My Azure DB Server name is dietrichsql.database.windows.net.

# ASP.NET MVC (.NET Framework)

Created project WebSongs1 on 22-Jan-24, as anASP.NET Web Application (.NET Framework), subtype MVC.

Started building it according to MS tutorial “Getting started with ASP.NET MVC 5”, at <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started> .

Noted the same confusion as when I first started teaching myself ASP.NET on 16-Feb=23: unlike Winforms, or straight HTML, you don’t get full call stack – a full enough view of what code is ultimately being called to bring up what’s on the screen and respond to keyboard and mouse activity.

@\* … \*@ to comment out CSHTML code.

## To link project to existing DB:

First, go to NuGet Package Manager and install Microsoft.EntityFrameworkCore.Design package. As well as Microsoft.EntityFrameworkCore, and .SqlServer.

(From <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/database-first-development/creating-the-web-application> ).

Make sure if it’s a view, that it’s got an inferable Primary Key! If it doesn’t:

* Add [Key] to the column definition in the Model module, and add **using System.ComponentModel.DataAnnotations;** to the top
* Alter the view by wrapping the key in ISNULL statement, so it can infer by non-null status.)
* Remove **.HasNoKey()** from the DB Context file’s definition of the view!

Right click Models, Add Item, ADO .NET Entity Data Model.

When adding controllers, in Add Scaffolding dialog box, make sure Generate Views checkbox is checked! That’s not mentioned in the tutorial!

# ASP.NET Core

Started project WebCoreSongs on 24-Jan-24.

Be sure the browse method next to the Start (green arrow) button is IIS Express, rather the https – when you’re starting the project. Or http. https will give you a security error message.

Used the online docs linked by the VS project:

[Get started with ASP.NET Core MVC | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio)

[Productivity tips for .NET developers - Visual Studio (Windows) | Microsoft Learn](https://learn.microsoft.com/en-us/visualstudio/ide/csharp-developer-productivity?view=vs-2022&utm_source=VisualStudio&utm_medium=aspnet-getstarted&utm_campaign=VisualStudio)

[.NET Application Architecture Guides (microsoft.com)](https://dotnet.microsoft.com/en-us/learn/dotnet/architecture-guides?utm_source=aspnet-start-page&utm_campaign=vside)

[Quickstart: Deploy an ASP.NET web app - Azure App Service | Microsoft Learn](https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net70&pivots=development-environment-vs#launch-the-publish-wizard?utm_source=aspnet-start-page&utm_campaign=vside)

[Azure for .NET developers | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/azure/?utm_source=aspnet-start-page&utm_campaign=vside)

## Link to existing DB:

First, go to NuGet Package Manager and install Microsoft.EntityFrameworkCore.Design package. As well as Microsoft.EntityFrameworkCore, and …SqlServer too, right?

In NuGet Console:

Scaffold-DbContext "Data Source=tcp:dietrichsql.database.windows.net,1433;Initial Catalog=Songbook; TrustServerCertificate=False;Connection Timeout=60; User ID=dietrichuser;Password=algonQin88; ApplicationIntent=ReadWrite; MultiSubnetFailover=False" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models/DB -Table Songbook.Songs

## Log of More work: 1/26/2024

Downloaded EF Core Power Tools, from https://marketplace.visualstudio.com/items?itemName=ErikEJ.EFCorePowerTools

Made a **Blazor app**. Also brought up View SQL Server Explorer on both projects… not that that will do me necces any good

## Generate Model

First, go into SSMS to establish a connection to the Azure database.

Right-click on project and select the EF Core Power Tools> Reverse Engineer.

Add database connection = server name dietrichsql.database.windows.net, Authent = Microsoft Entra MFA.

**MAKE SURE to include the tables that you’ve scaffolded before! Otherwise it will delete the code for them! BUT – if you include them, it will wipe out the changes you made since scaffolding! So how do you get around this godawful misdesign? Other than go into Git and cancel the unwanted changes?**

Set context to SongsContext. Because it uses the same Context for all tables and views.

Leave “Use table and column names directly from the database” unchecked! It will screw up other tables if you check it!

In Choose Your Settings, set a tbl-specific Context Name, and uncheck “Pluralize or singularize generated object names”!

## Generate View & Controller

To add view:

For the first table, go into Program.cs and add a line to register the new Controller. It should look like this:

builder.Services.AddDbContext<WebCoreSongs.Models.SongsContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("AZURE\_SQL\_CONNECTIONSTRING")));

In **Solution Explorer**, right-click the Controllers folder and select **Add > New Scaffolded Item**.

In the **Add New Scaffolded Item** dialog:

* In the left pane, select **Installed** > **Common** > **MVC**.
* Select **MVC Controller with views, using Entity Framework**.
* Select **Add**.

Complete the **Add MVC Controller with views, using Entity Framework** dialog:

* In the **Model class** drop down, select the model.
* In the **DB context class** row, select SongsContext.
* Leave **Database provider** set to “Configured from the selected DBContext”.
* **Views** and **Controller name**: Keep the default.
* Select **Add**.

MS Doc says “If you get an error message, select **Add** a second time to try it again.”

That worked for Viewartistnameforlistbox, but I got an error creating the View for the model – it can’t determine the primary key.

## App Deployment

Finally succeeded 19Feb24.

Publish with VS – Build | Publish command; select Azure Web Services.   
Be sure the .yml (web app deployment config) file (in this case, azure-static-web-apps-blue-beach-06a53a01e.yml in C:\Dietrich\GithubDocs\DietrichDocuments\.github\workflows) lists the correct location settings. Note that output\_location is NOT optional.

app\_location: "WebCoreSongs/WebCoreSongs" # App source code path - solution & proj!

api\_location: "" # Api source code path - optional

output\_location: "/build" # Built app content directory - optional

# Login and Security

Evidently, Managed identities can be used to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

You need a virtual machine to connect either a User or System managed identity to it, apparently.