

ASP.NET Active Dynamic Web Pages

עבודה עם Entity Framework

TomerBu

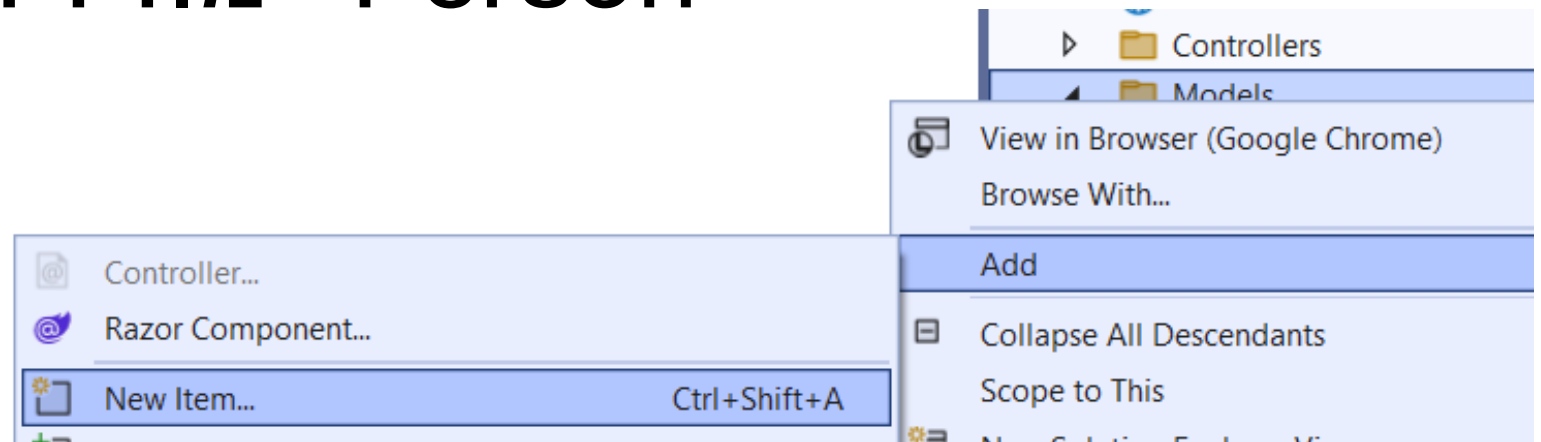
Entity Framework:

מיפוי של אובייקטים לדטה-בייס רלציוני:

Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. It eliminates the need for most of the data-access code that developers usually need to write.

ASP.NET Core MVC

שדל חדל – Person



```
using System.ComponentModel.DataAnnotations;

namespace Lec2.Models;

public class Person
{
    [Key]//Primary Key
    public int Id { get; set; }

    [Required]
    [MinLength(1)]
    [MaxLength(20)]
    public string Name { get; set; }
}
```

התקנת ספריות לעבודה עם Entity Framework חבילות NuGet לדטה-בייס:



Microsoft.EntityFrameworkCore by Microsoft

Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.



Microsoft.EntityFrameworkCore.Design by Microsoft

Shared design-time components for Entity Framework Core tools.



Microsoft.EntityFrameworkCore.SqlServer by Microsoft

Microsoft SQL Server database provider for Entity Framework Core.



Microsoft.EntityFrameworkCore.Tools by Microsoft

Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.

ASP.NET Core MVC

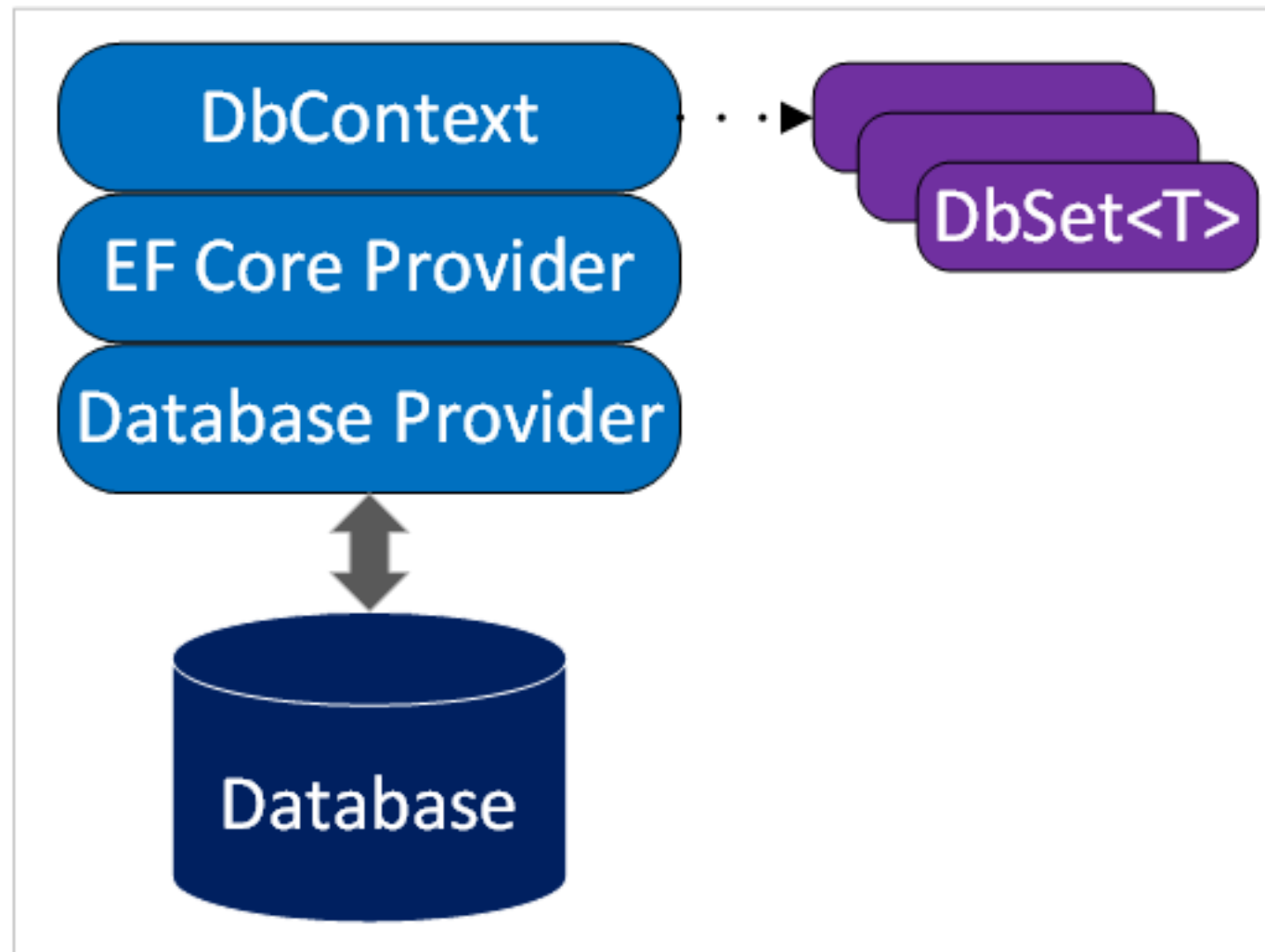
ביי-ס-הגדרות לדטה:

appsettings.json

```
-
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}

{
  "ConnectionStrings": {
    "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=aspnetcore;Trusted_Connection=True;MultipleActiveResultSets=true"
  }
}
```

Entity Framework



Entity Framework

מחלקה עבור DbContext

Data/Lec2DbContext.cs

ניצור תיקיה בשם Data ובתוכה class חדש:

```
namespace Lec2.Data;

using Lec2.Models;
using Microsoft.EntityFrameworkCore;

public class Lec2DbContext(DbContextOptions<Lec2DbContext> options) : DbContext(options)
{
    //ORM = Object Relational Mapper
    public DbSet<Person> People { get; set; }
}
```

הזרקה של ה DbContext שלנו:

הוספת DbContext לDI Container

Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        var builder = WebApplication.CreateBuilder(args);

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

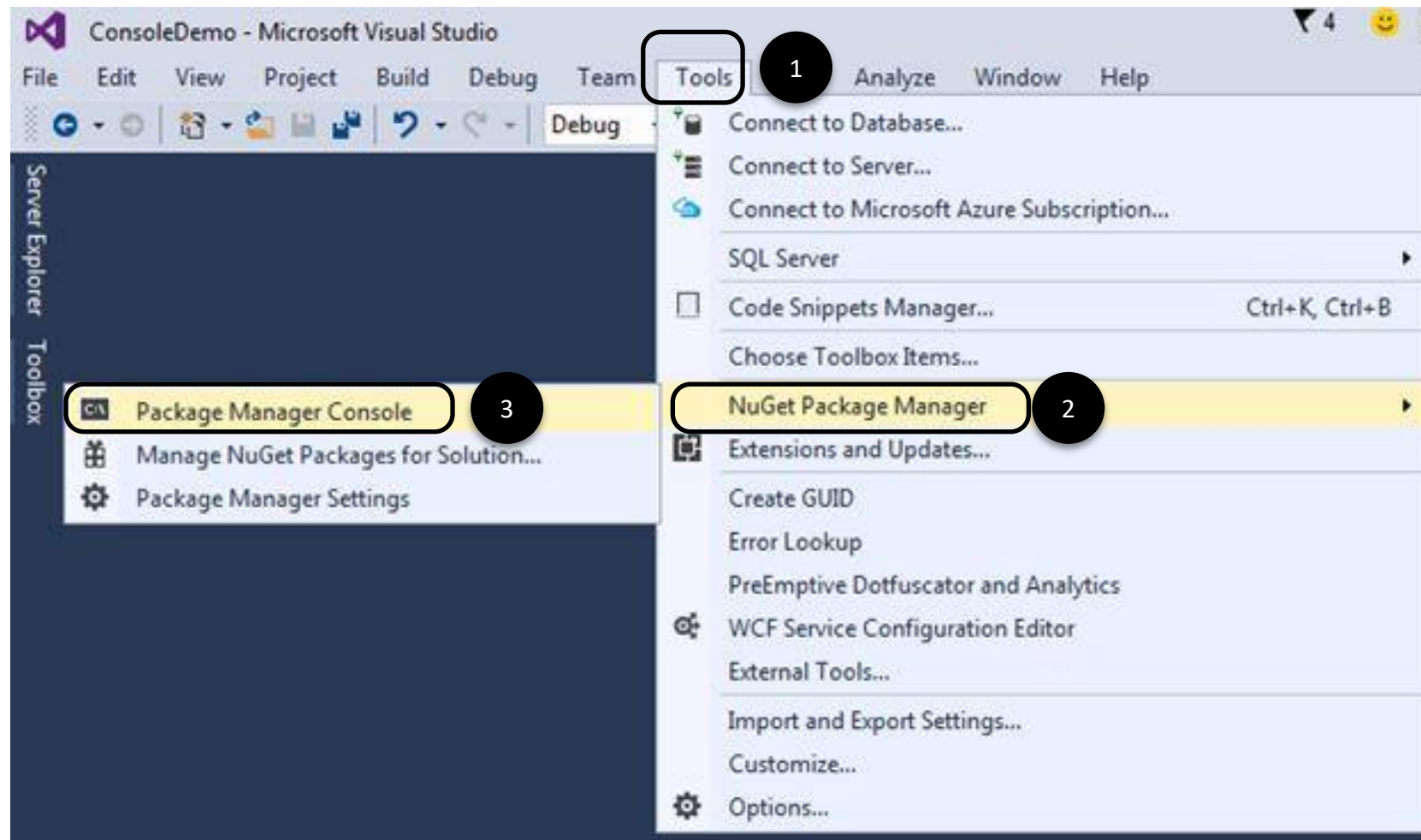
        //get the connection string from appsettings.json:
        var connectionString = builder.Configuration.GetConnectionString("MSSQL");

        //add our db context to the di container:
        //Now we can inject our dbContext everywhere.

        builder.Services.AddDbContext<Lec2DbContext>(o => o.UseSqlServer(connectionString));

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


הרצה של סקריפט ליצירת דטה-בייס ב Nuget Package Manager Console



```
PM> update-database
```

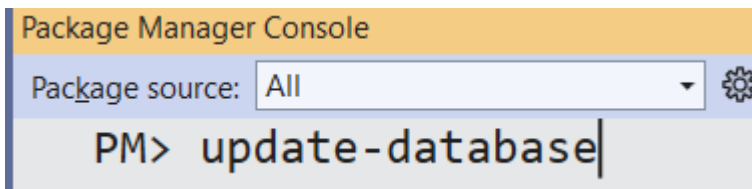
Below the command, there are tabs for 'List', 'Output', and 'Package Manager Console'.

זהו זה – יש לנו דטהבייס!

הרצה של סקריפט ליצירת דטה-בייס ב Nuget Package Manager Console

PM> add-migration AddPeopleTable

1



2

כדי ליצור את הטבלה –ניצור מיגרציה
ואז נעדכן את הדטה-בייס:

עכשיו נוצרו הטבלאות בדטה-בייס!

מילוי ערכים בדטה-בייס SSMS : אחרי שהטבלה נוצרה - נמלא כמה רשומות לדוגמה:

```
INSERT INTO People (Name) VALUES ('John'), ('Mike'), ('Sally');
```

```
SELECT * FROM People;
```

PeopleController.cs

עבודה עם DbContext - נזריק בבנאי ונשתמש במתודות המובנות לעבודה עם הדטה-בייס

```
using Lec2.Data;
using Lec2.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace Lec2.Controllers;

public class PeopleController(Lec2DbContext lec2Db) : Controller
{
    public async Task<IActionResult> Index()
    {
        var people = await lec2Db.People.ToListAsync();
        return View(people);
    }
}
```

הצגת המידע בView

@model List<Person>

Views/People/Index.cshtml

```
<h1>People</h1>
```

```
<table class="table table-striped">
```

```
  <thead>
```

```
    <tr>
```

```
      <th>Id</th>
```

```
      <th>Name</th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    @foreach (var p in Model)
```

```
    {
```

```
      <tr onclick="window.location.href='/People/Details/@p.Id'">
```

```
        <td>@p.Id</td>
```

```
        <td>@p.Name</td>
```

```
      </tr>
```

```
    }
```

```
  </tbody>
```

```
</table>
```

הצגת פרטים של Person:

```
using Lec2.Data;
using Lec2.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace Lec2.Controllers;

public class PeopleController(Lec2DbContext lec2Db) : Controller
{
    public async Task<IActionResult> Index()
    {
        var people = await lec2Db.People.ToListAsync();
        return View(people);
    }

    public async Task<IActionResult> Details(int id)
    {
        Person? person = await lec2Db.People.FirstOrDefaultAsync(p => p.Id == id);
        return View(person);
    }
}
```

הצגת המידע בView

Views/People/Details.cshtml

```
@model Person
```

```
<h1>Person Details</h1>
```

```
<p>Id: @Model.Id</p>
```

```
<p>Name: @Model.Name</p>
```

שיעורי בית:

1. צרו פרויקט MVC חדש בשם MoviesProject
2. צרו מודל לסרט -תכונות הסרט Id, title, overview, imageUrl -
3. הוסיפו את הספריות של Entity Framework לפרוייקט
4. צרו מחלקה בשם MoviesDbContext שיורשת מ DbContext עם תכונה DbSet<Movie>
5. בקונסולה של - nuget הריצו את הפקודות update-database ליצור דטה בייס
6. בקונסולה של nuget הריצו את הפקודה CreateTableMovies add-migration להוספת הטבלה
7. בקונסולה של - nuget הריצו את הפקודה update-database לבצע את יצירת הטבלה.
8. צרו Controller להצגת מידע על סרטים
9. צרו View בשם Index.cshtml בתיקיה Movies להצגת כל הסרטים
10. בלחיצה על קישור יש לעבור לעמוד פרטי הסרט ולהציג את הסרט בעמוד בודד.
11. הוסיפו את ה Controller שלכם לסרגל הניווט