



How To Connect MySQL With ASP.NET Core

ASK A QUESTION

CONTRIBUTE



Abhijit Patil

Apr 28 2017

Article

5

3

118.6k

[Download Free Office API](#)

Objective

In this article, we are going to understand how to connect .NET CORE Application with MySQL and read data from MySQL, using .NET Core MySQL connector.

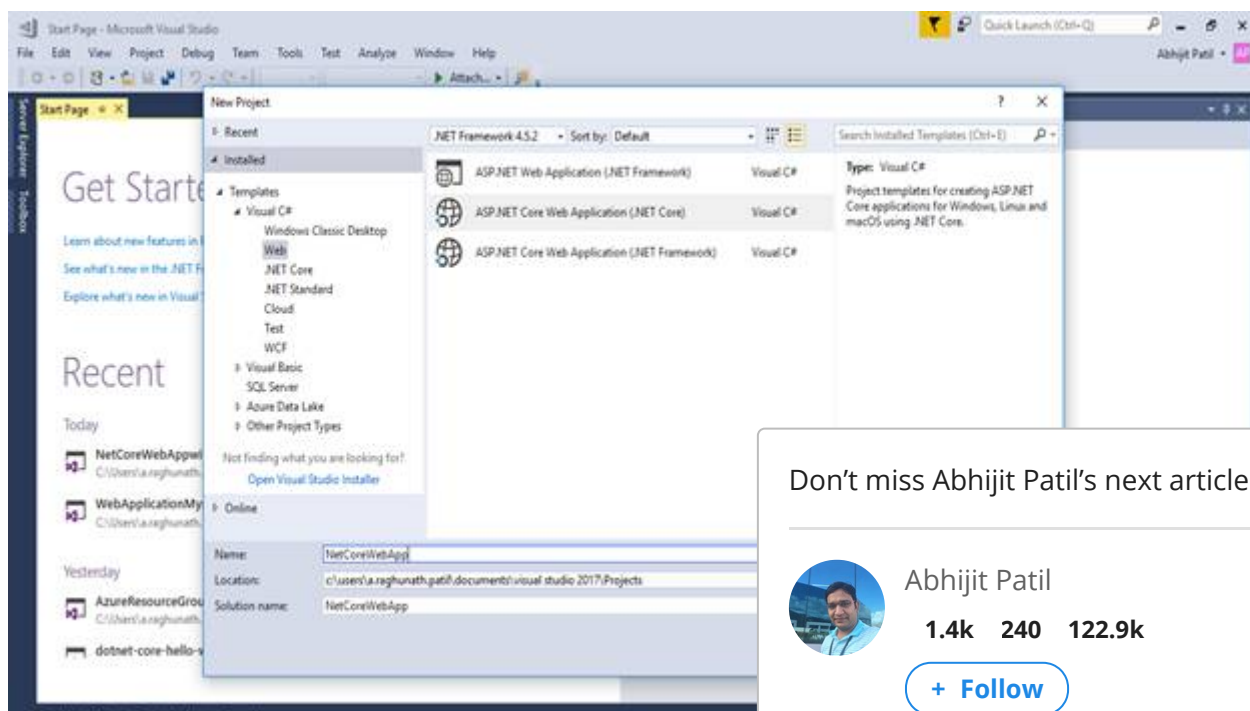
Prerequisites

1. MySQL (You can get MySQL IDE – MySQL Work Bench [here](#)).
2. NET CORE Environment Setup. (you can get it [here](#)).
3. Visual studio 2015 or Visual Studio 2017(download VS 2017 [here](#)).
4. MySQL database.

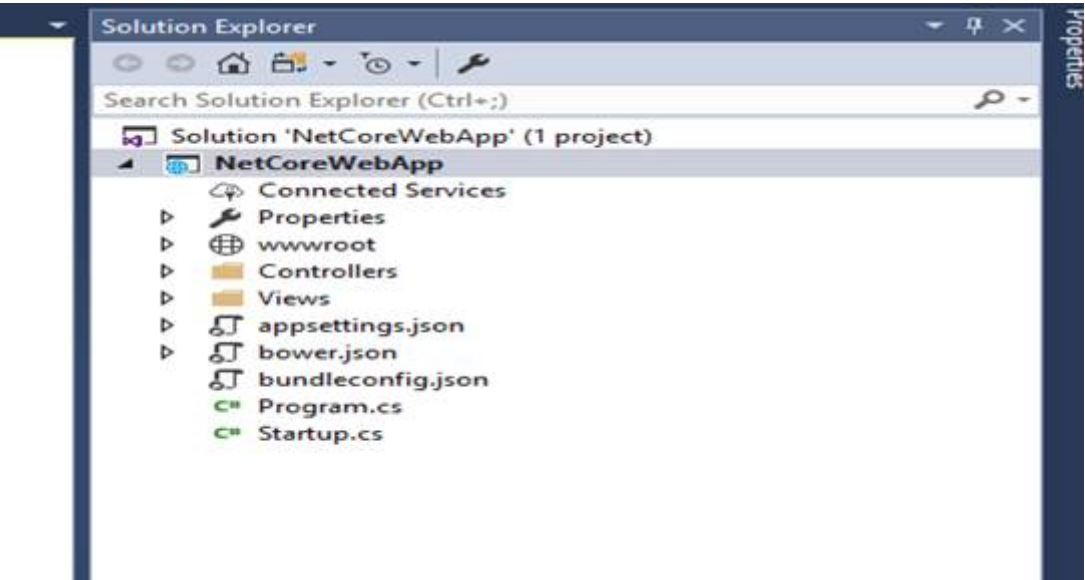
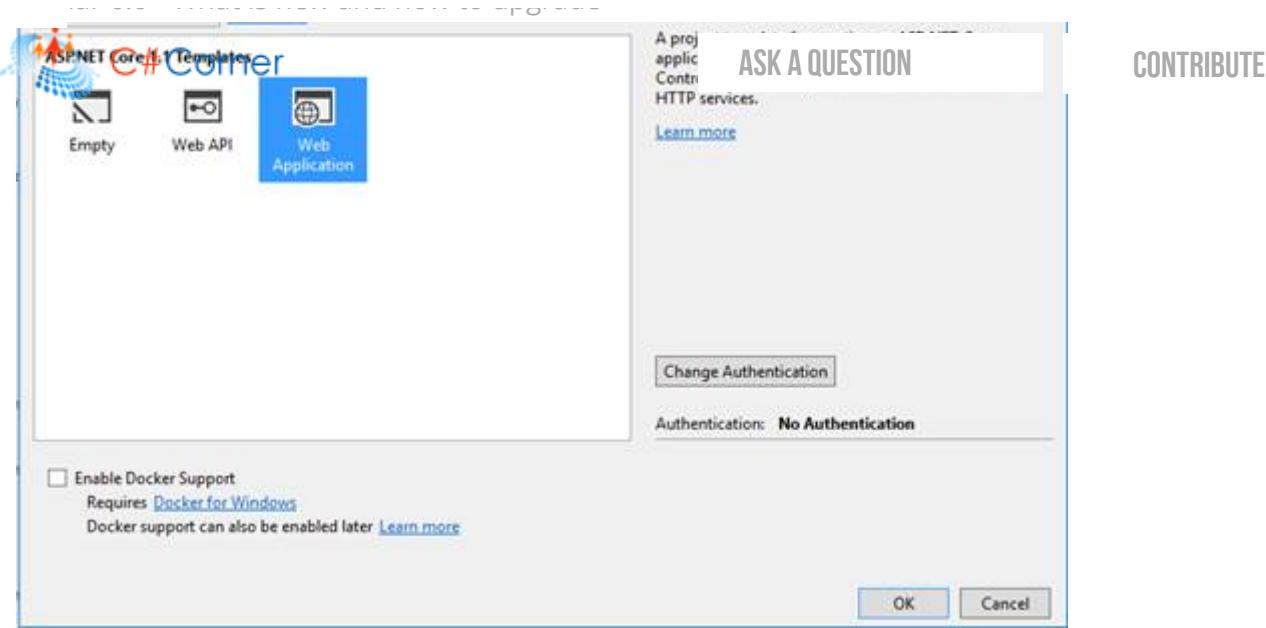
Steps to create .NET Core Web app with VS 2017 are given below.

Go to File -> New Project -> Select Web from templates->choose ASP.NET Core Web Application (.NET Core).

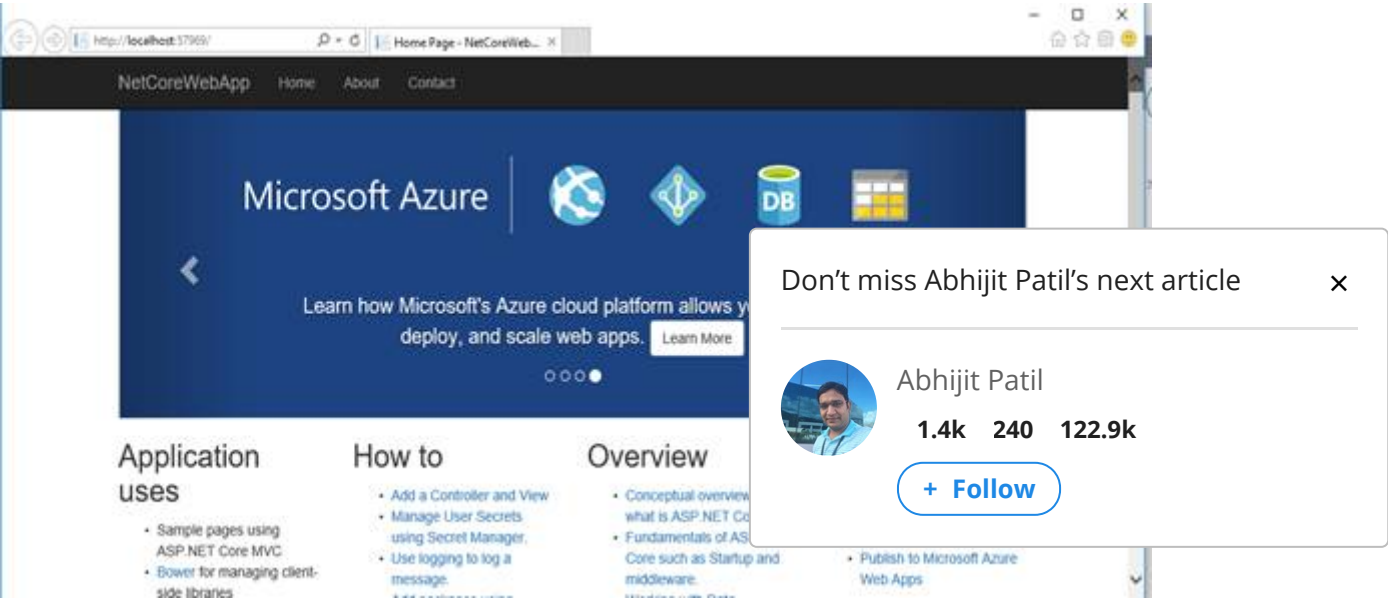
Provide the name for the Web app and click OK.



Select Web Application template and click OK. It will create .NET Core Web app for you.



Clean build and run the Application for testing.



to create MySQL database, we must have IDE for MySQL or we need to have MySQL CLI. We can create the database, either with IDE or using CLI commands: [ASK A QUESTION](#) [CONTRIBUTE](#)

We can create the database in both ways. Let's check out CLI first.

1. To create database, using CLI, we first need to login with password, which we have provided at the time of installation.
2. After login, type the command "create database MusicStoreDB;"

```
mysql> create database MusicStoreDB;
Query OK, 1 row affected (0.01 sec)
```

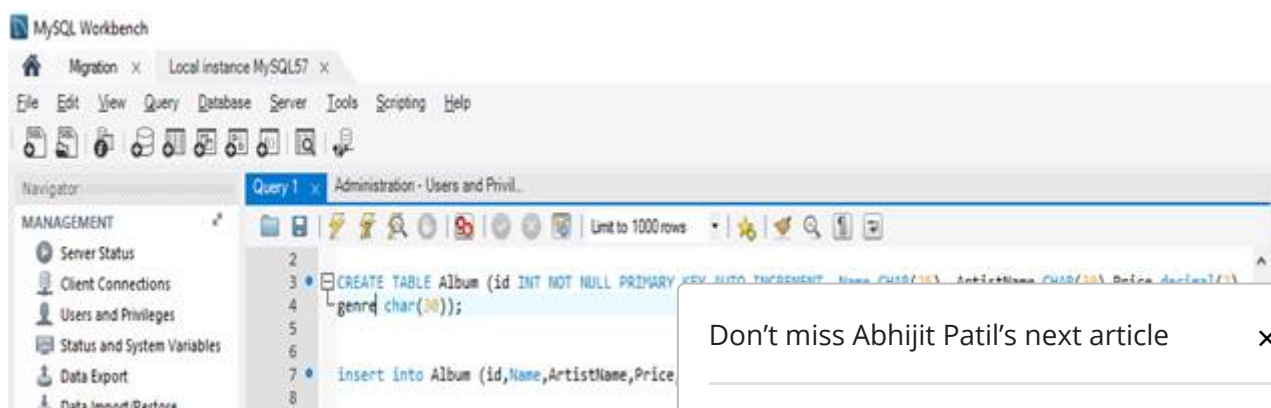
3. To check if DB is created or not, run the command given below.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| musicstoredb      |
| mysql            |
| performance_schema |
| sakila           |
| sys              |
| world            |
+-----+
```

4. Similarly, we run SQL queries on CLI, create tables and insert data into it.
5. Now, let's check with MySQL workbench IDE
6. Select Query template. Write SQL query on query editor. Click Lightning icon.



on top menu.



7. After creating table, insert records in created DB, run


Don't miss Abhijit Patil's next article ×



Abhijit Patil

1.4k 240 122.9k

+ Follow



[ASK A QUESTION](#)
[CONTRIBUTE](#)

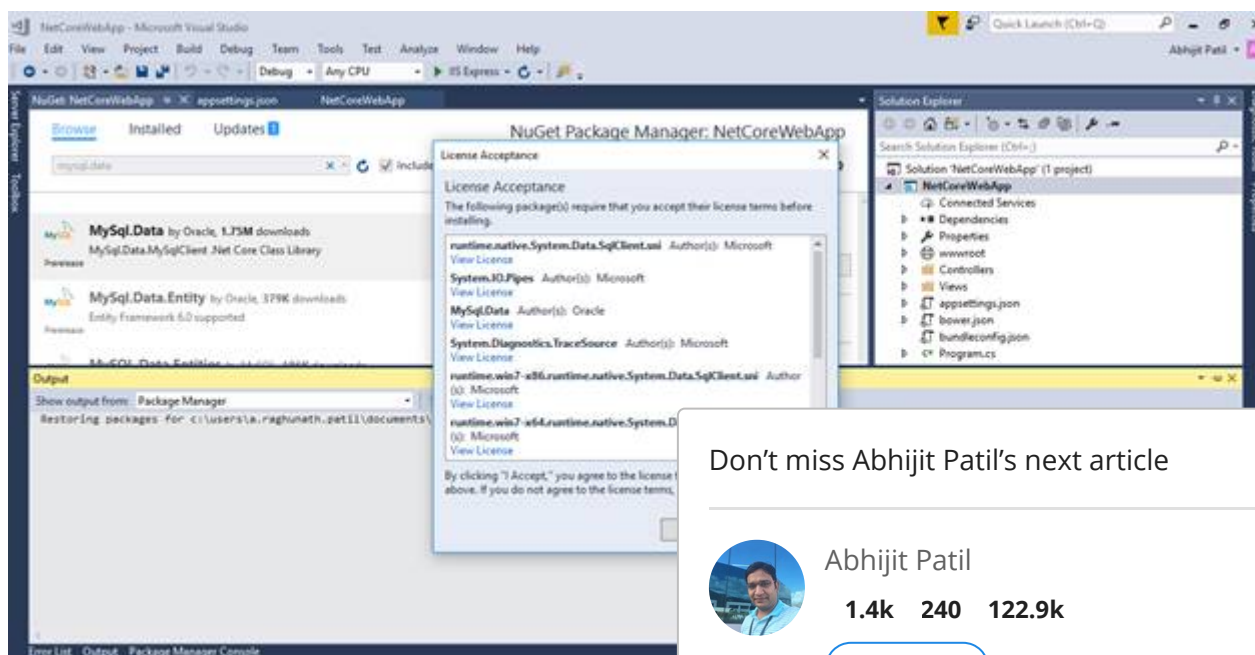
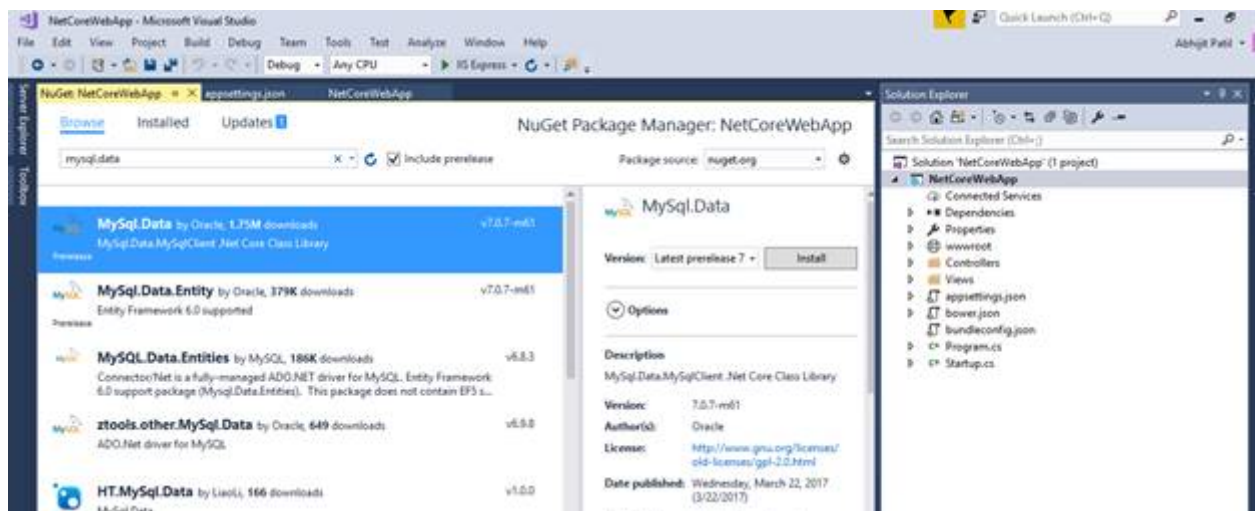
	id	Name	ArtistName	Price	genre
	1	One	AKON	15	DOO
	2	MvLove	AKON	15	DOO
	3	destinv	AKON	13	DOO
	4	MvSonas	John	15	DOO
		NULL	NULL	NULL	NULL

8. For now, we did for single table only. We are done with the database creation and data insertion.

Connect with MySQL

To connect with MySQL database, we must have some NuGet installed in our Application.

1. Go to Application -> right click on project name-> select Manage Nuget Packages-> Type MySql .Data



2. Go to Project root-> appsettings.json-> enter the co

Don't miss Abhijit Patil's next article



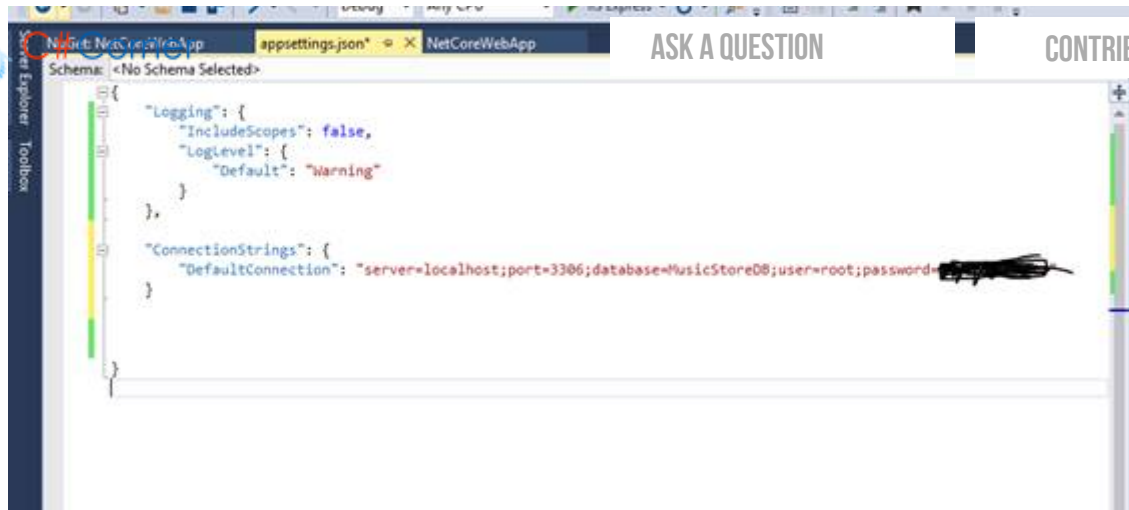
Abhijit Patil

1.4k 240 122.9k

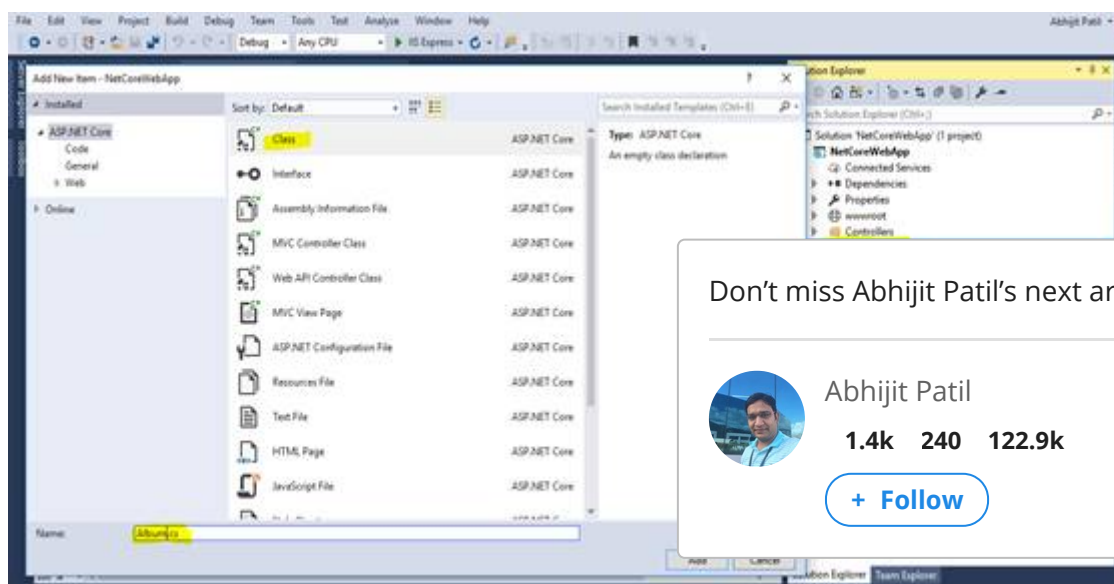
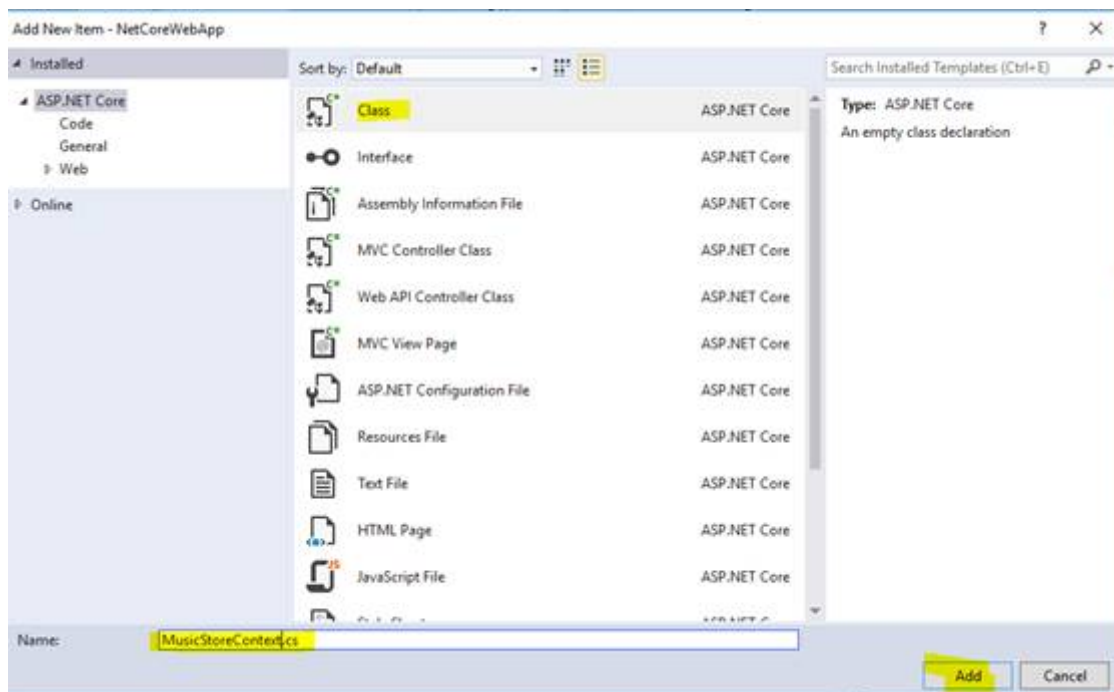
+ Follow

CUS

lar 6.0 - What is new and how to upgrade



3. Create new folder Models and add Class 'MusicStoreContext' & 'Album' in it.



4. Add Album properties in an Album class.

Don't miss Abhijit Patil's next article



Abhijit Patil

1.4k 240 122.9k

+ Follow

CUS

lar 6.0 - What is new and how to upgrade



C# Corner

```

04. {
05.     private MusicStoreContext con
06.
07.     public int Id { get; set; }
08.
09.     public string Name { get; set; }
10.
11.     public string ArtistName { get; set; }
12.
13.     public int Price { get; set; }
14.
15.     public string Genre { get; set; }
16. }
17. }

```

ASK A QUESTION

CONTRIBUTE

1. Create a new MusicStoreContext class, which will contain the connections and MusicStore data entities, as shown below.

```

01. using MySql.Data.MySqlClient;
02. using System;
03. using System.Collections.Generic;
04.
05. namespace NetCoreWebApp.Models
06. {
07.     public class MusicStoreContext
08.     {
09.         public string ConnectionString { get; set; }
10.
11.         public MusicStoreContext(string connectionString)
12.         {
13.             this.ConnectionString = connectionString;
14.         }
15.
16.         private MySqlConnection GetConnection()
17.         {
18.             return new MySqlConnection(ConnectionString);
19.         }
20.
21.     }
22. }
23. }

```

6. To use context in our Application, we need to register instance as a Service in our Application. To register context, we need to add on line of code in 'startup.cs' file under 'ConfigureServices' method.

```

01. using Microsoft.AspNetCore.Builder;
02. using Microsoft.AspNetCore.Hosting;
03. using Microsoft.Extensions.Configuration;
04. using Microsoft.Extensions.DependencyInjection;
05. using Microsoft.Extensions.Logging;
06. using NetCoreWebApp.Models;
07. namespace NetCoreWebApp
08. {
09.     public class Startup
10.     {
11.         public Startup(IHostingEnvironment env)
12.         {
13.             var builder = new ConfigurationBuilder()
14.                 .SetBasePath(env.ContentRootPath)
15.                 .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
16.                 .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)
17.                 .AddEnvironmentVariables();
18.             Configuration = builder.Build();
19.         }
20.
21.         public IConfigurationRoot Configuration { get; }

```

Don't miss Abhijit Patil's next article

x



Abhijit Patil

1.4k 240 122.9k

+ Follow

CUS

lar 6.0 - What is new and how to upgrade



C# Corner

```

25. {
26.     // Add framework services
27.     services.AddMvc();
28.     services.Add(new ServiceDescriptor(typeof(MusicStoreContext), new MusicStoreContext(), ServiceLifetime.Singleton));
29. }
30.
31. // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
32. public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)
33. {
34.     loggerFactory.AddConsole(Configuration.GetSection("Logging"));
35.     loggerFactory.AddDebug();
36.
37.     if (env.IsDevelopment())
38.     {
39.         app.UseDeveloperExceptionPage();
40.         app.UseBrowserLink();
41.     }
42.     else
43.     {
44.         app.UseExceptionHandler("/Home/Error");
45.     }
46.
47.     app.UseStaticFiles();
48.
49.     app.UseMvc(routes =>
50.     {
51.         routes.MapRoute(
52.             name: "default",
53.             template: "{controller=Home}/{action=Index}/{id?}");
54.     });
55. }
56. }
57. }

```

ASK A QUESTION

CONTRIBUTE

Fetch data from My SQL database

5. To get the data from the database, we need 'GetAllAlbums()' method, our DB context, add 'GetAllAlbums()' method in "MusicStoreContext" class.

```

01. public List<Album> GetAllAlbums()
02. {
03.     List<Album> list = new List<Album>();
04.
05.     using (MySQLConnection conn = GetConnection())
06.     {
07.         conn.Open();
08.         MySqlCommand cmd = new MySqlCommand("select * from Album where id < 10", conn);
09.
10.         using (var reader = cmd.ExecuteReader())
11.         {
12.             while (reader.Read())
13.             {
14.                 list.Add(new Album()
15.                 {
16.                     Id = Convert.ToInt32(reader["Id"]),
17.                     Name = reader["Name"],
18.                     ArtistName = reader["ArtistName"],
19.                     Price = Convert.ToDecimal(reader["Price"]),
20.                     Genre = reader["Genre"],
21.                 });
22.             }
23.         }
24.     }
25.     return list;
26. }

```

Don't miss Abhijit Patil's next article



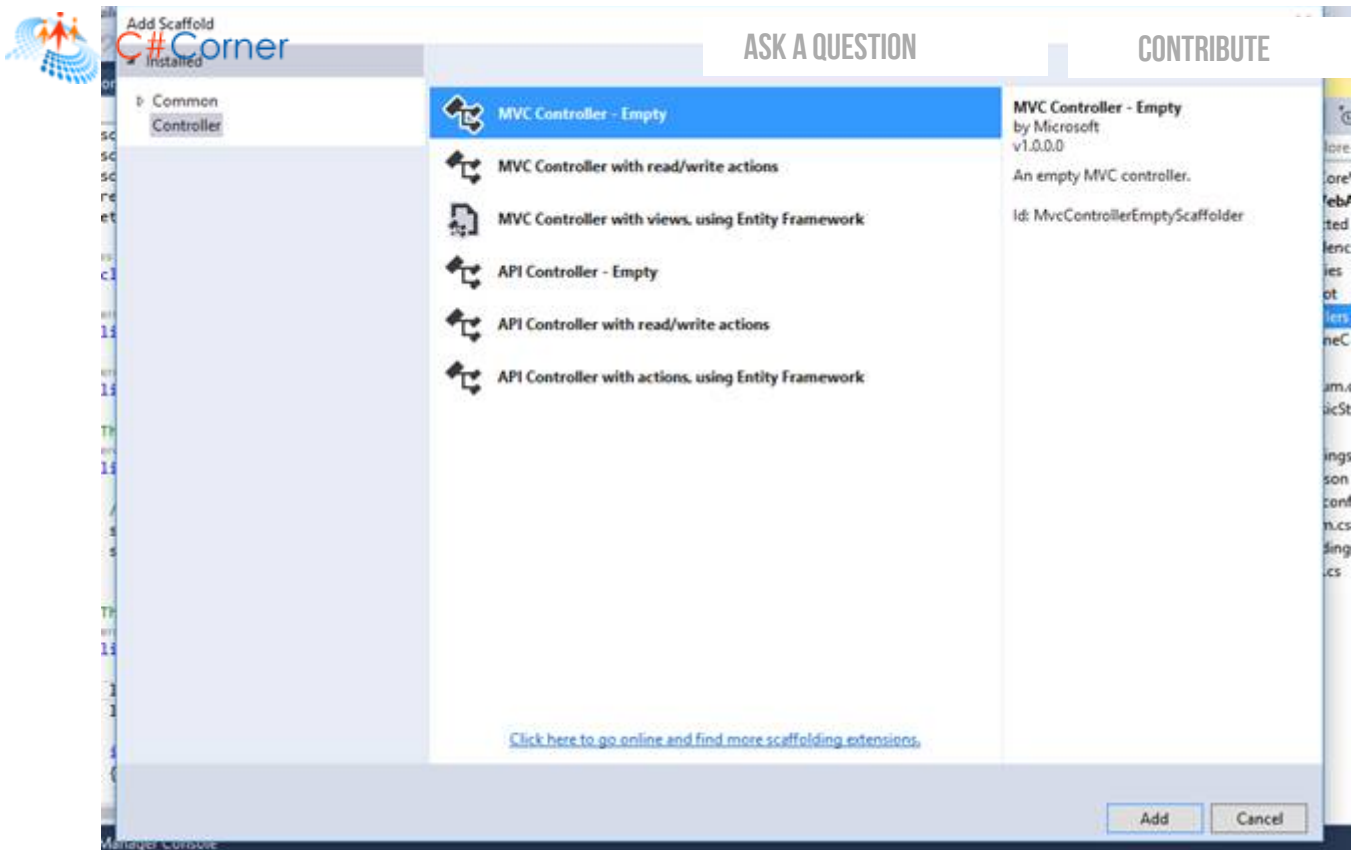
Abhijit Patil

1.4k 240 122.9k

+ Follow

CUS

lar 6.0 - What is new and how to upgrade



Add the code given below in Album Controller to get the data from DB.

```

01. namespace NetCoreWebApp.Controllers
02. {
03.     public class AlbumController : Controller
04.     {
05.         public IActionResult Index()
06.         {
07.             MusicStoreContext context = HttpContext.RequestServices.GetService(typeof(Net
08.
09.             return View(context.GetAllAlbums());
10.         }
11.     }
12.
13. }
14. }

```

6. After adding controller and code, we need view to display the data to the end user. Create folder under Views with name Albums. Right click on Albums folder and add new view Albums.

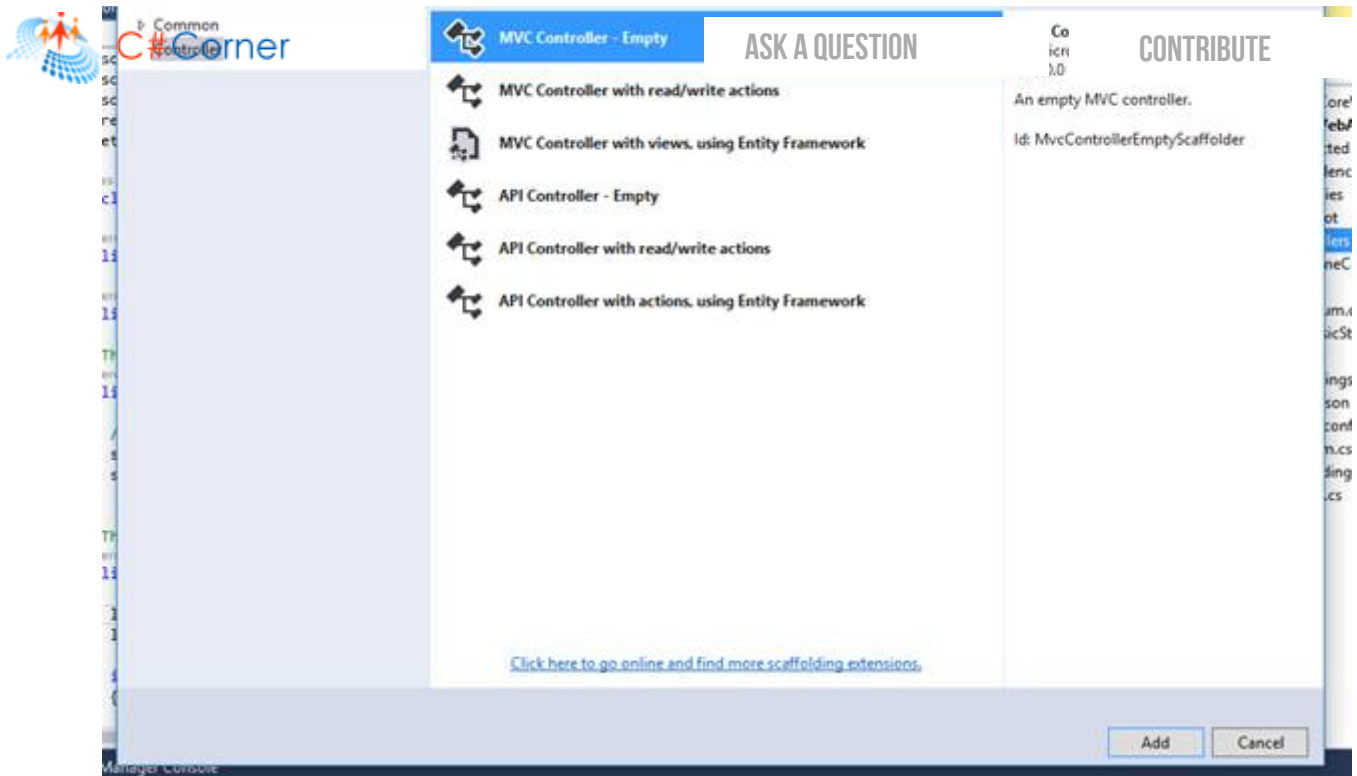
7. Select Layout page by clicking button. Now, click add.

Don't miss Abhijit Patil's next article

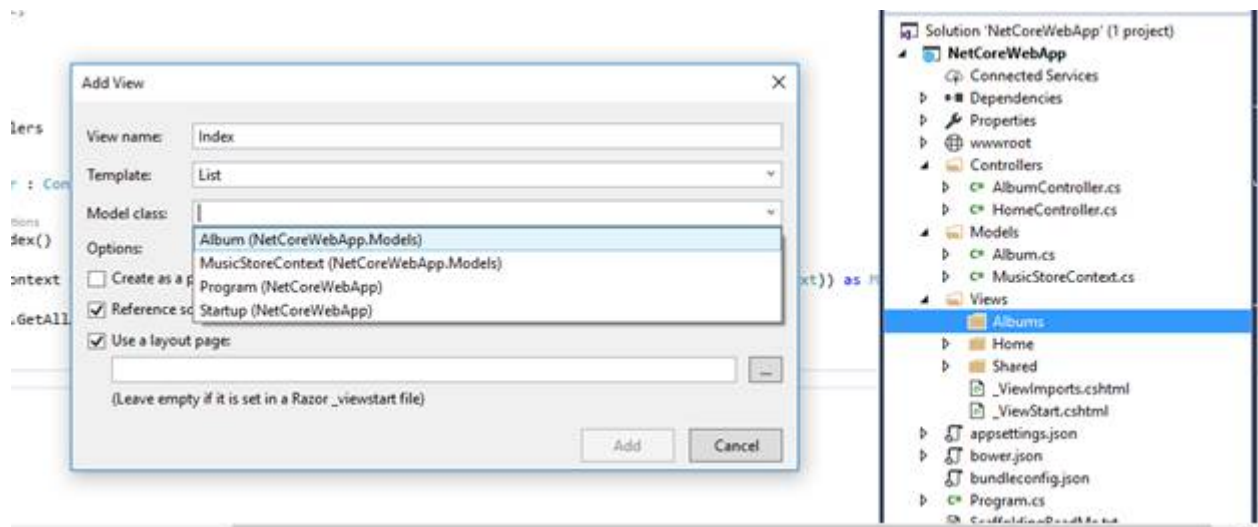
Abhijit Patil
1.4k 240 122.9k
+ Follow

CUS

lar 6.0 - What is new and how to upgrade



8. You can create view from by selecting data model in dropdown or you can create blank view with the default index name and add the code, as shown below.



9. To route to our Album action, we need to update Startup.cs and add Album controller name, so we after running an app directly. We get the "Album/index" page.

```
01. app.UseMvc(routes =>
02.     {
03.         routes.MapRoute(
04.             name: "default",
05.             template: "{controller=
06.         });
```

10. Now, just run an Application and we will get the out

Don't miss Abhijit Patil's next article



Abhijit Patil

1.4k 240 122.9k

+ Follow

CUS

lar 6.0 - What is new and how to upgrade

C# Corner

ViewData["Title"] = "Index";
Layout = "~/Views/Shared/_Layout.cshtml"

<h2>Albums</h2>
<table class="table">
 <tr>
 <th>ID</th>
 <th>Name</th>
 <th>ArtistName</th>
 <th>Price</th>
 <th>Genre</th>
 </tr>
 @foreach (var item in Model)
 {
 <tr>
 <td>
 @Html.DisplayTextFor(modelItem => item.Id)
 </td>
 <td>
 @Html.DisplayFor(modelItem => item.Name)
 </td>
 <td>
 @Html.DisplayFor(modelItem => item.ArtistName)
 </td>
 <td>
 @Html.DisplayFor(modelItem => item.Price)
 </td>
 <td>...</td>
 </tr>
 }
</table>

ASK A QUESTION

CONTRIBUTE

Conclusion

In this article, we have seen how to make the connection with MYSQL Server database. In case of remote, we need to update the connection string with an appropriate Server name and the port name.

Feel free to share suggestions and feedback.

ASP.NET Core

MySQL



Abhijit Patil

<https://www.c-sharpcorner.com/members/abhijit-p>

1411

122.9k

5

3



Type your comment here and press Enter Key



Avinash Barnwal

For .Net Core 2.x I've migrate to use Async Poco. Change all related code to async generate some error.



Richardson dos Santos Neves

1597 54 0

Don't miss Abhijit Patil's next article



Abhijit Patil

1.4k 240 122.9k

+ Follow


<https://www.c-sharpcorner.com/article/how-to-connect-mysql-with-asp-net-core/>

May 02

0 0

10/13

CUS
lar 6.0 - What is new and how to upgrade

 Richardson dos Santos Neves
1597 54 0

ASK A QUESTION

CONTRIBUTE



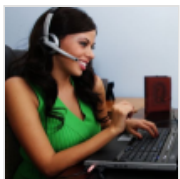
Does it still work with .netcore2.0 ?

Avinash Barnwal

1636 15 0

Oct 21, 2017

1 0 Reply



Hire a blockchain developer

Want to get your applications built on Blockchain technology? Come, meet the blockchain experts and hire a team or individuals as per your need.

TRENDING UP

- 01 ASP.NET Core And Web API - A Custom Wrapper For Managing Exceptions And Consistent Responses
- 02 Git For Absolute Beginners With Command Line Interface
- 03 Create Your First Smart Contract In Ethereum With Ganache And Remix IDE
- 04 CRUD Operations In ASP.NET Core Web API Using ADO.NET
- 05 Stratis - Introduction To Stratis Platform
- 06 ASP.NET Core 2 - Architecture And Design Pattern Ideology
- 07 Trick To Use Dark Theme In SQL Server Management Studio
- 08 Create Windows Services In C#
- 09 Blockchain Development - Programming Smart Contract In C# Part Two
- 10 What Is The Future Of Blockchain

Don't miss Abhijit Patil's next article



Abhijit Patil


1.4k 240 122.9k

+ Follow

View All


CUSS

lar 6.0 - What is new and how to upgrade




70% of enterprises plan to implement Blockchain.

Is your business Blockchain ready?



Leader in Blockchain enterprise development



ASK A QUESTION

CONTRIBUTE

Philadelphia

New York

London


Delhi

JOIN C# CORNER

and millions of developer friends worldwide.

Enter your email address

Don't miss Abhijit Patil's next article



Abhijit Patil

1.4k 240 122.9k

+ Follow

Learn ASP.NET MVC

Learn ASP.NET Core

Learn Python

Learn Oracle

Home

Events

Consultants

Jobs

Calendar

lar 6.0 - What is new and how to upgrade



[Privacy Policy](#) [Terms](#)

ASK A QUESTION

CONTRIBUTE

©2018 C# Corner. All contents are copyright of their authors.

Don't miss Abhijit Patil's next article [x](#)



Abhijit Patil

1.4k 240 122.9k

+ Follow