09

Acessando o MongoDB e incluindo um documento BSON

Transcrição

O próximo passo será construir um primeiro exemplo dentro do servidor do MongoDB.

Trabalharemos com o documento criado anteriormente, incluindo dentro de um banco de dados e de uma coleção no servidor MongoDB.

Criaremos uma nova classe, clicando com o botão direito do mouse sobre o nome do nosso arquivo "exemploMongoDB" e selecionando "Adicionar > Novo Item".

Surgirá uma caixa de diálogo onde escolheremos a primeira opção "Classe", e daremos a ela o nome "acessandoMongoDB".

Clicaremos em "Adicionar" para finalizar o processo.

Utilizaremos o mesmo código com o qual trabalhamos anteriormente:

```
using System;
using System.Collections.Generic;
using System.Ling;
using System.Text;
using System.Threading.Tasks;

namespace exemploMongoDB
{
    class programAssincrono
    {
        static void Main(string[] args)
        {
            Task T = MainSync(args);
            Console.WriteLine("Pressione ENTER")
            Console.ReadLine();
        }

        static async Task MainSync(string[] args);
        {
            {
//    "Título":"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);
    }
}
```

Ao colar o código no arquivo que criamos no Visual Studio, percebemos que ele apresenta um erro. Tanto na variável `var doc = new BsonDocument` quanto na `var assuntoArray = new BsonArray ()` .

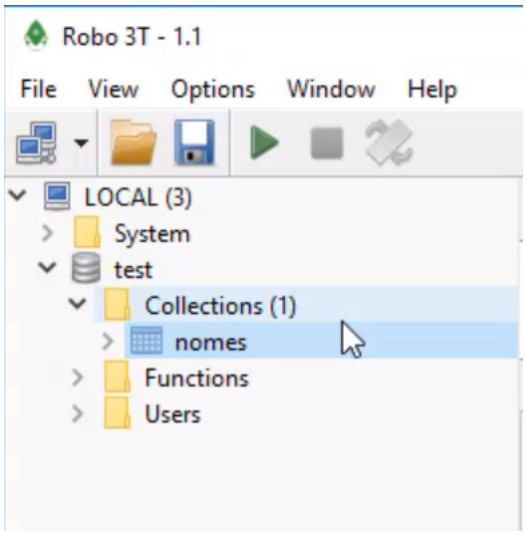
Isso porque nos falta inserir a referência `using MongoDB.Bson` . Além disso, vamos acrescentar ainda a referência ao `MongoDB.Driver` .

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

A `MongoDB.Bson` é utilizada para trabalhar somente com documentos "JSON". Enquanto isso, a `MongoDB.Driver` permite o acesso ao servidor MongoDB e, consequentemente, ao banco de dados e às coleções.

Ao final do código, faremos a conexão com o servidor. Este acesso é feito em três fases.

Abrindo o Robomongo, vemos que nosso servidor tem uma localização `localhost 27017` . Temos um banco de dados e, dentro dele, uma coleção.



Iremos criar este caminho no nosso programa.

A primeira coisa a fazer é criar uma variável string, que conterá o acesso ao servidor. No texto da string incluiremos informações como a localização do servidor, da porta, e assim por diante.

Se pesquisarmos na internet por "string connection mongodb", verermos no manual do MongoDB (https://docs.mongodb.com/manual/reference/connection-string/) que há diversas formas de escrevê-la.

Neste caso teremos uma string de conexão mais simples, pois nosso servidor é único e local.

```
// acesso ao servidor do MongoDB

string stringConexao = "mongodb://localhost":27017;
```

Este string de conexão simples já é o suficiente para acessarmos nosso servidor. Feito isso, criaremos uma variável do tipo servidor de MongoDB.

A variável tem um tipo especial, é o `IMongoClient`, oriundo do `MongoDB.Driver`. Ela terá a denominação `cliente`, e será uma `new MongoClient` com o parâmetro `stringConexao`.

Portanto, o resultado final será `IMongoClient cliente = new MongoClient(stringConexao);`.

Teremos armazenada dentro da variável `cliente` a conexão com um servidor do MongoDB.

```
// acesso ao servidor do MongoDB

string stringConexao = "mongodb://localhost":27017;
IMongoClient cliente = new MongoClient(stringConexao);
```

O próximo passo será estabelecermos uma conexão com uma biblioteca.

```
// acesso ao servidor do MongoDB

string stringConexao = "mongodb://localhost:27017";
IMongoClient cliente = new MongoClient(stringConexao);

// acesso ao banco de dados
```

Para isso, teremos uma variável especial que representa o banco de dados do MongoDB, que é `IMongoDatabase`. A variável se chamará `bancoDados` e, como ela está hierarquicamente abaixo do servidor, será `cliente.GetDatabase`.

Nos parâmetros colocaremos o nome do banco de dados. Como estamos trabalhando com livros, ele se chamará `Biblioteca`.

Não precisamos nos preocupar com o fato de que não há ainda um banco denominado `Biblioteca`. Isso porque, neste caso, a função `GetDatabase` criará um automaticamente.

```
// 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");
```

O próximo passo é acessarmos nossa coleção.

```
// 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
```

Assim como temos a variável `IMongoClient` para o servidor, e a `IMongoDatabase` para o banco de dados, agora teremos a `IMongoCollection`.

Esta variável é de um tipo lista, ou seja, é um array que possui diversos elementos e pode chegar a ter muitos documentos.

Assim como nas demais listas do C#, indicaremos sua natureza utilizando o símbolo `<>`.

A coleção é do tipo `BsonDocument` e se chamará "colecao". Como vimos anteriormente, um `BsonDocument` corresponde a um arquivo "JSON".

Utilizaremos a função `GetCollection`, que funciona de maneira muito similar à `GetDatabase`. Do mesmo modo, ela criará uma coleção quando não houver nenhuma. Ela também é uma lista do tipo `BsonDocument`, e terá o nome de "Livros".

```
// 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<BsonDocument> colecao = bancoDados.GetCollection<Bsondocument>("Livros");
```

Isso fará com que, ao criar o banco de dados "Biblioteca", também seja criada uma coleção "Livros".

Feito isso, incluiremos o documento, que está localizado na variável `doc`.

Por ser um programa assíncrono, vamos utilizar a função `await` e, para a coleção, será utilizada a função `InsertOneAsync`. O parâmetro do comando é o `doc`.

Em seguida vamos criar um comando `Console.WriteLine("Documento Incluido")` .

O resulado final será:

```
using System;
using System.Collections.Generic;
using System.Ling;
using System.Text;
using System.Threading.Tasks;

namespace exemploMongoDB
{
    class programAssincrono
    {
        static void Main(string[] args)
        {
            Task T = MainSync(args);
            Console.WriteLine("Pressione ENTER")
            Console.ReadLine();
        }

        static async Task MainSync(string[] args);
        {
            {
//    "Título":"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);
    }
}

        // 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<BsonDocument> colecao = bancoDados.GetCollection<Bsondocument>("Livros'

//incluindo documento

await colecao.InsertOneAsync(doc);

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

Agora vamos executar este programa. Clicaremos sobre o nome do nosso arquivo com o botão direito do mouse e selecionaremos a opção "Propriedades".

Em "Objeto de Inicialização" selecionaremos a função `exemplosMongoDB.acessandoMongoDB` .

Clicaremos sobre o botão "Iniciar" na barra de menu superior, assim executaremos o programa.

Feito isso, surgirá uma nova janela, onde lê-se:

> {"Título" : "Guerra dos Tronos", "Autor" : "George R R Martin", "Ano" : 1999, "Paginas" : 856, "Assunto": ["Fantasia","Ação"]}
>
> Pressione ENTER
>
> Documento Incluido

O "Pressione ENTER" aparece primeiro porque estamos executando de forma assíncrona.

Retornaremos ao Robomongo e clicaremos com o botão direito do mouse sobre o primeiro item em nosso diretório, escolheremos a primeira opção "Refresh" para atualizar nosso banco.

Veremos que o item "Biblioteca" surgirá e, dentro dele, haverá a pasta "Collections" com o item "Livros" dentro.

Isso conclui o primeiro exemplo acessando o MongoDB.