Asp.Net Core CheatSheet

Index

Start a new project

Task	CLI Command
Create new console application	dotnet new webapp -o aspnetcoreapp
Run the app	cd aspnetcoreapp dotnet run
Create local Certificate	dotnet dev-certs https --trust

back to top

Configuration files

Development

1. Appsettings.json - Development `Non-Private`
2. Manage User Secrets - Development `Private`

back to top

Environment Variables

1. project => projectName properties => Debug

back to top

How to access Config data and Environmental Variables

1. @inject Microsoft.Extensions.Cionfiguration.IConfiguration configuration
2. Use @configuration["KEY"]

back to top

Tag Helpers

1. To switch them on add ``` @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers ``` to _ViewImports.cshtml.

Built in Tag Helpers

back to top

Create a Model

1. In Solution Explorer, right-click the RazorPagesMovie project > Add > New Folder. Name the folder Models.

2. Right click the Models folder. Select Add > Class. Name the class Movie and replace the contents of the Movie class with the following code:

```
using System;
using System.ComponentModel.DataAnnotations.Schema;

namespace RazorPagesMovie.Models
```

```
{
    public class Movie
    {
        public int ID { get; set; }
        public string Title { get; set; }
        public DateTime ReleaseDate { get; set; }
        public string Genre { get; set; }
        public decimal Price { get; set; }
    }
}
```

back to top

Create Razor Pages and EntityFramework

In Solution Explorer, right click on the `Pages folder > Add > New Folder.` Name the folder Movies

1. In Solution Explorer, right click on the `Pages/Movies folder > Add > New Scaffolded Item.`
2. In the Add Scaffold dialog, `select Razor Pages using Entity Framework (CRUD) > Add.`
3. Complete the Add Razor Pages using Entity Framework (CRUD) dialog:

> “In the Model class drop down, select Movie (RazorPagesMovie.Models). In the Data context class row, select the + (plus) sign and accept the generated name RazorPagesMovie.Models.RazorPagesMovieContext. Select Add.”

The scaffold process creates and updates the following files:

```
Files created
Pages/Movies: Create, Delete, Details, Edit, Index.
Data/RazorPagesMovieContext.cs
```

back to top

Dependency Injection

1. The scaffolding tool automatically created a `DB context` and registered it with the dependency injection container. In the `Startup.ConfigureServices`

```
services.AddDbContext<RazorPagesMovieContext>(options =>
            options.UseSqlServer(Configuration.GetConnectionString("RazorPagesMovieContext")));
```

The main page which coordinates Entity framework is the `DbContext class`

Here it's called RazorPagesMovieContext

Entity set created `DbSet<Movie>` which corresponds with a database table.

The `connection string` comes from `DbContextOptions` this is read from a `config file` for example `appsettings.json`

back to top