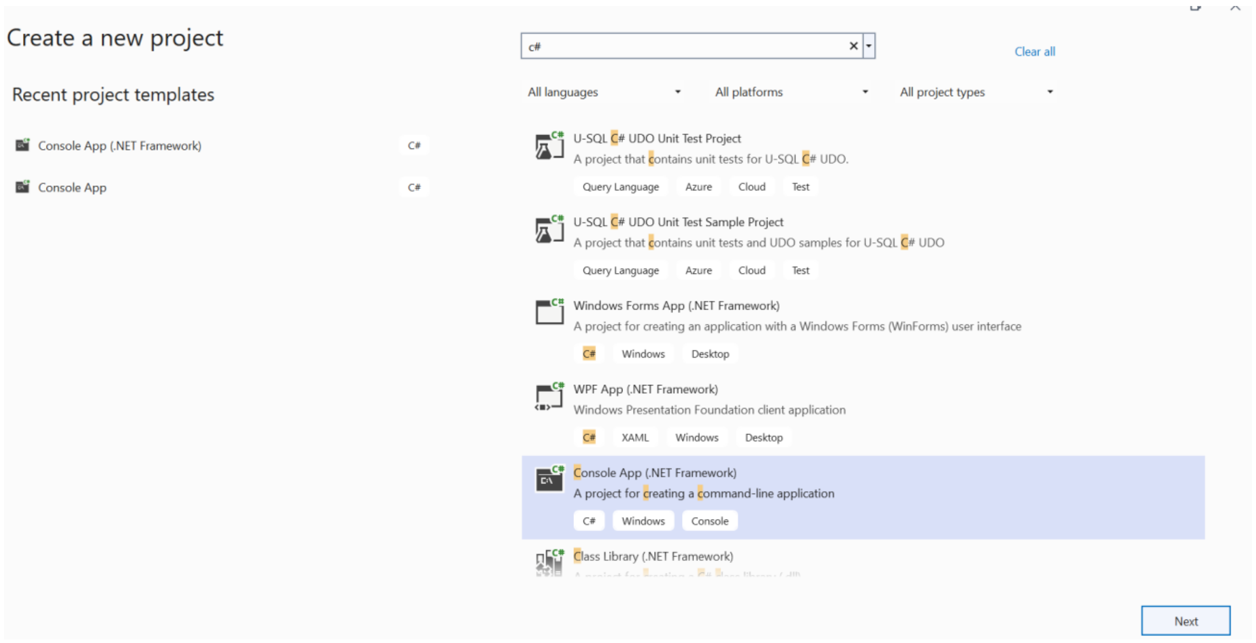
Lab Sheet 11

Entity Framework

Using Entity Framework, create posts for a blog which have Id, Time Stamp, Title and Body for the posts created. All the posts should reside in a Post table in a database.

1. Select a new console application project and name your application



Create a new project

Recent project templates

Console App (.NET Framework) C#

Console App C#

c#

All languages · All platforms · All project types · Clear all

U-SQL C# UDO Unit Test Project
A project that contains unit tests for U-SQL C# UDO.
Query Language · Azure · Cloud · Test

U-SQL C# UDO Unit Test Sample Project
A project that contains unit tests and UDO samples for U-SQL C# UDO
Query Language · Azure · Cloud · Test

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C# · Windows · Desktop

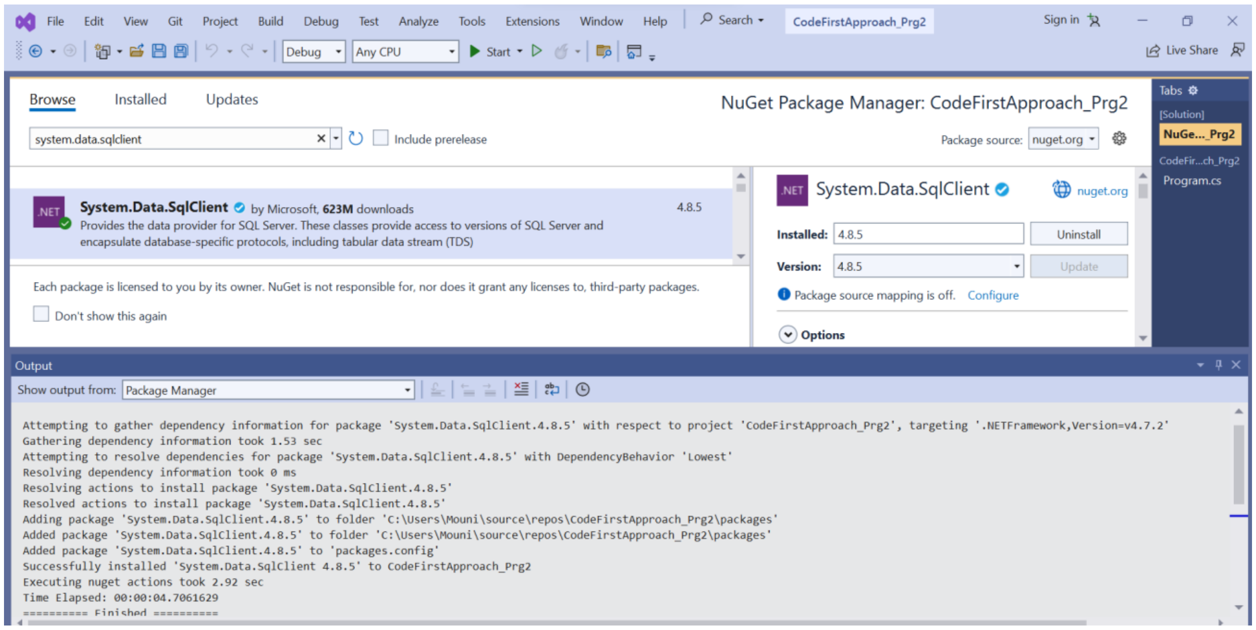
WPF App (.NET Framework)
Windows Presentation Foundation client application
C# · XAML · Windows · Desktop

Console App (.NET Framework)
A project for creating a command-line application
C# · Windows · Console

Class Library (.NET Framework)

Next

2. Install System.Data.SqlClient package



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search CodeFirstApproach_Prg2 Sign in
Debug · Any CPU · Start · Live Share
Tabs · [Solution] · NuGe..._Prg2 · CodeFir...ch_Prg2 · Program.cs

Browse · Installed · Updates

NuGet Package Manager: CodeFirstApproach_Prg2

system.data.sqlclient · Include prerelease · Package source: nuget.org

System.Data.SqlClient by Microsoft, 623M downloads — 4.8.5
Provides the data provider for SQL Server. These classes provide access to versions of SQL Server and encapsulate database-specific protocols, including tabular data stream (TDS)

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Don't show this again

System.Data.SqlClient · nuget.org

Installed: 4.8.5 · Uninstall

Version: 4.8.5 · Update

Package source mapping is off. Configure

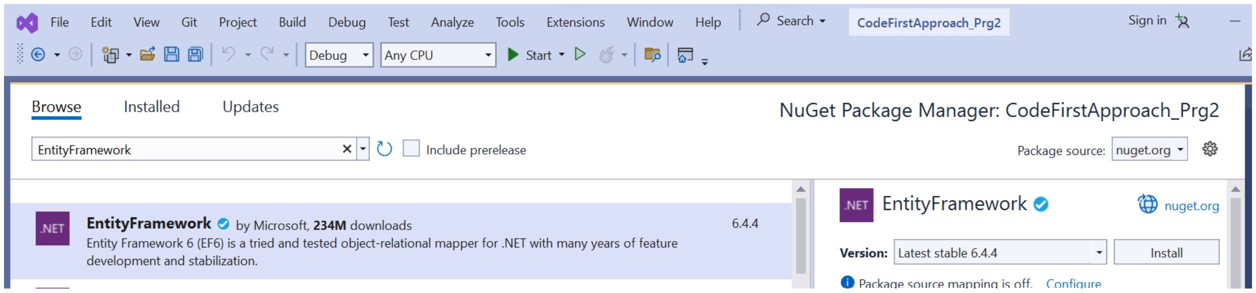
Options

Output

Show output from: Package Manager

```
Attempting to gather dependency information for package 'System.Data.SqlClient.4.8.5' with respect to project 'CodeFirstApproach_Prg2', targeting '.NETFramework,Version=v4.7.2'
Gathering dependency information took 1.53 sec
Attempting to resolve dependencies for package 'System.Data.SqlClient.4.8.5' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
Resolving actions to install package 'System.Data.SqlClient.4.8.5'
Resolved actions to install package 'System.Data.SqlClient.4.8.5'
Adding package 'System.Data.SqlClient.4.8.5' to folder 'C:\Users\Mouni\source\repos\CodeFirstApproach_Prg2\packages'
Added package 'System.Data.SqlClient.4.8.5' to folder 'C:\Users\Mouni\source\repos\CodeFirstApproach_Prg2\packages'
Added package 'System.Data.SqlClient.4.8.5' to 'packages.config'
Successfully installed 'System.Data.SqlClient 4.8.5' to CodeFirstApproach_Prg2
Executing nuget actions took 2.92 sec
Time Elapsed: 00:00:04.7061629
========== Finished ==========
```

3. Install EntityFramework package



NuGet Package Manager: CodeFirstApproach_Prg2

Browse | Installed | Updates

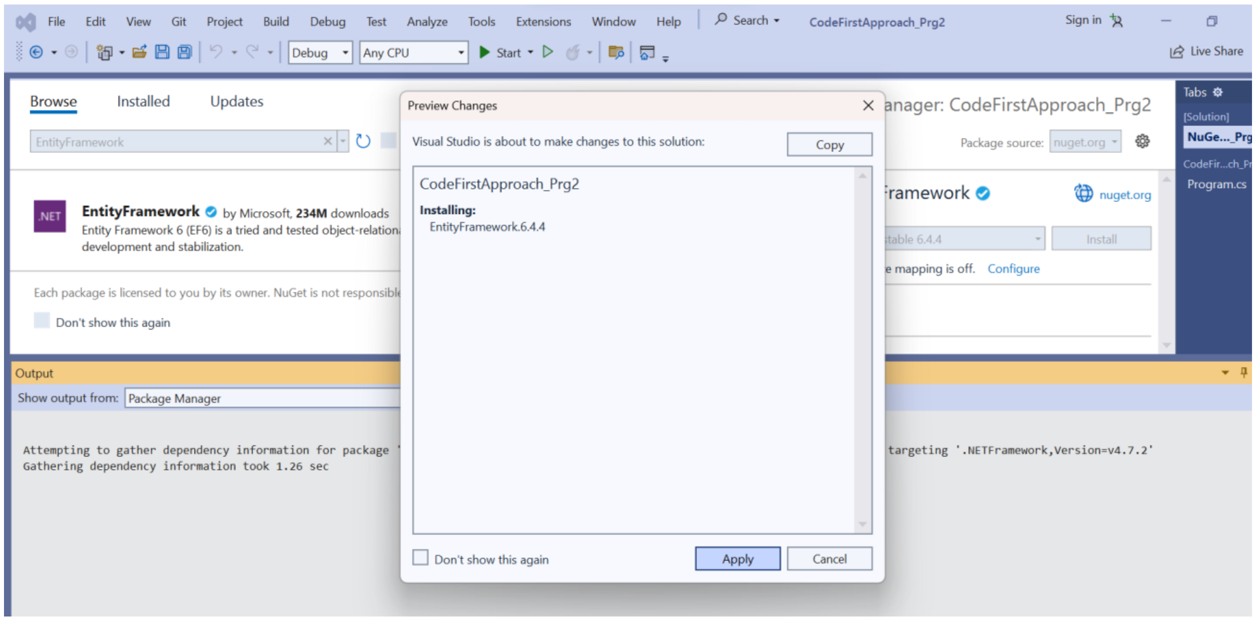
EntityFramework ☐ Include prerelease

Package source: nuget.org

EntityFramework by Microsoft, 234M downloads 6.4.4
Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature development and stabilization.

Version: Latest stable 6.4.4 Install

Package source mapping is off. Configure



Preview Changes

Visual Studio is about to make changes to this solution:

CodeFirstApproach_Prg2
Installing:
EntityFramework.6.4.4

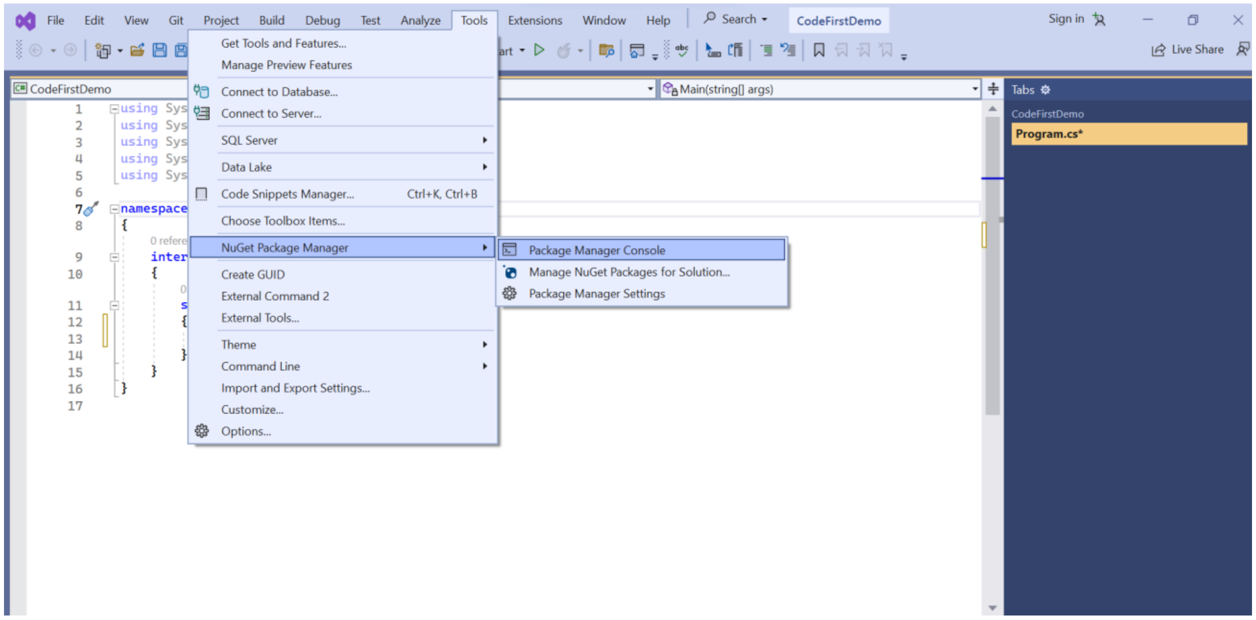
☐ Don't show this again Apply Cancel

Show output from: Package Manager

```
Attempting to gather dependency information for package '            targeting '.NETFramework,Version=v4.7.2'
Gathering dependency information took 1.26 sec
```

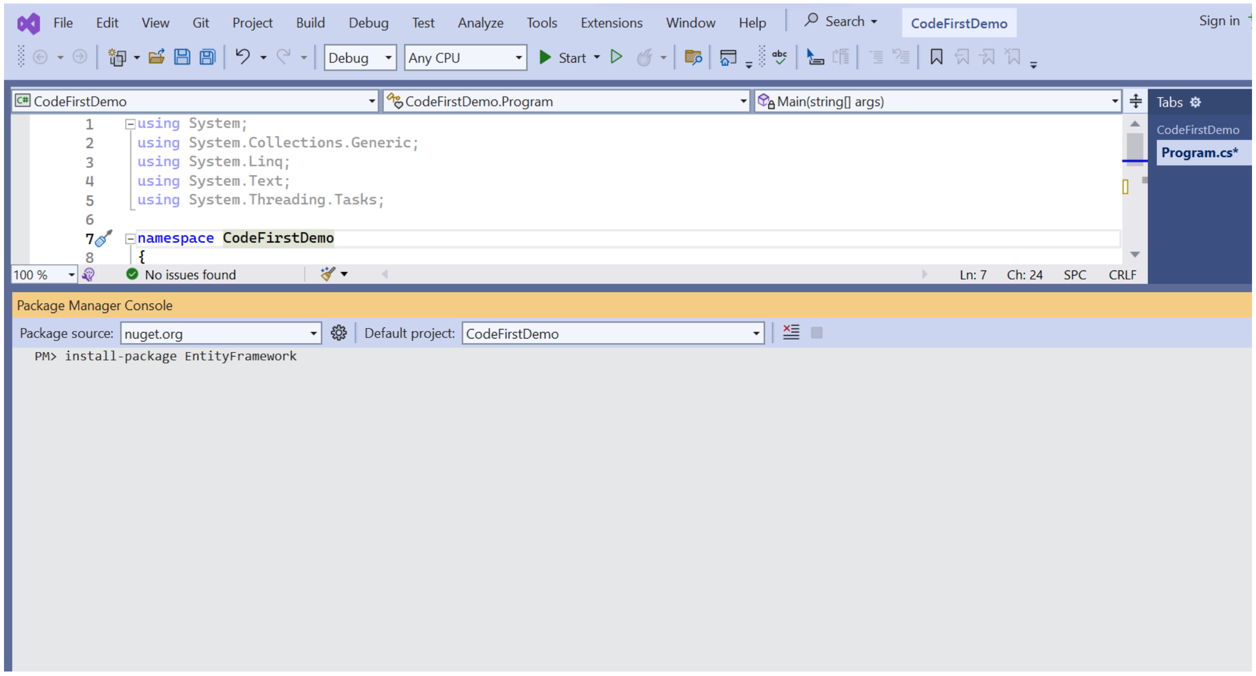
Or install this package from console

From Tools, select Package Manager Console



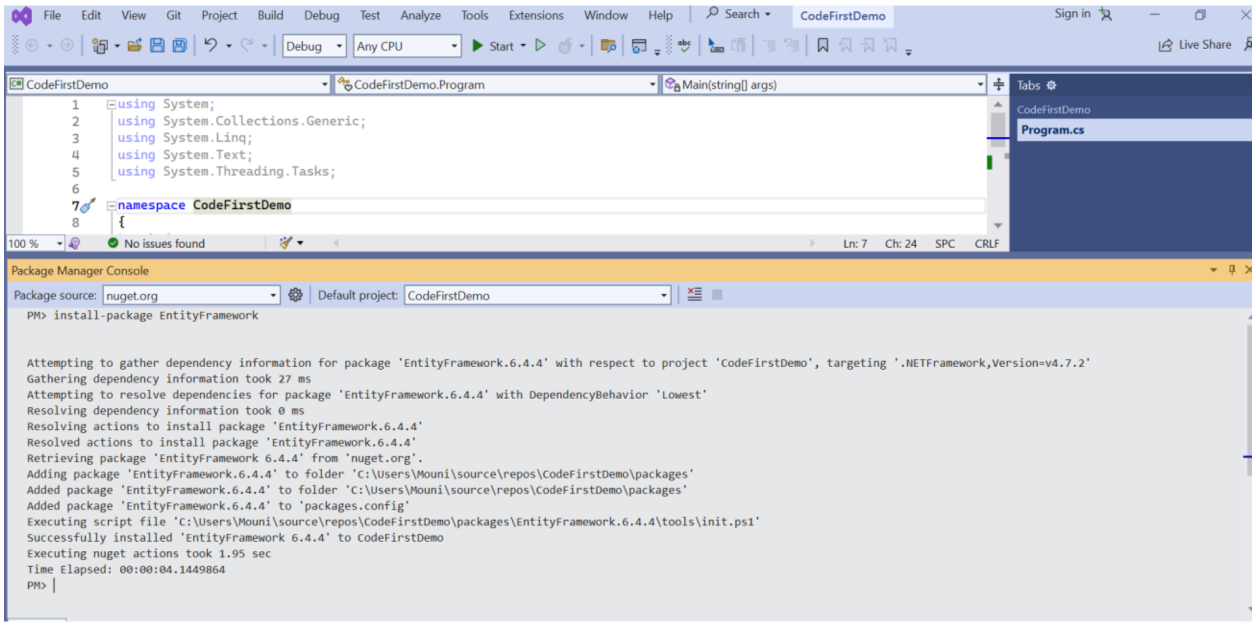
From the console, install EntityFramework.

- **install-package EntityFramework**



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

namespace CodeFirstDemo
{
```

```
PM> install-package EntityFramework
```



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

namespace CodeFirstDemo
{
```

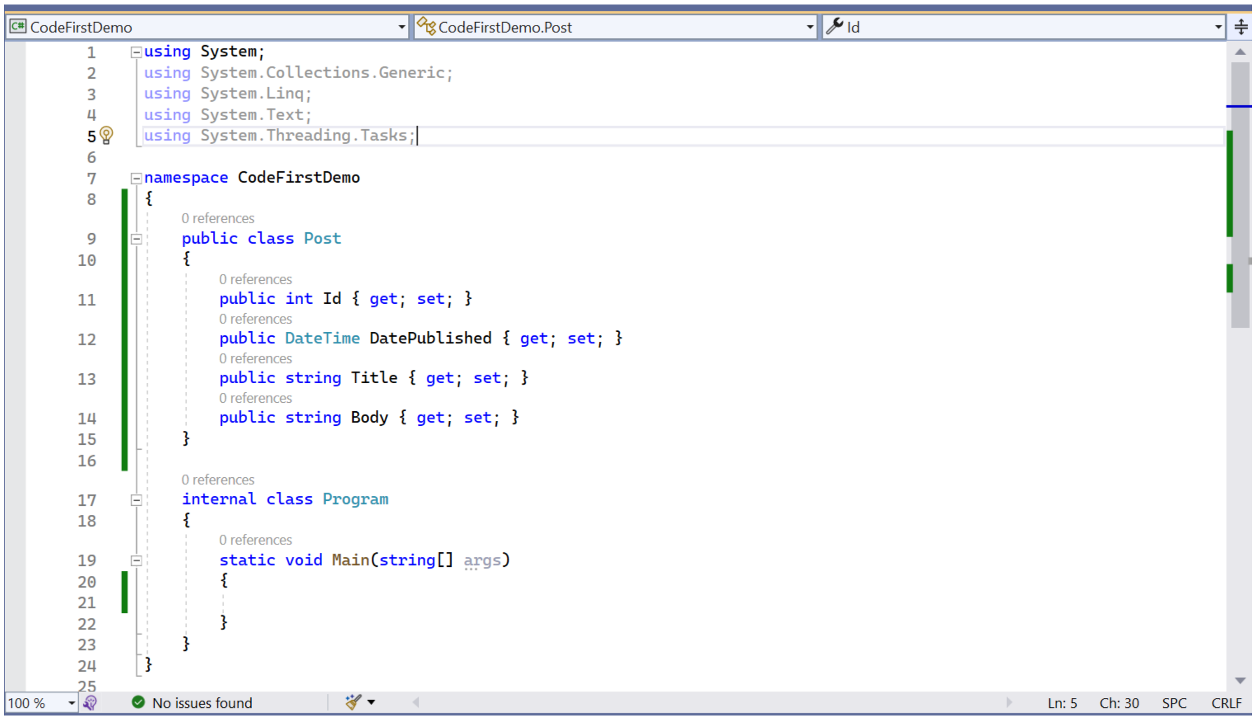
```
PM> install-package EntityFramework


Attempting to gather dependency information for package 'EntityFramework.6.4.4' with respect to project 'CodeFirstDemo', targeting '.NETFramework,Version=v4.7.2'
Gathering dependency information took 27 ms
Attempting to resolve dependencies for package 'EntityFramework.6.4.4' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
Resolving actions to install package 'EntityFramework.6.4.4'
Resolved actions to install package 'EntityFramework.6.4.4'
Retrieving package 'EntityFramework 6.4.4' from 'nuget.org'.
Adding package 'EntityFramework.6.4.4' to folder 'C:\Users\Mouni\source\repos\CodeFirstDemo\packages'
Added package 'EntityFramework.6.4.4' to folder 'C:\Users\Mouni\source\repos\CodeFirstDemo\packages'
Added package 'EntityFramework.6.4.4' to 'packages.config'
Executing script file 'C:\Users\Mouni\source\repos\CodeFirstDemo\packages\EntityFramework.6.4.4\tools\init.ps1'
Successfully installed 'EntityFramework 6.4.4' to CodeFirstDemo
Executing nuget actions took 1.95 sec
Time Elapsed: 00:00:04.1449864
PM> |
```

4.

Create a Post class with following code:

```
public class Post {

    public int Id { get; set; }

    public DateTime DatePublished { get; set; }

    public string Title { get; set; }
```

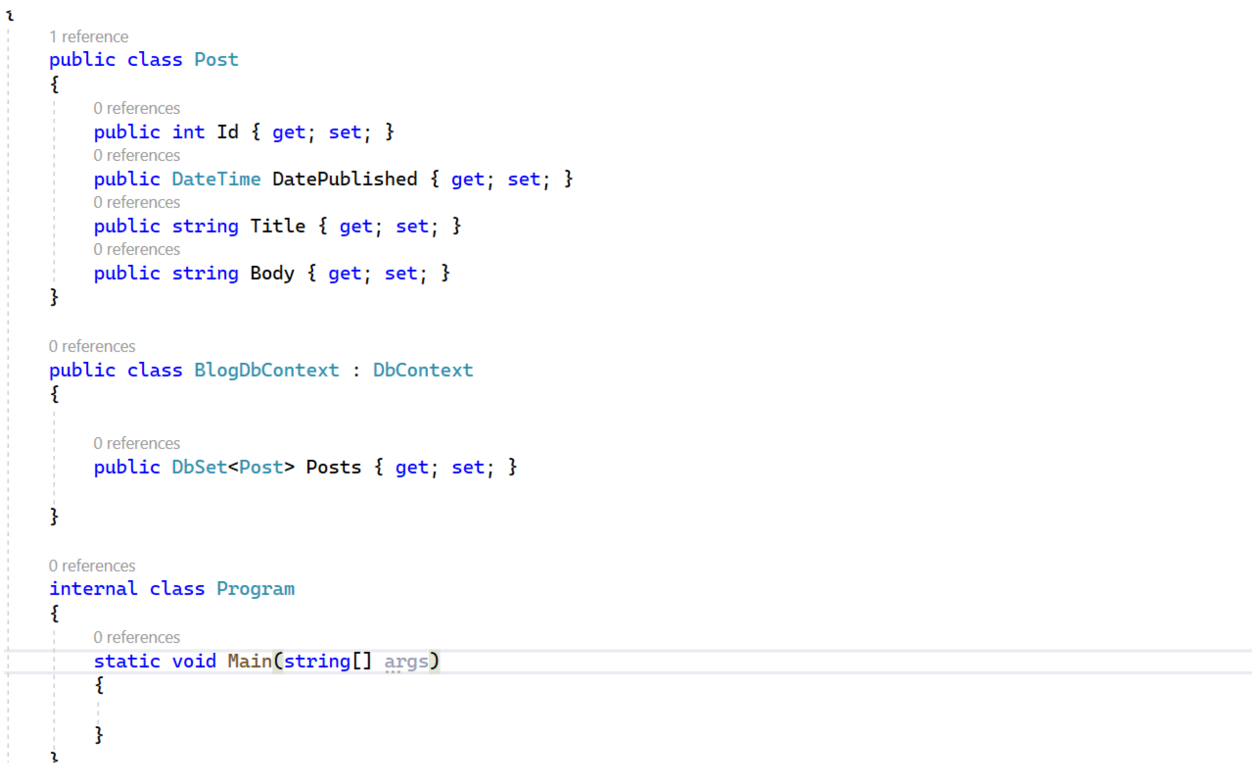
```
        public string Body { get; set; }

    }
```



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

namespace CodeFirstDemo
{
    0 references
    public class Post
    {
        0 references
        public int Id { get; set; }
        0 references
        public DateTime DatePublished { get; set; }
        0 references
        public string Title { get; set; }
        0 references
        public string Body { get; set; }
    }

    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {

        }
    }
}
```

Create a BlogDbContext class with the following code:

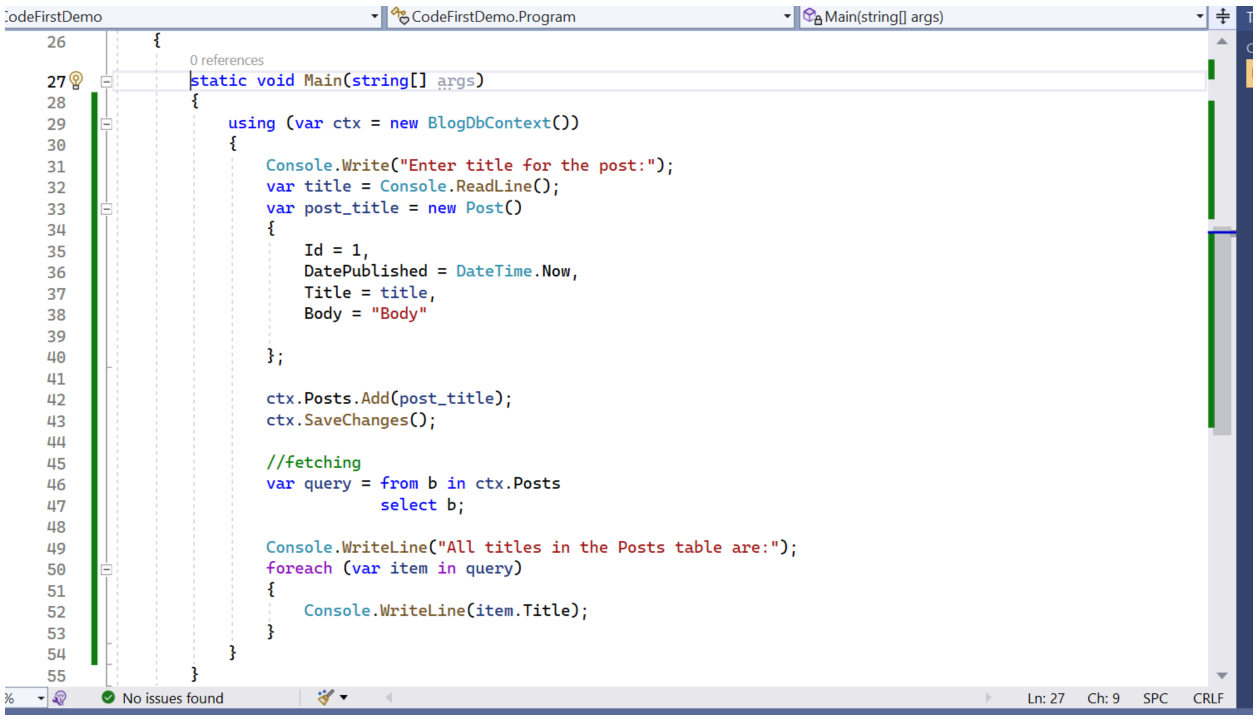
```
  public class BlogDbContext: DbContext
 {
     public DbSet<Post> Posts { get; set; }
  }
```


```
1 reference
public class Post
{
    0 references
    public int Id { get; set; }
    0 references
    public DateTime DatePublished { get; set; }
    0 references
    public string Title { get; set; }
    0 references
    public string Body { get; set; }
}

0 references
public class BlogDbContext : DbContext
{

    0 references
    public DbSet<Post> Posts { get; set; }

}

0 references
internal class Program
{
    0 references
    static void Main(string[] args)
    {

    }
}
```

In the main method, write following code for adding and reading values of the posts

```
internal class Program
   {
      static void Main(string[] args)
      {
         using (var ctx = new BlogDbContext())
         {
            Console.Write("Enter title for the post:");
            var title = Console.ReadLine();
            var post_title = new Post()
                        {
                                    Id = 1,
                                    DatePublished = DateTime.Now,
                                    Title = title,
                                    Body = "Body"
```

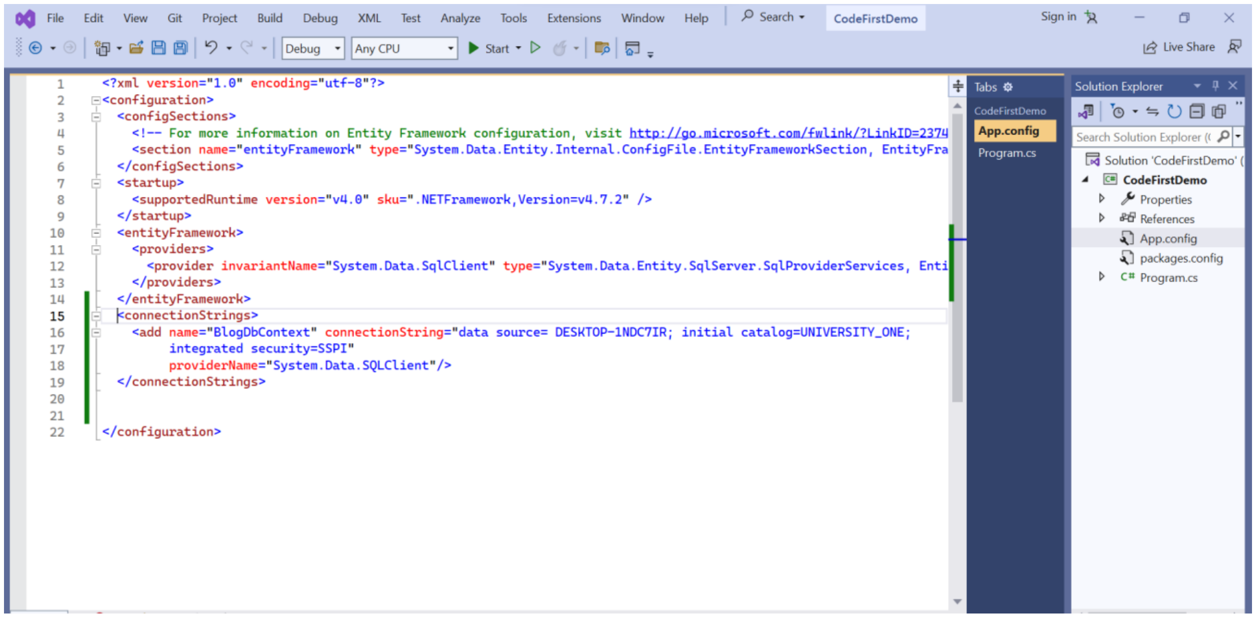
```
                };

            ctx.Posts.Add(post_title);
            ctx.SaveChanges();

            //fetching
            var query = from b in ctx.Posts
                        select b;

            Console.WriteLine("All titles in the Posts table are:");
            foreach (var item in query)
            {
                Console.WriteLine(item.Title);
            }
        }
    }
}
```



```
    {
        static void Main(string[] args)
        {
            using (var ctx = new BlogDbContext())
            {
                Console.Write("Enter title for the post:");
                var title = Console.ReadLine();
                var post_title = new Post()
                {
                    Id = 1,
                    DatePublished = DateTime.Now,
                    Title = title,
                    Body = "Body"

                };

                ctx.Posts.Add(post_title);
                ctx.SaveChanges();

                //fetching
                var query = from b in ctx.Posts
                            select b;

                Console.WriteLine("All titles in the Posts table are:");
                foreach (var item in query)
                {
                    Console.WriteLine(item.Title);
                }
            }
        }
```

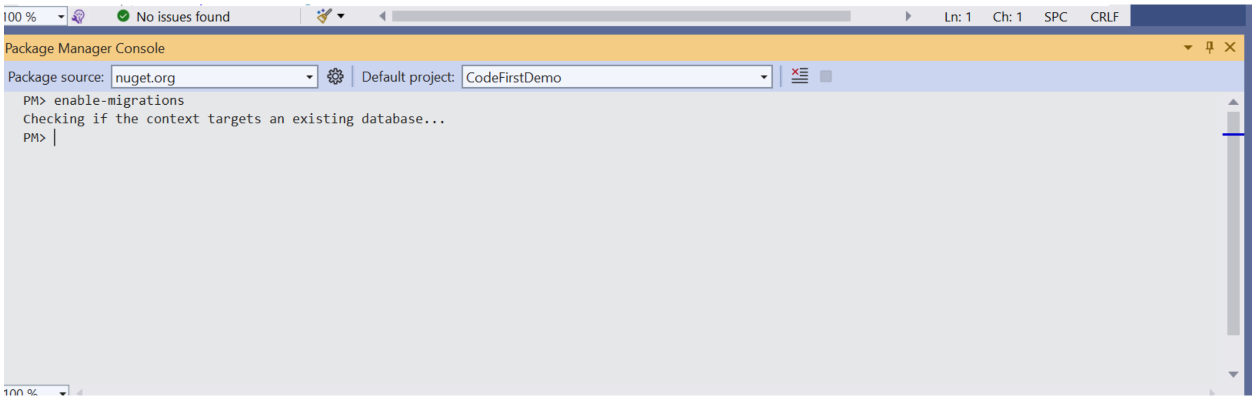
5. In the AppConfig file, enter the connection string, to establish the connection to required database (for the data source, give the server name of your computer). Data source and initial catalog are your server name and database respectively, so change them as per your system config and the db name that you created.

```
  <connectionStrings>

      <add name="BlogDbContext" connectionString="data source= DESKTOP-
      1NDC7IR; initial catalog=UNIVERSITY_ONE;integrated security=SSPI"
      providerName="System.Data.SqlClient"/>

</connectionStrings>
```



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=2374
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFra
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <entityFramework>
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, Enti
    </providers>
  </entityFramework>
  <connectionStrings>
    <add name="BlogDbContext" connectionString="data source= DESKTOP-1NDC7IR; initial catalog=UNIVERSITY_ONE;
         integrated security=SSPI"
         providerName="System.Data.SQLClient"/>
  </connectionStrings>


</configuration>
```

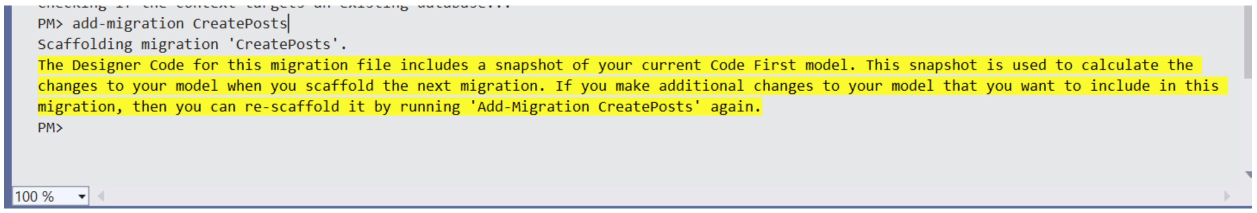
6. Enable code first migrations, so add migrations through Package Manager console
   - ➢ enable-migrations



```
PM> enable-migrations
Checking if the context targets an existing database...
PM> |
```

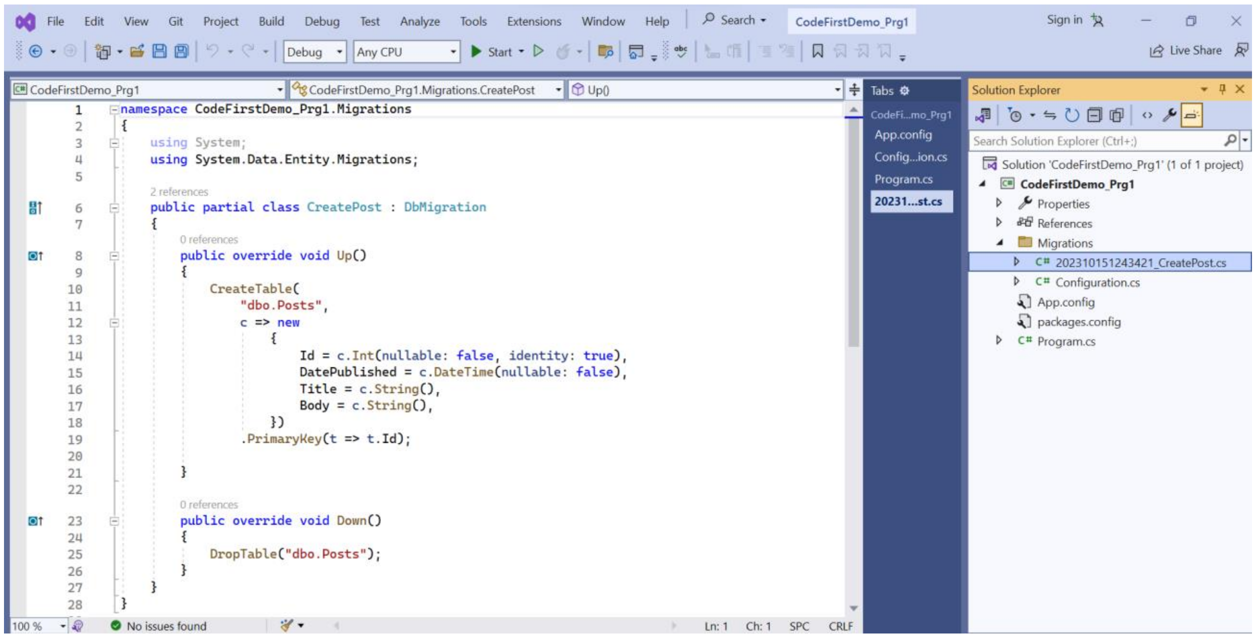
7. For any change in the model of our application, we create migrations. The name for the migrations can be the type of change we have made. (Here we are creating a Post class)

   - ➢ add-migration CreatePost



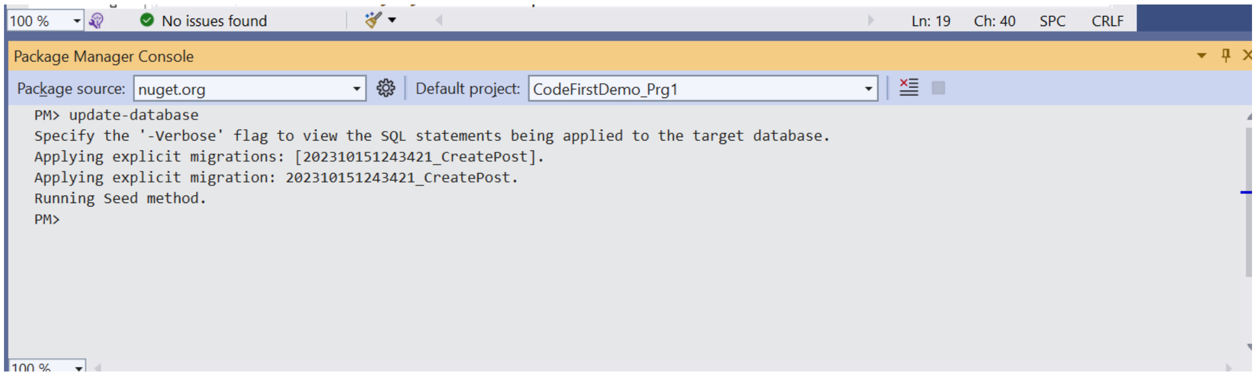
```
PM> add-migration CreatePosts
Scaffolding migration 'CreatePosts'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate the
changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to include in this
migration, then you can re-scaffold it by running 'Add-Migration CreatePosts' again.
PM>
```

A new file will be created in the Migrations folder



```
namespace CodeFirstDemo_Prg1.Migrations
{
    using System;
    using System.Data.Entity.Migrations;

    public partial class CreatePost : DbMigration
    {
        public override void Up()
        {
            CreateTable(
                "dbo.Posts",
                c => new
                    {
                        Id = c.Int(nullable: false, identity: true),
                        DatePublished = c.DateTime(nullable: false),
                        Title = c.String(),
                        Body = c.String(),
                    })
                .PrimaryKey(t => t.Id);

        }

        public override void Down()
        {
            DropTable("dbo.Posts");
        }
    }
}
```

8. Run the migrations, to do this, run the command
   - update-database

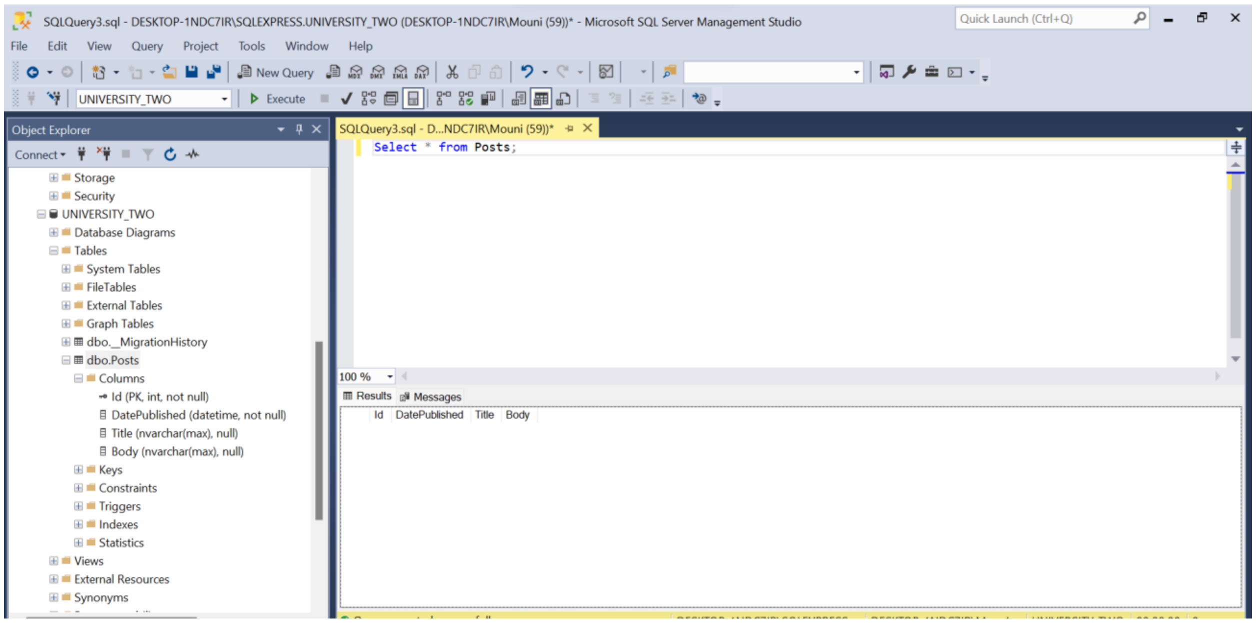


Package source: nuget.org
Default project: CodeFirstDemo_Prg1

```
PM> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [202310151243421_CreatePost].
Applying explicit migration: 202310151243421_CreatePost.
Running Seed method.
PM>
```

(If any network error is seen at this step, enable the tcp-ip connection and port in the sql-server configuration).

9. On the SSMS, check the table and the database that you have updated.



```
Select * from Posts;
```

```
Id  DatePublished  Title  Body
```

ASSIGNMENT 11

Using Entity Framework, create a DB table with Student details of Student ID and Student Name. All the Student details should reside in StudentEF table.