



Session 3

*Working with
ADO.NET and
Entity
Framework*

Session Overview

- Describe ADO.NET
- Explain Entity Framework
- Describe data handling in ASP.NET MVC with a code-first database

Overview of ADO.NET

1

- ADO.NET is used by developers to work with data in Web applications.

2

- It is a core component of .NET Framework

3

- It helps in establishing a connection between an application and data sources

Data Layer (1-2)

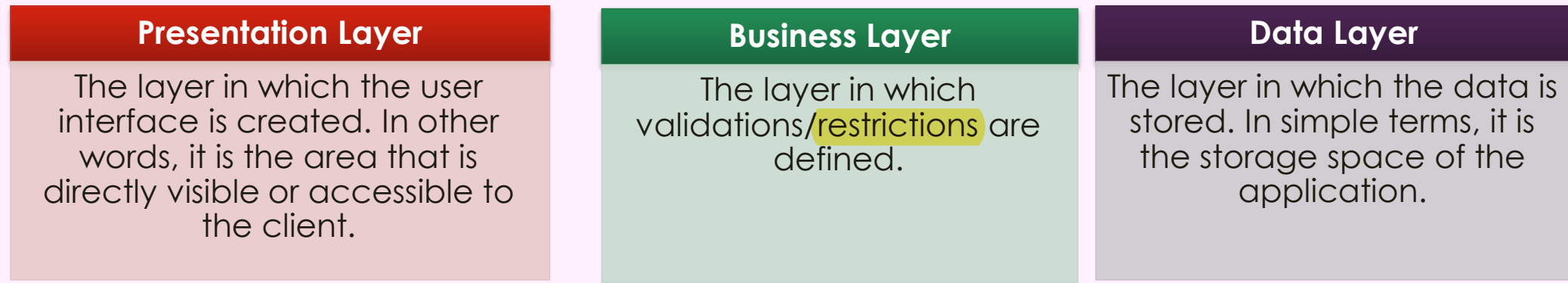
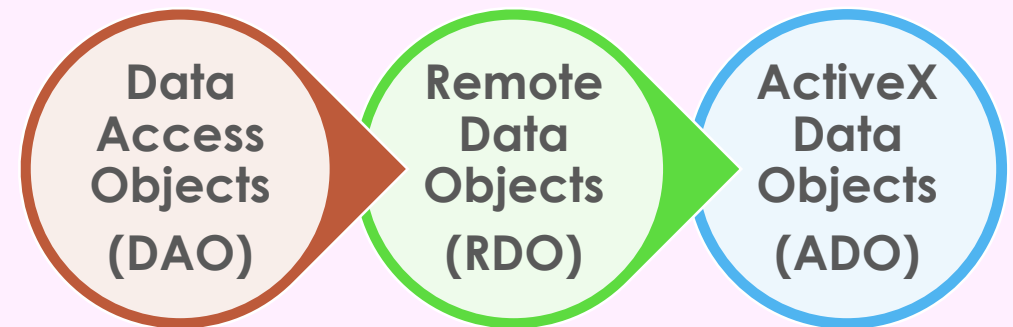


Figure 3.1: Microsoft Data Layers



Data Layer (2-2)

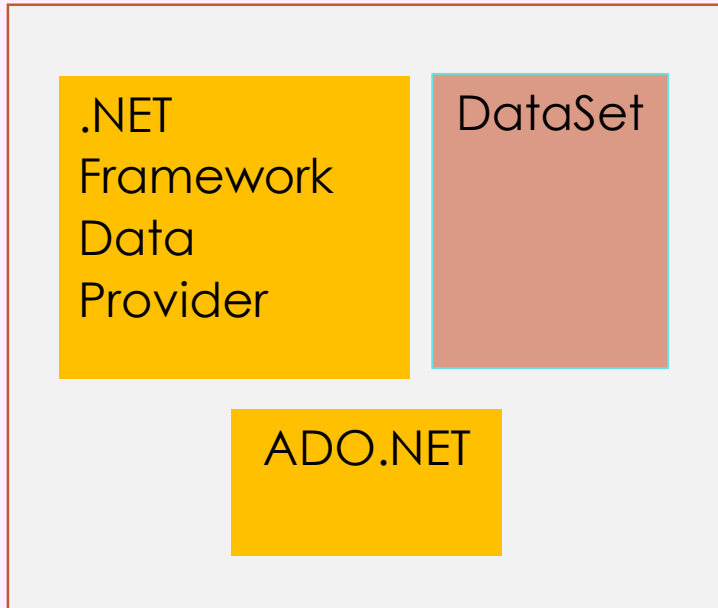


Figure 3.2: ADO.NET Components

Connection

Responsible for providing connectivity to a data source.

Command

Provides access to database commands.

DataAdapter

Acts as a bridge between the DataSet object and the data source.

DataReader

Provides a high-performance stream of data from the data source.

Entity Framework (1-2)

Entity Framework (EF) is an Object Relational Mapping (ORM) tool that is used to connect to the database.

In EF, Language Integrated Query (LINQ) is utilized to access the database and interact with auto-generated code.

EF establishes a bridge between the business entity and the data tables.

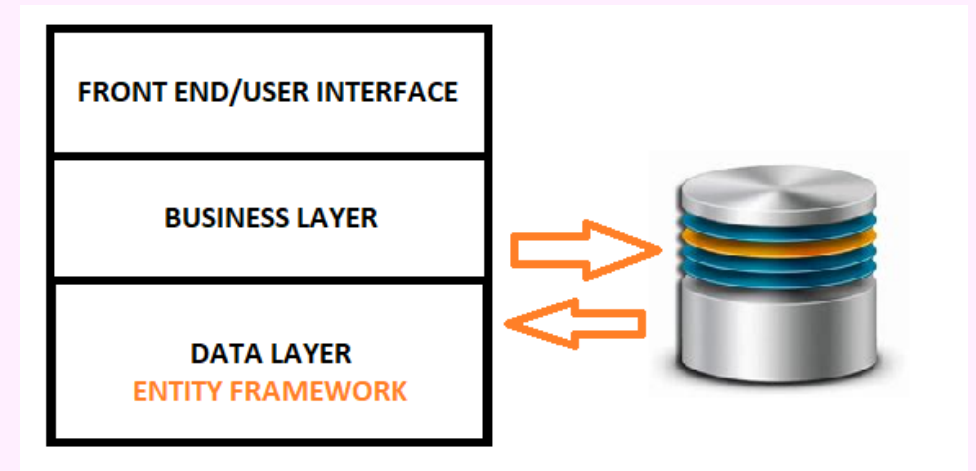



Figure 3.3: Entity Framework

Entity Framework (2-2)



EF uses LINQ queries instead of SQL queries and handles procedural and parameterized queries.

EF enables caching to allow queries to be answered from the cache in case of repeat queries.

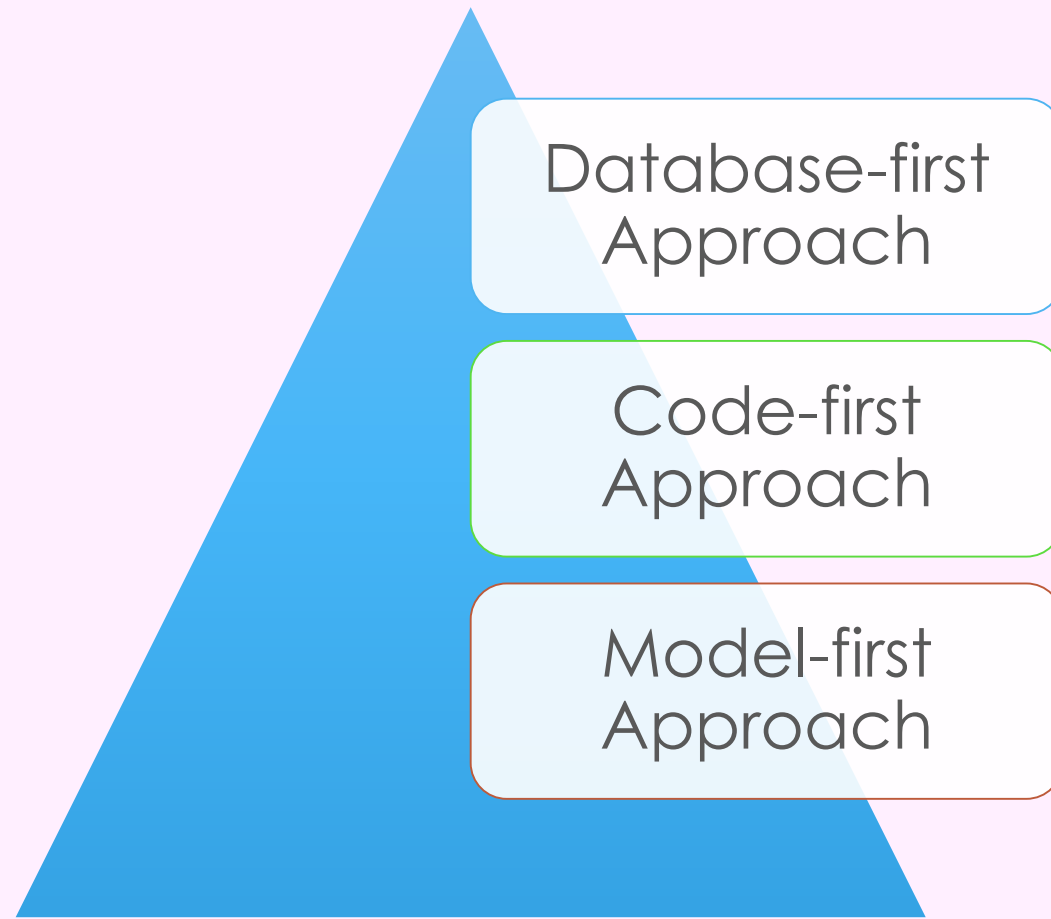
EF allows concurrency and ensures that any changes that are being overridden are retrieved by another user.

EF builds an Entity Data Model (EDM).

EF does automatic transaction management while requesting or saving data.

EF generates the required database commands for Create, Read, Update, and Delete (CRUD) operations and then, executes them.

Entity Framework Workflows



Data Handling in ASP.NET MVC with a Code-first Database(1-8)

Steps to create an application through a code-first approach in ASP.NET Core MVC:

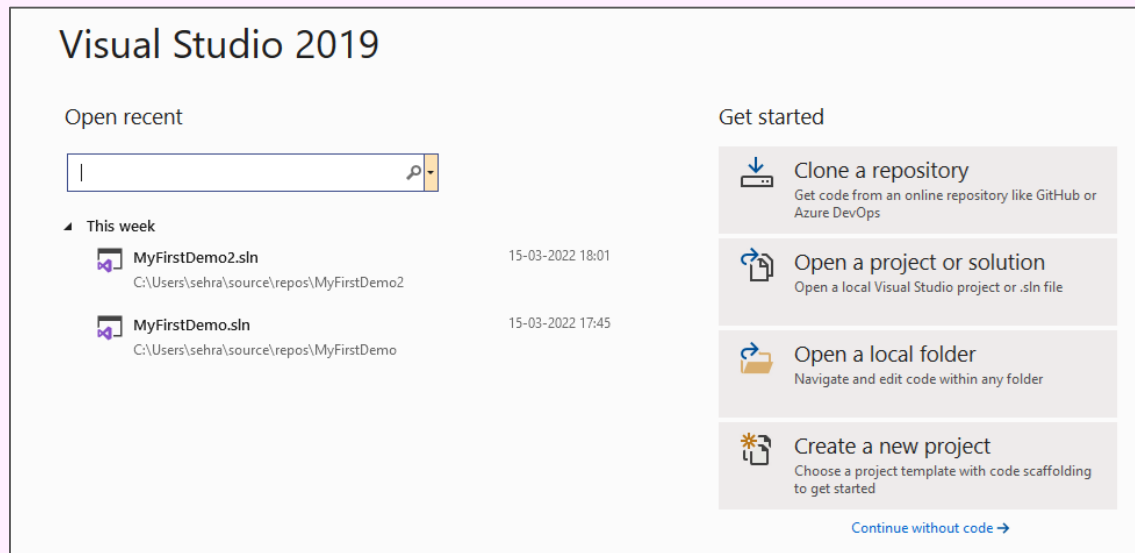


Figure 3.4: Create New Project Using Visual Studio 2019

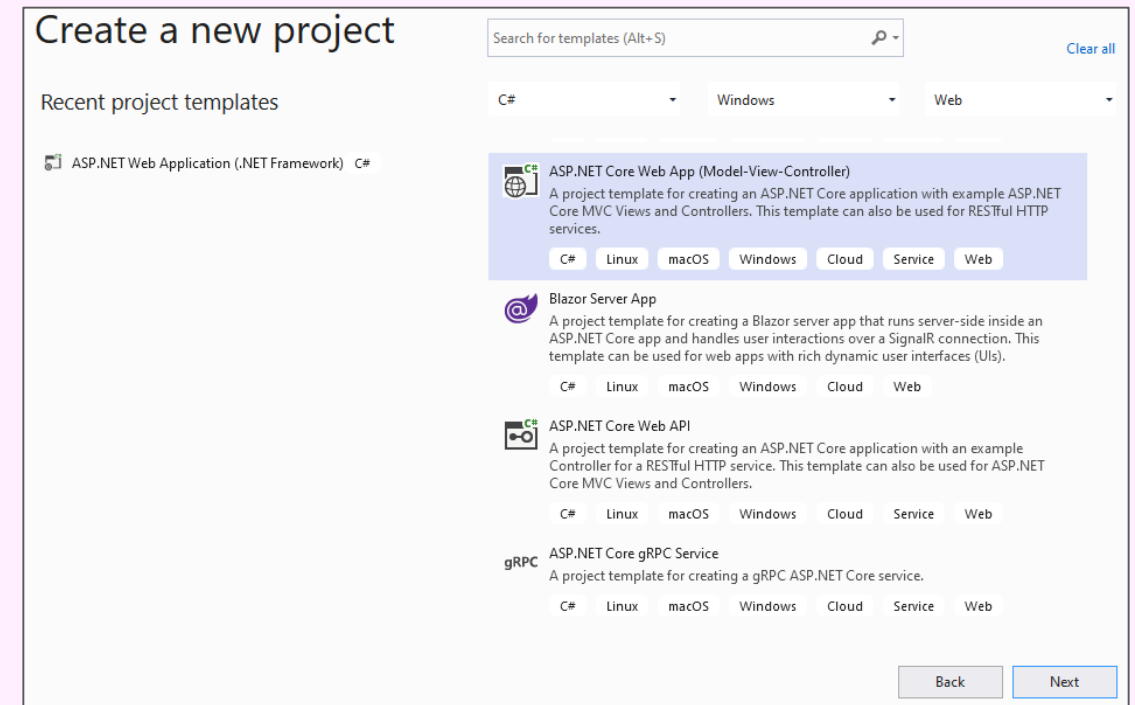


Figure 3.5: ASP .NET Core Web App

Data Handling in ASP.NET MVC with a Code-first Database(2-8)

The screenshot shows the 'Configure your new project' dialog box. At the top, it says 'Configure your new project'. Below that, there are tabs for 'ASP.NET Core Web App (Model-View-Controller)', 'C#', 'Linux', 'macOS', 'Windows', 'Cloud', 'Service', and 'Web'. The 'ASP.NET Core Web App (Model-View-Controller)' tab is selected. The 'Project name' field contains 'MyCodeFirstApproachDemo'. The 'Location' field contains 'C:\Users\username\source\repos'. The 'Solution name' field contains 'MyCodeFirstApproachDemo'. There is a checkbox labeled 'Place solution and project in the same directory' which is currently unchecked. At the bottom right, there are 'Back' and 'Next' buttons.

Figure 3.6: Configure Your New Project

The screenshot shows the 'Additional information' dialog box. At the top, it says 'Additional information'. Below that, there are tabs for 'ASP.NET Core Web App (Model-View-Controller)', 'C#', 'Linux', 'macOS', 'Windows', 'Cloud', 'Service', and 'Web'. The 'ASP.NET Core Web App (Model-View-Controller)' tab is selected. The 'Target Framework' dropdown is set to '.NET Core 3.1 (Long-term support)'. The 'Authentication Type' dropdown is set to 'None'. There is a checkbox labeled 'Configure for HTTPS' which is checked. There is a checkbox labeled 'Enable Docker' which is unchecked. The 'Docker OS' dropdown is set to 'Linux'. There is a checkbox labeled 'Enable Razor runtime compilation' which is unchecked. At the bottom right, there are 'Back' and 'Create' buttons.

Figure 3.7: Specifying Additional Information

Data Handling in ASP.NET MVC with a Code-first Database(3-8)

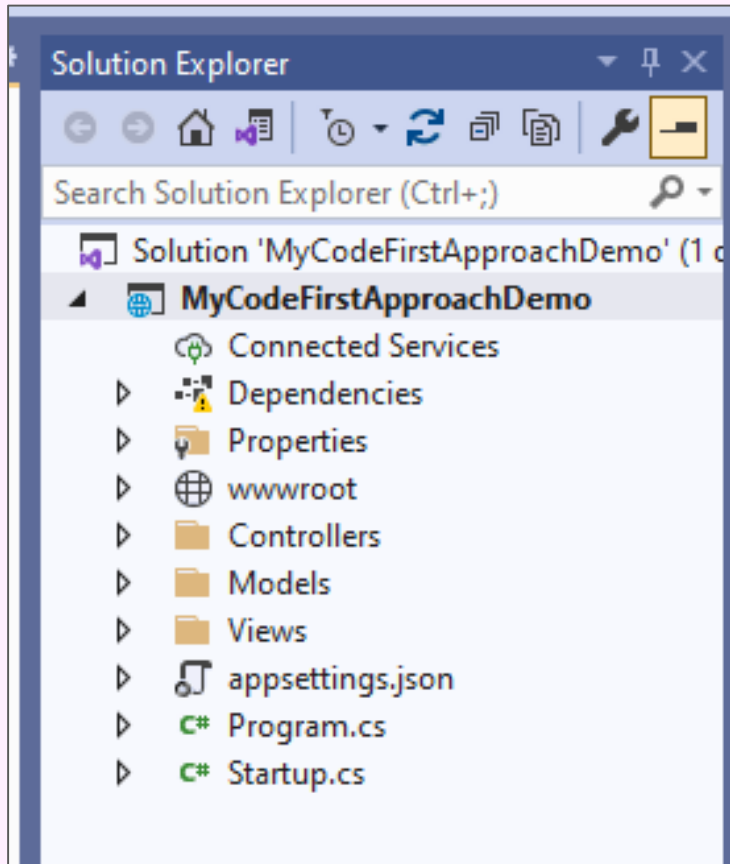


Figure 3.8: Solution Explorer Showing Dependencies Node

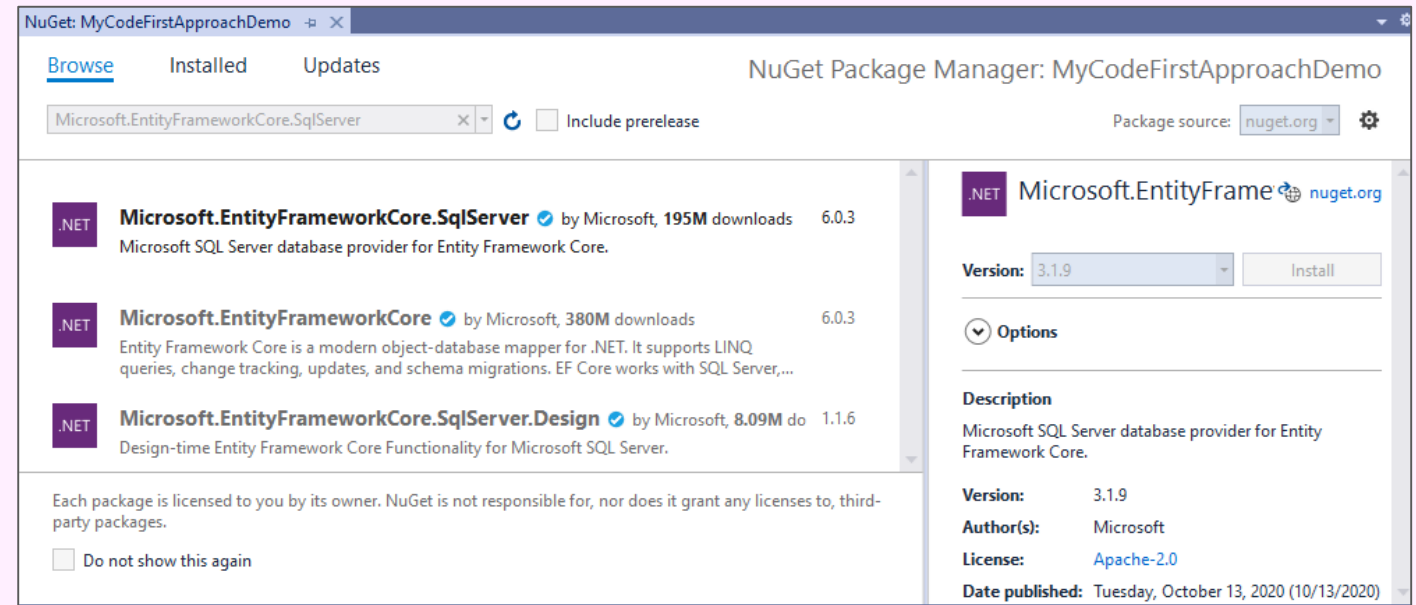


Figure 3.9: NuGet Package Manager

Data Handling in ASP.NET MVC with a Code-first Database(4-8)

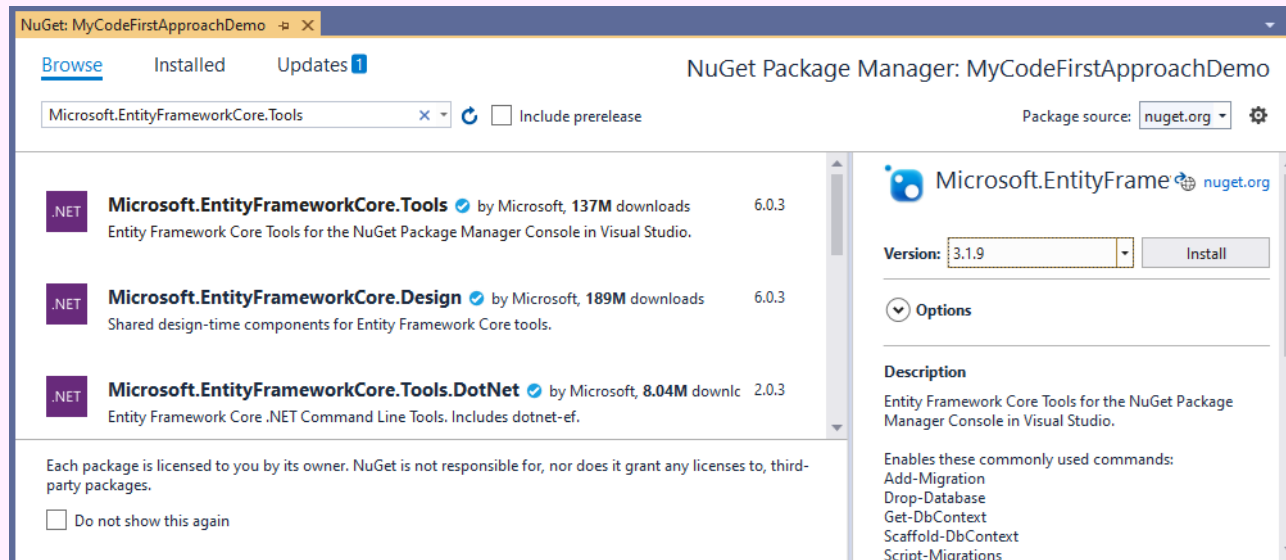


Figure 3.10: EF Tools

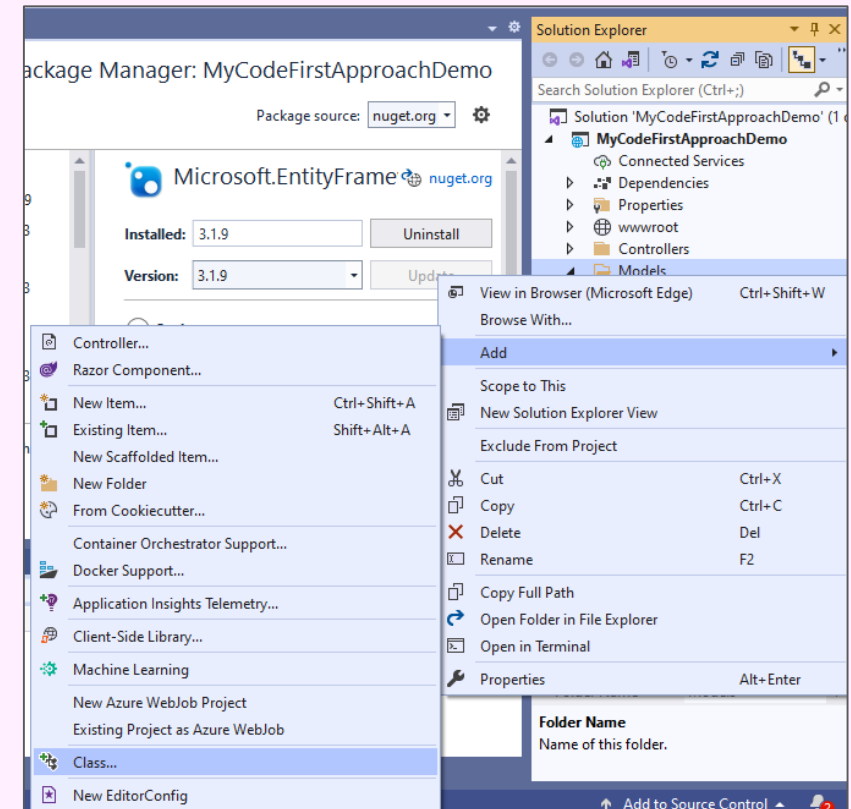


Figure 3.11: Add a Class

Data Handling in ASP.NET MVC with a Code-first Database(5-8)

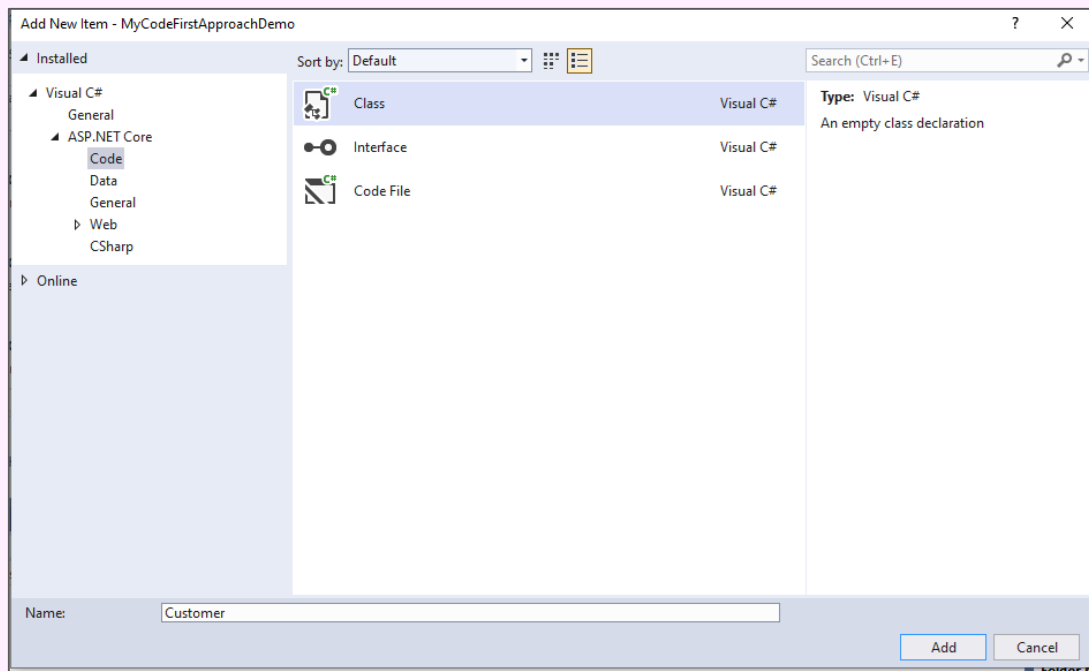


Figure 3.12: Change Default Name of Class

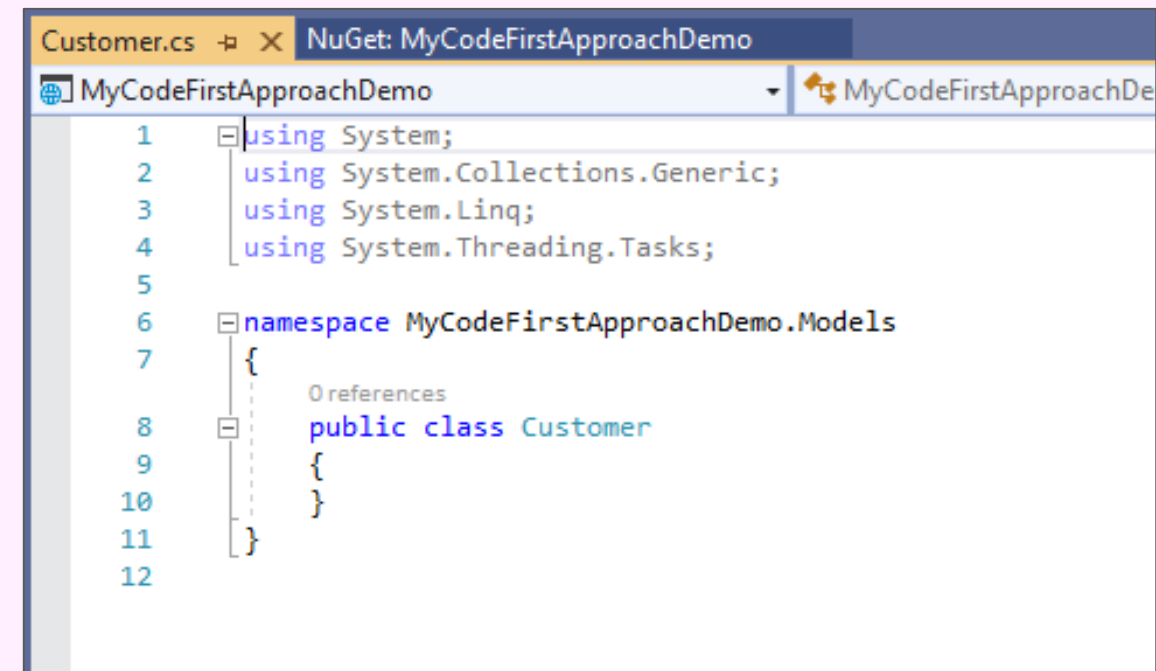


Figure 3.13: Default Auto-generated Class Template

Data Handling in ASP.NET MVC with a Code-first Database(6-8)

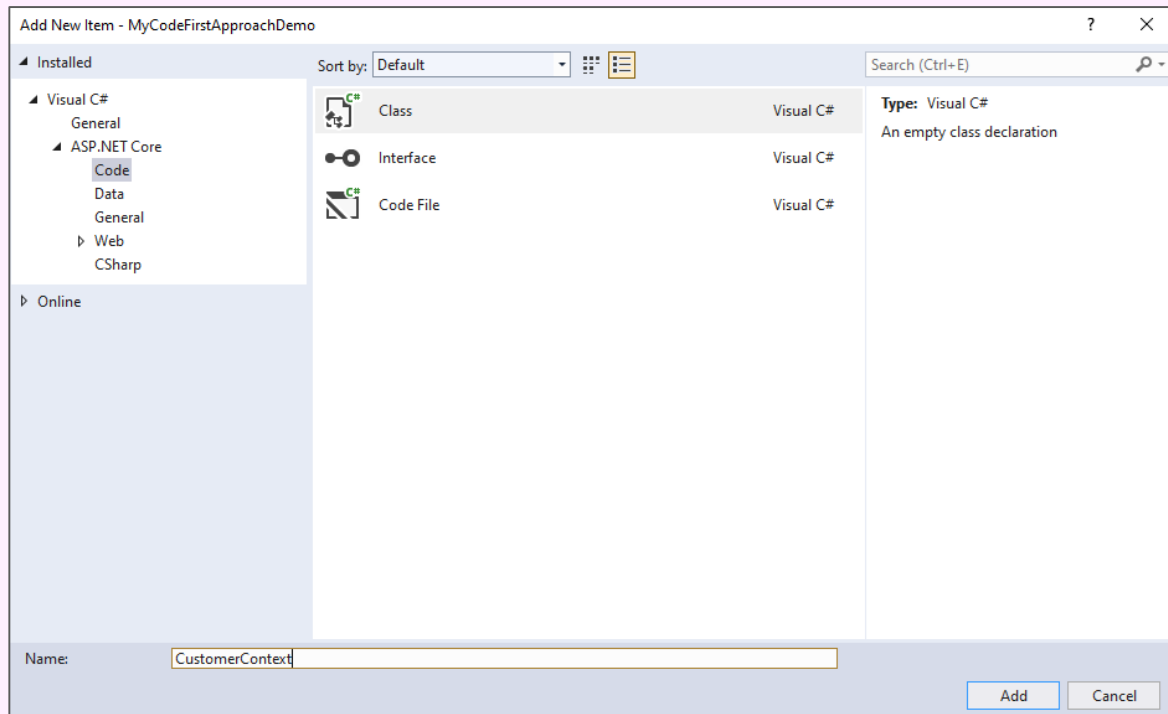


Figure 3.14: Adding Database Context

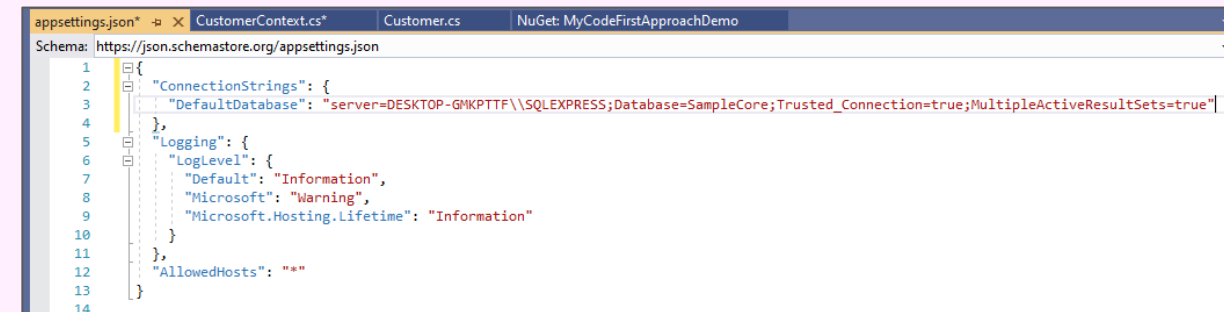


Figure 3.15: Configuration File

Data Handling in ASP.NET MVC with a Code-first Database(7-8)

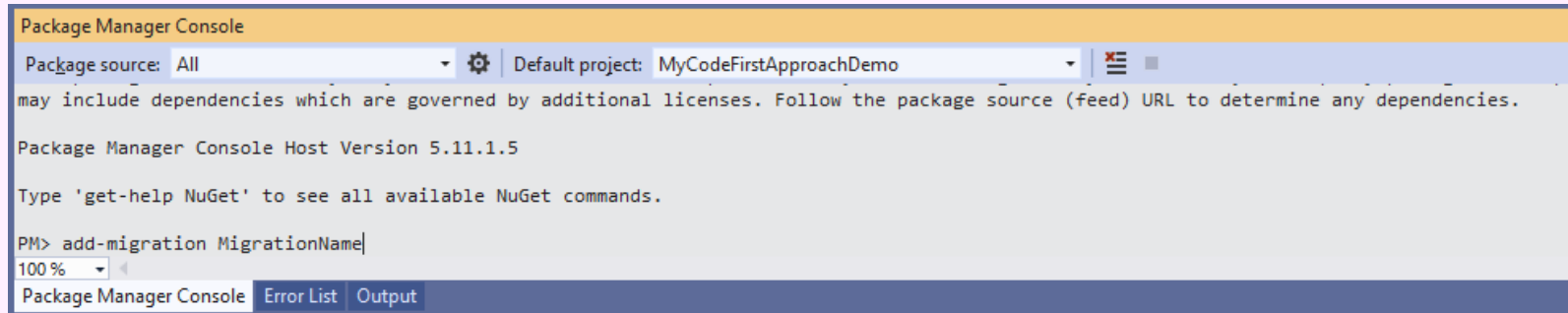


Figure 3.16: Migration

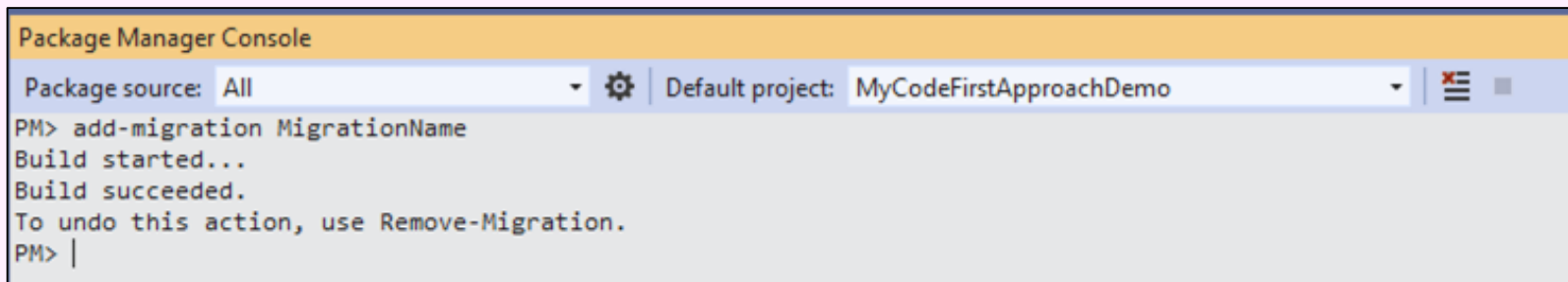


Figure 3.17: Build Message

Data Handling in ASP.NET MVC with a Code-first Database(8-8)

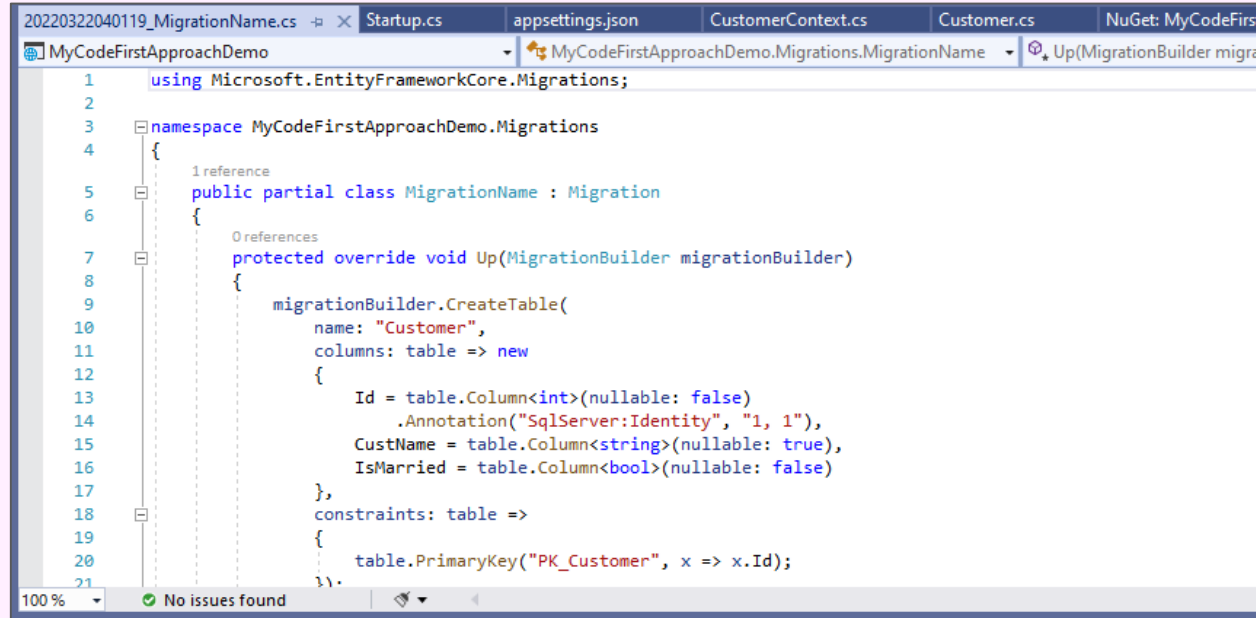


Figure 3.18: MigrationName Class

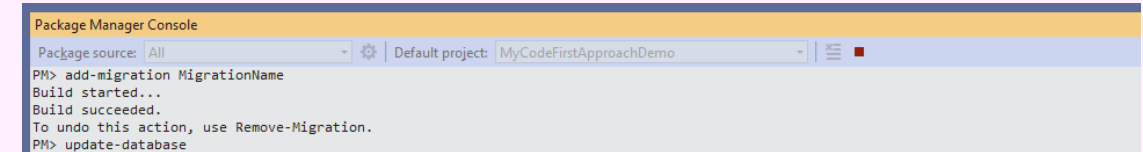


Figure 3.19: update-database Command

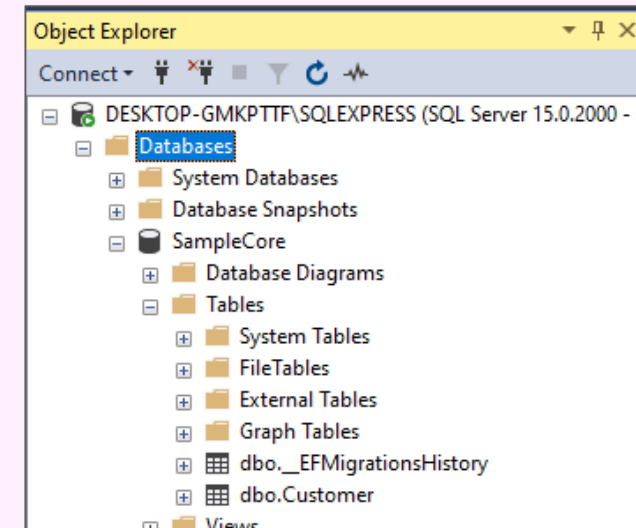


Figure 3.20: Database and Table Created in SQL Server 2019

Summary

- ADO.NET is a core component of the .NET framework and is used for establishing a connection between an application and data sources.
- Data layer is the storage space where data is stored.
- RDO is an object-oriented Data Access interface that combines the simple capabilities of DAO with the low power and flexibility of ODBC.
- ADO is a layer that allows application code to access any data that is stored in a generic manner without the requirement for database implementation.
- The Entity System is an Object-Relational Mapping (ORM)-based database management framework.
- EF is an improved version of ADO.NET that provides programmers with a completely automated database interface.
- EF uses LINQ queries instead of SQL queries.
- In the code-first approach, domain classes are created first.
- EF wizard generates the database tables based on the code.