

Recuperando valores das shadow properties

Transcrição

Nesta aula, iremos entender como ler o valor de uma shadow property. Na classe `Program` eliminaremos as propriedades `PrimeiroNome`, `UltimoNome` e `last_update`.

```
namespace Alura.Fulmes.App
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var contexto = new AluraFilmesContexto())
            {
                contexto.LogSQLToConsole();

                var ator = new Ator();
                ator.PrimeiroNome = "Tom";
                ator.UltimoNome = "Hanks";
                contexto.Entry(ator).Property("last_update").CurrentValue = DateTime.Now;

                contexto.Atores.Add(ator);

                contexto.SaveChanges();
            }
        }
    }
}
```

Depois de eliminada a parte do código referenciada, iremos adicionar à propriedade `Atores` o primeiro ator da tabela. Iremos, ainda, imprimir `ator` e o valor que está na propriedade `last_update`.

```
namespace Alura.Filmes.App
{
    class program
    {
        static void Main(string[] args)
        {
            using (var contexto = new AluraFilmesContexto())
            {
                contexto.LogSQLToConsole();

                var ator = contexto.Atores.First();

                Console.WriteLine(ator);
                Console.WriteLine(contexto.Entry(ator).Property("last_update").CurrentValue);

                contexto.SaveChanges();
            }
        }
    }
}
```

```

    }
}
}

```

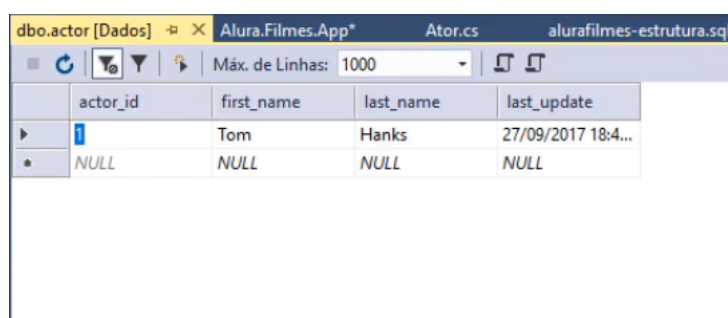
Feito isso, pressionaremos "F5" para executar o programa. Será exibida a mensagem:

`A data reader was disposed.

Ator (1) : Tom Hanks 27/09/2017 18:42:44`

Temos o ator Tom Hanks e a propriedade last_update impressa através do método Entry() .

Ao pensar nos usos da propriedade last_update , como faremos se quisermos listar os dez últimos atores que sofreram modificação recentemente? Para isso, trabalharemos com o banco de dados que possui um maior número de atores(AluraFilmes), pois o banco que estamos trabalhando (AluraTST) só possui um ator.



actor_id	first_name	last_name	last_update
1	Tom	Hanks	27/09/2017 18:42:44
NULL	NULL	NULL	NULL

Para realizarmos essa modificação de banco de dados, na parte superior da tela, iremos selecionar a classe AluraFilmesContexto.cs e modificaremos no SqlServer o Database de AluraFlmesTST para AluraFilmes .

```

namespace Alura.Filmes.App.Dados
{
    public class AluraFilmesContexto : DbContext
    {
        public DbSet<Ator> Atores { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Server=(localdb)mssqllocaldb;Database=AluraFilmes;Trusted_Connection=true;")
        }
    }
}

<!-- ... -->

```

Voltaremos para a classe Program . Precisaremos usar uma classe estática do Entity Framework Core para poder acionar os valores dos dez atores que foram modificados recentemente. Nosso código está assim até o momento:

```

namespace Alura.Filmes.App
{
    class program
    {
        static void Main(string[] args)
        {

```

```

    {
        using (var contexto = new AluraFilmesContexto())
        {
            contexto.LogSQLtoConsole();

            var ator = contexto.Atores.First();

            Console.WriteLine(ator);
            Console.WriteLine(contexto.Entry(ator).Property("last_update").CurrentValue);

            contexto.SaveChanges();
        }
    }
}

```

Removeremos a parte do código que diz respeito à variável `ator` e à propriedade `last_update`. Criaremos uma variável denominada `atores`, em que iremos fazer uma *query* em cima do conjunto de atores, que é a propriedade `DbSet`. Faremos um comentário nessa parte do código para nos organizarmos. Queremos selecionar os dez primeiros atores ordenados pela coluna `last_update` de forma decrescente. Usaremos, portanto, o método `OrderByDescending()`, acompanhado de uma expressão lambda em que `a` em que o argumento de entrada é uma instância do `ator`. Usaremos uma classe estática do Entity Framework Core denominada `EF` que possui múltiplas funções. A função iremos utilizar neste caso é a `Property` do tipo `DateTime` que está na instância `ator` - representada pela variável `a` - cujo nome é `last_update`. Como queremos os últimos dez atores, utilizaremos o método `Take()`.

```

namespace Alura.Filmes.App
{
    class program
    {
        static void Main(string[] args)
        {
            using (var contexto = new AluraFilmesContexto())
            {
                contexto.LogSQLtoConsole();

                //listar os 10 atores modificados recentemente
                var atores = contexto.Atores
                    .OrderByDescending(a => EF.Property<DateTime>(a, "last_update"))
                    .Take(10);
            }
        }
    }
}

```

Iremos listar os atores e solicitar que o `ator` esteja separado pela propriedade `last_update`. Feito isso, adicionaremos o valor.

```

namespace Alura.Filmes.App
{
    class program
    {
        static void Main(string[] args)
        {

```

```

        using (var contexto = new AluraFilmesContexto())
        {
            contexto.LogSQLtoConsole();

            //listar os 10 atores modificados recentemente
            var atores = contexto.Atores
                .OrderByDescending(a => EF.Property<DateTime>(a, "last_update"))
                .Take(10);

            foreach (var ator in atores)
            {
                Console.WriteLine(ator + " - " + contexto.Entry(ator).Property("last_update").Cur
            }
        }
    }
}

```

Ao iniciarmos o programa veremos o resultado da execução do código.

```

Executed DbCommand (39ms) [Parameters=[@__p_0='?'], CommandType='Text', CommandTimeout='30']
SELEC TOP(@__p_0) [a].[actor_id], [a].[first_name], [a].[last_name], [a].[last_update]
FROM [acotr] AS [a]
ORDER BY [a].[last_update] DESC

```

```

Executed DbCommand (39ms) [Parameters=[@_p_0='?'], CommandType='Text', CommandTimeout = '30']
SELECT TOP()(@_p_0) [a].[actor_id], [a].[first_name], [a].[last_name], [a].[last_update]
FROM [acotr] AS [a]
ORDER BY [a].[last_update] DESC

```

```

Ator (13): UMA WOOD - 08/10/2017 00:00:00
Ator (201): MEL GIBSON - 29/08/2017 16:36:23
Ator (92): KIRSTEN AKROYD - 12/08/2017 00:00:00
Ator (58): CHRISTIAN AKROYD - 12/08/2017 00:00:00
Ator (76): ANGELINA ASTAIRE - 12/08/2017 00:00:00
Ator (59): DUSTIN TAUTOU - 08/02/2017 00:00:00
Ator (18): DAN TORN - 08/02/2017 00:00:00
Ator (129): DARYL CRAWFORD - 08/02/2017 00:00:00
Ator (56): DAN HARRIS - 08/02/2017 00:00:00
Ator (95): DARYL WAHLBERG - 08/02/2017 - 00:00:00

```

```

A data reader was disposed.

```

Vimos que os valores estão sendo exibidos e ordenados de forma decrescente dos modificados mais recentemente. Essa é a maneira que podemos utilizar shadow properties no nosso código.

