

Tworzenie pustej aplikacji webowej z obsługą Entity Framework i migracji dla MySQL przy użyciu lokalnego dotnet-ef

1. Tworzenie nowego projektu

```
dotnet new web -n MyWebApp
cd MyWebApp
```

2. Dodanie pakietów NuGet

Dodaj niezbędne pakiety NuGet do projektu:

```
dotnet add package Microsoft.EntityFrameworkCore
dotnet add package Pomelo.EntityFrameworkCore.MySql
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.Tools
```

3. Tworzenie modelu danych

Utwórz folder **Models** i dodaj klasę **Employee**:

```
// Models/Employee.cs
namespace MyWebApp.Models
{
    public class Employee
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Position { get; set; }
        public decimal Salary { get; set; }
    }
}
```

4. Tworzenie kontekstu bazy danych

Utwórz folder **Data** i dodaj klasę **AppDbContext**:

```
// Data/AppDbContext.cs
using Microsoft.EntityFrameworkCore;
using MyWebApp.Models;

namespace MyWebApp.Data
{

```

```
public class AppDbContext : DbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options) :
base(options) { }

    public DbSet<Employee> Employees { get; set; }
}
```

5. Konfiguracja połączenia z bazą danych

Edytuj plik `appsettings.json`, aby dodać łańcuch połączenia do MySQL:

```
// appsettings.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=localhost;Database
=my
webapp;User=root;Password=yourpassword;"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

6. Rejestracja kontekstu bazy danych

Edytuj plik `Program.cs`, aby zarejestrować kontekst bazy danych:

```
// Program.cs
using Microsoft.EntityFrameworkCore;
using MyWebApp.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<AppDbContext>(options =>
options.UseMySQL(builder.Configuration.GetConnectionString("DefaultConnection"),
    new MySQLServerVersion(new Version(8, 0, 21))));
```

```
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

7. Instalacja narzędzi EF

Zainstaluj narzędzia EF lokalnie:

```
dotnet tool install --global dotnet-ef
```

8. Tworzenie migracji i aktualizacja bazy danych

Utwórz migrację i zaktualizuj bazę danych:

```
dotnet ef migrations add InitialCreate
dotnet ef database update
```

9. Uruchomienie aplikacji

Uruchom aplikację:

```
dotnet run
```

To wszystko! Teraz masz pustą aplikację webową z obsługą Entity Framework i migracji dla MySQL.

Similar code found with 2 license types