AZ-204

Developing solutions for Microsoft Azure

Lab 04 – Part 2

Constructing a polyglot data solution

Table of Contents

1 Pre-requisites

1.1 Sign in to the lab virtual machine

Sign in to your Windows 10 virtual machine (VM) by using the following credentials:

- Username: **Admin**
- Password: **Pa55w.rd**

Note: Instructions to connect to the virtual lab environment will be provided by your instructor.

1.2 Review the installed applications

Find the taskbar on your Windows 10 desktop. The taskbar contains the icons for the applications that you'll use in this lab:

- Microsoft Edge
- File Explorer
- Azure CLI
- Windows PowerShell
- **Complete Lab-04-Part 1 - Prerequisite**

2 Exercise 4: Migrating SQL data to Azure Cosmos DB

2.1 Task 1: Create a migration project

1. In the **Visual Studio Code** window, access the shortcut menu or right-click or activate the shortcut menu for the Explorer pane, and then select **Open in Terminal**.

   Navigate to **LabFiles\Allfiles\Labs\04\Starter\AdventureWorks** in the terminal/command prompt

2. At the open command prompt, enter the following command, and then select Enter to create a new .NET console project named **AdventureWorks.Migrate** in a folder with the same name:

```
 dotnet new console --name AdventureWorks.Migrate
```

Note: The **dotnet new** command will create a new **console** project in a folder with the same name as the project.

3. At the command prompt, enter the following command, and then select Enter to add a reference to the existing **AdventureWorks.Models** project:

```
 dotnet add
.\AdventureWorks.Migrate\AdventureWorks.Migrate.csproj
reference .\AdventureWorks.Models\AdventureWorks.Models.csproj
```

Note: The **dotnet add reference** command will add a reference to the model classes contained in the **AdventureWorks.Models** project.

4. At the command prompt, enter the following command, and then select Enter to add a reference to the existing **AdventureWorks.Context** project:

```
 dotnet add
.\AdventureWorks.Migrate\AdventureWorks.Migrate.csproj
reference
.\AdventureWorks.Context\AdventureWorks.Context.csproj
```

Note: The **dotnet add reference** command will add a reference to the context classes contained in the **AdventureWorks.Context** project.

5. At the command prompt, enter the following command, and then select Enter to switch your terminal context to the **AdventureWorks.Migrate** folder:

```
  cd .\AdventureWorks.Migrate\
```

6. At the command prompt, enter the following command, and then select Enter to import version 2.2.6 of **Microsoft.EntityFrameworkCore.SqlServer** from NuGet:

```
 dotnet add package Microsoft.EntityFrameworkCore.SqlServer --
version 3.0.1
```

Note: The **dotnet add package** command will add the **Microsoft.EntityFrameworkCore.SqlServer** package from **NuGet**. For more information, go to: Microsoft.EntityFrameworkCore.SqlServer.

7. At the command prompt, enter the following command, and then select Enter to import version 3.4.1 of **Microsoft.Azure.Cosmos** from NuGet:

```
dotnet add package Microsoft.Azure.Cosmos --version 3.4.1
```

Note: The **dotnet add package** command will add the **Microsoft.Azure.Cosmos** package from **NuGet**. For more information, go to: Microsoft.Azure.Cosmos.

8. At the command prompt, enter the following command, and then select Enter to build the .NET console application:

```
dotnet build
```

8. Select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

2.2 Task 2: Create a .NET class

1. In the Explorer pane of the **Visual Studio Code** window, expand the **AdventureWorks.Migrate** project.
2. Open the **Program.cs** file.
3. From the code editor tab for the **Program.cs** file, delete all the code in the existing file.
4. Add the following lines of code to import the **AdventureWorks.Models** and **AdventureWorks.Context** namespaces from the referenced **AdventureWorks.Models** and **AdventureWorks.Context** projects:

```
using System;
using AdventureWorks.Context;
using AdventureWorks.Models;
```

5. Add the following line of code to import the **Microsoft.Azure.Cosmos** namespace from the **Microsoft.Azure.Cosmos** package imported from NuGet:

```
using Microsoft.Azure.Cosmos;
```

6. Add the following line of code to import the **Microsoft.EntityFrameworkCore** namespace from the **Microsoft.EntityFrameworkCore.SqlServer** package imported from NuGet:

```
using Microsoft.EntityFrameworkCore;
```

7. Add the following lines of code to add **using** directives for the built-in namespaces that will be used in this file:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
```

8. Within the **Program** class, enter the following line of code to create a new string constant named **sqlDBConnectionString**:

```
private const string sqlDBConnectionString = "";
```

9. **Update the **sqlDBConnectionString** string constant by setting its value to the **ADO.NET (SQL Authentication) connection string** of the SQL database that you recorded earlier in this lab.

Note: It's important that you use your updated connection string here. The original connection string copied from the portal won't have the username and password necessary to connect to the SQL database.

10. Within the **Program** class, enter the following line of code to create a new string constant named **cosmosDBConnectionString**:

```
 private const string cosmosDBConnectionString = "";
```

11. Update the cosmosDBConnectionString string constant by setting its value to the PRIMARY CONNECTION STRING of the Azure Cosmos DB account that you recorded earlier in this lab.
12. Within the **Program** class, enter the following code to create a new asynchronous **Main** method:

```
  public static async Task Main(string[] args)
        {

        }
```

13. Within the **Main** method, add the following line of code to print an introductory message to the console:

```
 await Console.Out.WriteLineAsync("Start Migration");
```

Save the **Program.cs** file.

.

14. At the open command prompt, enter the following command, and then select Enter to switch your terminal context to the **AdventureWorks.Migrate** folder:
15. At the command prompt, enter the following command, and then select Enter to build the .NET console application:

```
dotnet build
```

18. Select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

2.3 Task 3: Get SQL database records by using Entity Framework

1. Within the **Main** method of the **Program** class within the **Program.cs** file, add the following lines of code

```
public static async Task Main(string[] args)
        {
     await Console.Out.WriteLineAsync("Start Migration");
     using AdventureWorksSqlContext context = new AdventureWorksSqlContext(sqlDBConnectionString);
             List<Model> items = await context.Models.Include(m => m.Products)
                                                  .ToListAsync<Model>();
            await Console.Out.WriteLineAsync($"Total Azure SQL DB Records: {items.Count}");

        }
```

2. Save the **Program.cs** file and build the project

```
dotnet build
```

7. **Note**: If there are any build errors, review the **Program.cs** file in the **Allfiles (F):\Allfiles\Labs\04\Solution\AdventureWorks\AdventureWorks.Migrate** folder.
8. Select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

2.4 Task 4: Insert items into Azure Cosmos DB

1. Within the **Main** method of the Program class, add the following line of code to create a new **database** named **Retail** and insert the records from SQL Server

```
  using CosmosClient client = new CosmosClient(cosmosDBConnectionString);

                Database database = await client.CreateDatabaseIfNotExistsAsyn
c("Retail");
                Container container = await database.CreateContainerIfNotExist
sAsync("Online", partitionKeyPath: $"/{nameof(Model.Category)}");
                int count = 0;
                 foreach (var item in items)
                {
```

```
                          ItemResponse<Model> document = await container.Upser
tItemAsync<Model>(item);
                            await Console.Out.WriteLineAsync($"Upserted docume
nt #{++count:000} [Activity Id: {document.ActivityId}]");
                             await Console.Out.WriteLineAsync($"Total Azure Co
smos DB Documents: {count}");
                }
```

2. Save the **Program.cs** file and build the application
3. At the command prompt, enter the following command, and then select Enter to build the .NET console application:

```
dotnet build
```

12. **Note**: If there are any build errors, review the **Program.cs** file in the **Allfiles (F):\Allfiles\Labs\04\Solution\AdventureWorks\AdventureWorks.Migrate** folder.

2.5 Task 5: Perform a migration

1. At the open command prompt, enter the following command, and then select Enter to run the .NET console application:

```
dotnet run
```

Note: The **dotnet run** command will start the console application.

2. Observe the various data that prints to the screen, including initial SQL record count, individual upsert activity identifiers, and final Azure Cosmos DB document count.
3. Select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

2.6 Task 6: Validate the migration

1. Return to the **Microsoft Edge** browser window with the Azure portal.
2. In the Azure portal’s navigation pane, select the **Resource groups** link.
3. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
4. From the **PolyglotData** blade, select the **polycosmos*[yourname]*** Azure Cosmos DB account that you created earlier in this lab.
5. From the **Azure Cosmos DB account** blade, find and select the **Data Explorer** link from the blade.
6. In the Data Explorer pane, expand the **Retail** database node.
7. Expand the **Online** container node, and then select **New SQL Query**.

   Note: The label for this option might be hidden. You can get labels by hovering over the icons in the Data Explorer pane.

8. From the query tab, enter the following text:

```
SELECT * FROM models
```

Select **Execute Query**, and then observe the list of JSON models that the query returns.

Back in the query editor, replace the existing text with the following text:

```
SELECT VALUE COUNT(1) FROM models
```

10. Select **Execute Query**, and then observe the result of the **COUNT** aggregate operation.
11. Return to the **Visual Studio Code** window.

2.7 Review

In this exercise, you used Entity Framework and the .NET SDK for Azure Cosmos DB to migrate data from SQL Database to Azure Cosmos DB.