

Začínáme s EF Core

Článek • 11. 01. 2023 • 3 min ke čtení

V tomto kurzu vytvoříte konzolovou aplikaci .NET Core, která provádí přístup k datům v databázi SQLite pomocí Entity Framework Core.

Tento kurz můžete postupovat pomocí sady Visual Studio ve Windows nebo pomocí rozhraní příkazového řádku .NET Core ve Windows, macOS nebo Linuxu.

[Podívejte se na ukázkou tohoto článku na GitHubu](#) .

Požadavky

Nainstalujte následující software:

.NET Core CLI

- [.NET Core SDK](#) .

Vytvoření nového projektu

.NET Core CLI

.NET CLI

```
dotnet new console -o EFGetStarted
cd EFGetStarted
```

Instalace Entity Framework Core

Pokud chcete nainstalovat EF Core, nainstalujete balíček pro poskytovatele databází EF Core, na které chcete cílit. Tento kurz používá SQLite, protože běží na všech platformách, které podporuje .NET Core. Seznam dostupných poskytovatelů najdete v tématu [Poskytovatelé databází](#).

.NET Core CLI

.NET CLI

```
dotnet add package Microsoft.EntityFrameworkCore.Sqlite
```

Vytvoření modelu

Definujte třídu kontextu a třídy entit, které tvoří model.

.NET Core CLI

- V adresáři projektu vytvořte **soubor Model.cs** s následujícím kódem.

C#

```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;

public class BloggingContext : DbContext
{
    public DbSet<Blog> Blogs { get; set; }
    public DbSet<Post> Posts { get; set; }

    public string DbPath { get; }

    public BloggingContext()
    {
        var folder = Environment.SpecialFolder.LocalApplicationData;
        var path = Environment.GetFolderPath(folder);
        DbPath = System.IO.Path.Join(path, "blogging.db");
    }

    // The following configures EF to create a Sqlite database file in the
    // special "local" folder for your platform.
    protected override void OnConfiguring(DbContextOptionsBuilder options)
        => options.UseSqlite($"Data Source={DbPath}");
}

public class Blog
{
    public int BlogId { get; set; }
```

```
public string Url { get; set; }

public List<Post> Posts { get; } = new();
}

public class Post
{
    public int PostId { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }

    public int BlogId { get; set; }
    public Blog Blog { get; set; }
}
```

EF Core může také [zpětně analyzovat](#) model z existující databáze.

Tip: Tato aplikace záměrně udržuje věci jednoduché pro přehlednost. [Připojovací řetězce](#) by neměly být uloženy v kódu pro produkční aplikace. Každou třídu jazyka C# můžete také rozdělit do vlastního souboru.

Vytvoření databáze

V následujících [krocích se migrace](#) používají k vytvoření databáze.

.NET Core CLI

- Spustíte následující příkazy:

.NET CLI

```
dotnet tool install --global dotnet-ef
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet ef migrations add InitialCreate
dotnet ef database update
```

Tím se nainstaluje [dotnet ef](#) a balíček návrhu, který se vyžaduje ke spuštění příkazu v projektu. Příkaz `migrations` vygeneruje migraci a vytvoří počáteční sadu tabulek pro model. Příkaz `database update` vytvoří databázi a použije na ni novou migraci.

Vytvoření, čtení, aktualizace & odstranění

- Otevřete soubor *Program.cs* a nahraďte obsah následujícím kódem:

C#

```
using System;
using System.Linq;

using var db = new BloggingContext();

// Note: This sample requires the database to be created before running.
Console.WriteLine($"Database path: {db.DbPath}.");

// Create
Console.WriteLine("Inserting a new blog");
db.Add(new Blog { Url = "http://blogs.msdn.com/adonet" });
db.SaveChanges();

// Read
Console.WriteLine("Querying for a blog");
var blog = db.Blogs
    .OrderBy(b => b.BlogId)
    .First();

// Update
Console.WriteLine("Updating the blog and adding a post");
blog.Url = "https://devblogs.microsoft.com/dotnet";
blog.Posts.Add(
    new Post { Title = "Hello World", Content = "I wrote an app using EF Core!" });
db.SaveChanges();

// Delete
Console.WriteLine("Delete the blog");
db.Remove(blog);
db.SaveChanges();
```

Spuštění aplikace

.NET Core CLI

.NET CLI

```
dotnet run
```

Další kroky

- Pokud chcete použít EF Core ve webové aplikaci, postupujte podle [kurzu ASP.NET Core](#).
- Další informace o [výrazech dotazů LINQ](#)
- [Nakonfigurujte model](#) tak, aby určoval například [požadovanou](#) a [maximální délku](#).
- Použití [migrace](#) k aktualizaci schématu databáze po změně modelu