatabase("Biblioteca")
//
//      //// acesso a coleção
//
//      //IMongoCollection<Livro> colecao = bancoDados.GetCollection<Livro>("Livros");
//
//      //Acessando através da Classe de conexão
//
//      var conexaoBiblioteca = new conectandoMongoDB();
//
//      ////incluindo documento
//
//      await colecao.Biblioteca.Livros.InsertOneAsync(livro);
//
//      Console.WriteLine("Documento Incluído");
//  }
}
```

Para realizar o acesso e para que o documento seja incluído, acessando a coleção Livros , inserimos:

```
//Acessando através da Classe de conexão
//
var conexaoBiblioteca = new conectandoMongoDB();
```

```
////incluindo documento

await colecao.Biblioteca.Livros.InsertOneAsync(Livro);

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

Precisamos respeitar o nome da classe `Livro`, inclusive considerando os caracteres em maiúsculo ou minúsculo.

Iremos salvar e clicar com o botão direito do mouse sobre o nome do arquivo, selecionaremos "Propriedades" e, em "Objeto de Inicialização", definiremos como "exemplosMongoDB.manipulandoClassesExternas". O próximo passo é executar o programa.

Feito isso, surgirá uma mensagem informando que o documento foi incluído.

Abriremos o Robomongo e clicaremos duas vezes com o botão esquerdo do mouse sobre a coleção "Livros", verificando que há três elementos