

ASP.NET Core MVC

Paulo Vieira

2020

Create a new MVC app

```
dotnet new mvc --auth Individual
```

Use authentication DB for our data?

```
sqlite3 app.db < StarterStore.sql
```

Use EntityFramework and Scaffolding

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

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

```
dotnet add package Microsoft.EntityFrameworkCore.Design
```

```
dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design
```

Scaffolding models from existing database

```
dotnet-ef dbcontext scaffold --context-dir Data  
--context "ApplicationDbContext" "DataSource=app.db"  
Microsoft.EntityFrameworkCore.Sqlite --output-dir Models --force
```

Scaffold Controllers

```
dotnet tool install -g dotnet-aspnet-codegenerator
```

```
dotnet-aspnet-codegenerator controller --controllerName CustomersController  
--model Customers -outDir "Controllers"  
--dataContext "ApplicationDbContext" -sqlite
```

Reminding MVC

- **Models:** Classes that represent the data entities and view models used in the website.
- **Views:** Razor files, that is, .cshtml files, that render data in view models into HTML web pages.

Reminding MVC

- **Controllers:** Classes that execute code when an HTTP request arrives at the web server. The code usually creates view model that may contain entity models and passes it to a view to generate an HTTP response to send back to the web browser or other client.

Anatomy of a **ASP.NET** Core MVC Application

- **Areas:** This folder contains files needed for features like **ASP.NET** Core Identity for authentication.
- **bin, obj:** These folders contain the compiled assemblies for the project.
- **Controllers:** This folder contains C# classes that have methods (known as actions) that fetch a model and pass it to a view

Anatomy of a **ASP.NET** Core MVC Application

- **Data:** This folder contains Entity Framework Core migration classes used by the **ASP.NET** Core Identity system to provide data storage for authentication and authorization.
- **Models:** This folder contains C# classes that represent all of the data gathered together by a controller and passed to a view.

Anatomy of a **ASP.NET** Core MVC Application

- **Properties:** This folder contains a configuration file for IIS and launching the website during development named `launchSettings.json`.
- **Views:** This folder contains the `.cshtml` Razor files that combine HTML and C# code to dynamically generate HTML responses.

[Price, 2019]