

03

Lendo Arquivo Json de Produtos

Transcrição

Ao rodarmos nossa aplicação novamente, percebemos que cometemos um engano ao obtermos uma instância do `DataService` — o ASP.NET Core não reconhece este tipo. Isso porque na verdade criamos o registro de uma interface `IDataService`, sendo necessário trocarmos o `GetService`:

```
serviceProvider.GetService<IDataService>().InicializaDB();
```

Isso, sim, fará com que seja gerada uma instância a partir desta interface. Acessaremos `InicializaDB()` da classe `DataService` e começaremos a fazer a carga do nosso catálogo de produtos. Felizmente, já temos um arquivo JSON contendo as informações acerca destes produtos.

Na pasta "`_Recursos`" da Solução há uma subpasta chamada "dados", que por sua vez contém "`livros.json`", com código, nome e preço de cada um dos produtos, no caso, livros. Faremos uma cópia deste arquivo no nosso projeto "`CasaDoCodigo`".

Como iremos ler este arquivo `.json`?

Poderemos utilizar a classe `File` do .NET para acessá-lo. Importaremos a referência (o *namespace*), e faremos a leitura de todo o texto do arquivo, com `ReadAllText()`. Passaremos o nome do arquivo como parâmetro deste método e precisaremos armazenar o texto em uma variável denominada `json`.

Para realizarmos a conversão deste JSON em uma lista de objetos a serem utilizados para alimentar o banco de dados, usaremos uma biblioteca bem conhecida no .NET, o **Newtonsoft**. Para isto, chamaremos a classe desta biblioteca e o método para desserializar e transformar o texto em objetos, passando o tipo para o qual queremos converter como parâmetro, mas antes disso criaremos um tipo chamado `Livro`.

Coparemos os campos do arquivo `livros.json` e criaremos três propriedades automáticas para esta nova classe, definindo-os com `{ get; set; }`. Feito isto, utilizaremos `Livro` para desserializar o arquivo JSON, passando o tipo, no caso uma lista de `Livro`. Passaremos a nossa *string* como parâmetro, que é a variável `json`, e então armazenaremos este conteúdo em uma lista, um novo objeto, chamado `livros`. Incluiremos também um *breakpoint* na última linha do método.

```
class DataService : IDataService
{
    private readonly ApplicationContext contexto;

    public DataService(ApplicationContext contexto)
    {
        this.contexto = contexto;
    }

    public void InicializaDB()
    {
        contexto.Database.EnsureCreated();

        var json = File.ReadAllText("livros.json");
        var livros = JsonConvert.DeserializeObject<List<Livro>>(json);
    }
}
```

```
class Livro
{
    public stringCodigo { get; set; }
    public stringNome { get; set; }
    public decimalPreco { get; set; }
}
```

Vamos rodar a aplicação para verificar se a conversão é bem sucedida, iniciando pela criação do banco e, em seguida, pela carga do arquivo `livros.json`. Após a passagem pelo *breakpoint*, na linha 22, acessaremos o conteúdo de `livros` e veremos as informações de código, nome e preço, como gostaríamos.

Adiante, começaremos a importar estes dados para o nosso banco de dados.