**Database Connectivity With And Without Entity Framework In MVC**

Introduction

I am writing this article because a beginner requested me to do so. Hopefully, this will be helpful for all MVC beginners. First, I will discuss the process without Entity Framework and then, I will discuss the same with Entity Framework. I will just retrieve data from the SQL database. I am not going to explain CRUD (Create Update Read and Delete) operations in this article.

Database connection without Entity Framework

**Step 1**

Open up your favorite SQL Server database with any version. It really doesn’t matter what version it is.

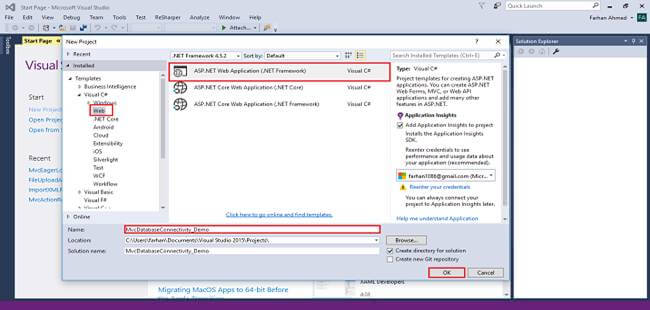
1. **CREATE** **TABLE** [dbo].[Employees](
2. [EmployeeId] [**int**] IDENTITY(1,1) NOT NULL,
3. [**Name**] [nvarchar](50) NULL,
4. [Gender] [**char**](10) NULL,
5. [Age] [**int**] NULL,
6. [Position] [nvarchar](50) NULL,
7. [Office] [nvarchar](50) NULL,
8. [HireDate] [datetime] NULL,
9. [Salary] [**int**] NULL,
10. **CONSTRAINT** [PK\_\_Employee\_\_7AD04F11A993CC91] **PRIMARY** **KEY** CLUSTERED
11. (
12. [EmployeeId] **ASC**
13. )**WITH** (PAD\_INDEX = **OFF**, STATISTICS\_NORECOMPUTE = **OFF**, IGNORE\_DUP\_KEY = **OFF**, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]
14. ) **ON** [**PRIMARY**]
16. GO

**Step 1**

Open Visual Studio 2015 or an editor of your choice and create a new project.

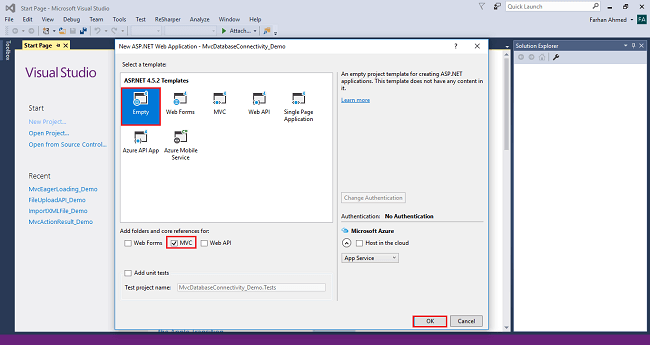
**Step 2**

Choose the "web application" project and give an appropriate name to your project.



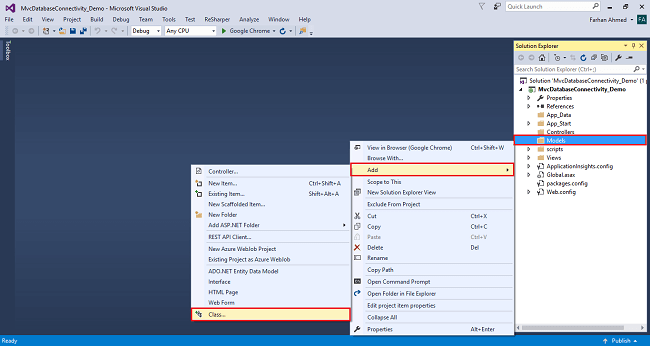
**Step 3**

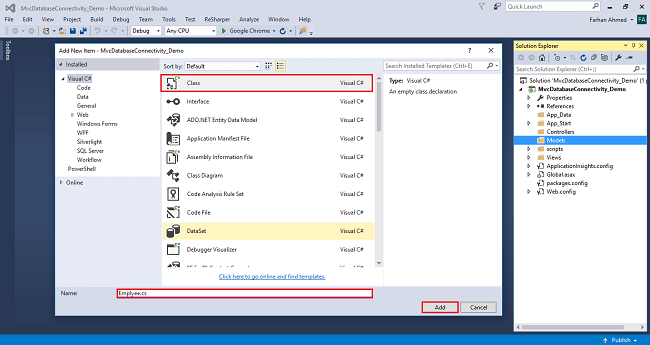
Select "empty" template, check on MVC checkbox and click OK.



**Step 4**

Right-click the Models folder and add a class with name employee and create similar property and data type as we have in SQL database table.





1. **using** System;
3. **namespace** MvcDatabaseConnectivity\_Demo.Models
4. {
5. **public** **class** Emplyee
6. {
7. **public** **int** EmployeeId { **get**; **set**; }
8. **public** **string** Name { **get**; **set**; }
9. **public** **string** Gender { **get**; **set**; }
10. **public** **int** Age { **get**; **set**; }
11. **public** **string** Position { **get**; **set**; }
12. **public** **string** Office { **get**; **set**; }
13. **public** DateTime HireDate { **get**; **set**; }
14. **public** **int** Salary { **get**; **set**; }
15. }
16. }

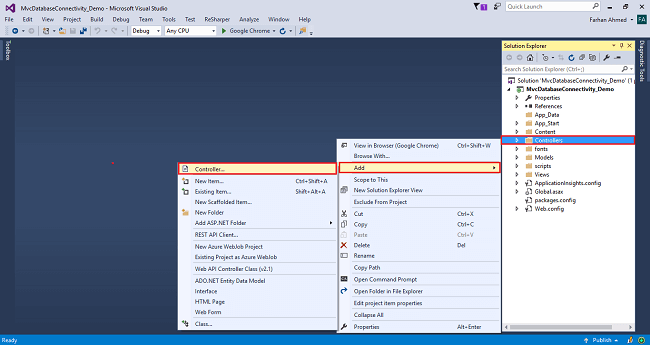
**Step 5**

Open Web.config file and add your database connection, and write the below code

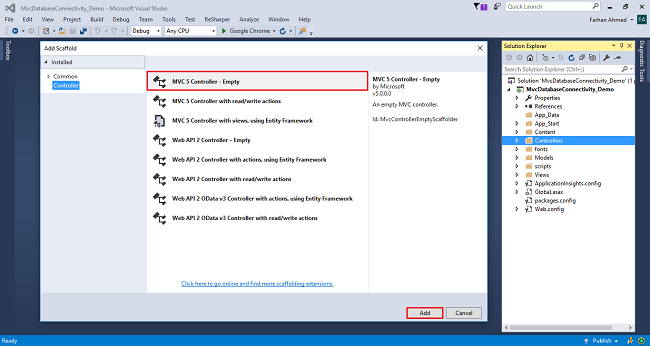
1. <connectionStrings>
2. <add name ="DBCS" connectionString="data source=FARHAN\SQLEXPRESS; database=Sample; Integrated Security=true" providerName="System.Data.SqlClient"/>
3. </connectionStrings>

**Step 6**

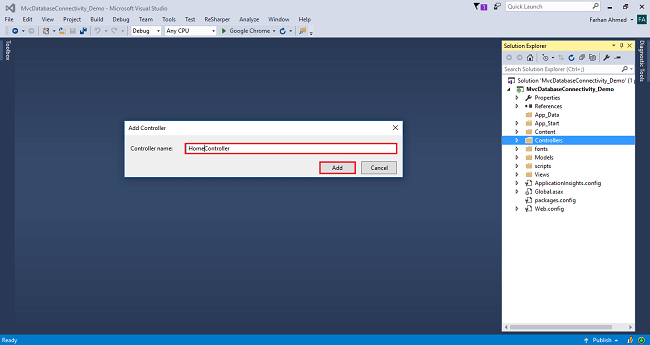
Right-click on the Controllers folder and add a controller.



A window will appear. Choose MVC5 Controller-Empty and click "Add".



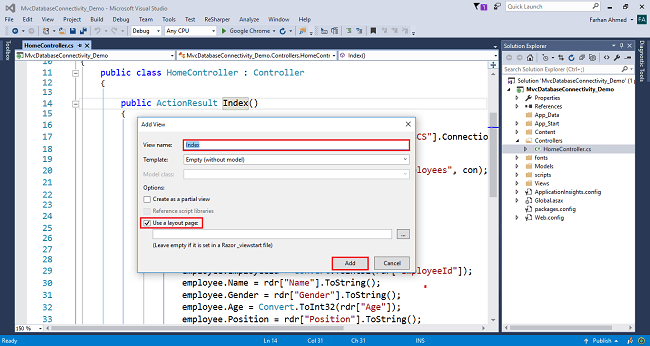
After clicking on "Add", another window will appear with DefaultController. Change the name to HomeController and click "Add". The HomeController will be added under the *Controllers* folder. Don’t change the Controller suffix for all controllers, change only the highlight, and instead of Default, just change Home:



1. **using** MvcDatabaseConnectivity\_Demo.Models;
2. **using** System;
3. **using** System.Collections.Generic;
4. **using** System.Configuration;
5. **using** System.Data;
6. **using** System.Data.SqlClient;
7. **using** System.Web.Mvc;
9. **namespace** MvcDatabaseConnectivity\_Demo.Controllers
10. {
11. **public** **class** HomeController : Controller
12. {
14. **public** ActionResult Index()
15. {
16. List<Emplyee> employeeList = **new** List<Emplyee>();
17. **string** CS = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;
18. **using** (SqlConnection con = **new** SqlConnection(CS))
19. {
20. SqlCommand cmd = **new** SqlCommand("SELECT \* FROM Employees", con);
21. cmd.CommandType = CommandType.Text;
22. con.Open();
24. SqlDataReader rdr = cmd.ExecuteReader();
25. **while** (rdr.Read())
26. {
27. var employee = **new** Emplyee();
29. employee.EmployeeId = Convert.ToInt32(rdr["EmployeeId"]);
30. employee.Name = rdr["Name"].ToString();
31. employee.Gender = rdr["Gender"].ToString();
32. employee.Age = Convert.ToInt32(rdr["Age"]);
33. employee.Position = rdr["Position"].ToString();
34. employee.Office = rdr["Office"].ToString();
35. employee.HireDate =Convert.ToDateTime(rdr["HireDate"]);
36. employee.Salary = Convert.ToInt32(rdr["Salary"]);
37. employeeList.Add(employee);
38. }
39. }
40. **return** View(employeeList);
41. }
42. }
43. }

**Step 7**

Right-click on the Index method in HomeController; the "Add View" window will appear with the default index name checked (use a Layout page). Click on "Add.

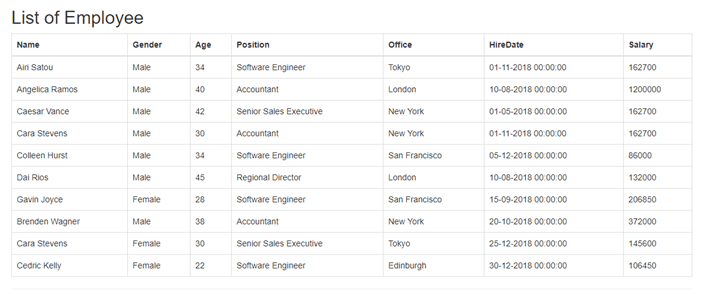


**Code for Index View**

1. @model IEnumerable<MvcDatabaseConnectivity\_Demo.Models.Employee>
2. @{
3. ViewBag.Title = "Index";
4. }
6. <h2>List of Employee</h2>
7. <table **class**="table table-bordered">
8. <thead>
9. <tr>
10. <th>@Html.DisplayNameFor(m => m.Name)</th>
11. <th>@Html.DisplayNameFor(m => m.Gender)</th>
12. <th>@Html.DisplayNameFor(m => m.Age)</th>
13. <th>@Html.DisplayNameFor(m => m.Position)</th>
14. <th>@Html.DisplayNameFor(m => m.Office)</th>
15. <th>@Html.DisplayNameFor(m => m.HireDate)</th>
16. <th>@Html.DisplayNameFor(m => m.Salary)</th>
17. </tr>
18. </thead>
19. <tbody>
20. @foreach (**var** emp **in** Model)
21. {
22. <tr>
23. <td>@emp.Name</td>
24. <td>@emp.Gender</td>
25. <td>@emp.Age</td>
26. <td>@emp.Position</td>
27. <td>@emp.Office</td>
28. <td>
29. @**if** (emp.HireDate != **null**)
30. {
31. @emp.HireDate
32. }
33. </td>
34. <td>@emp.Salary</td>
35. </tr>
36. }
37. </tbody>
38. </table>

**Step 8**

Build and run the project by pressing Ctrl+F5.



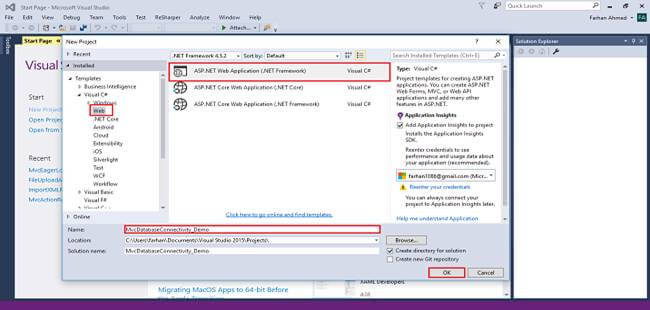
Database connection with Entity Framework

**Step 1**

Open Visual Studio 2015 or an editor of your choice and create a new project.

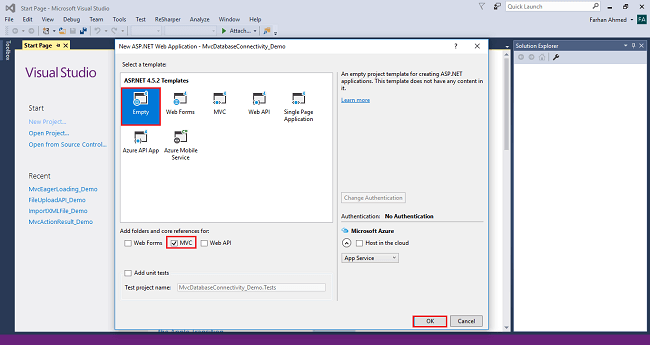
**Step 2**

Choose the "web application" project and give an appropriate name to your project.



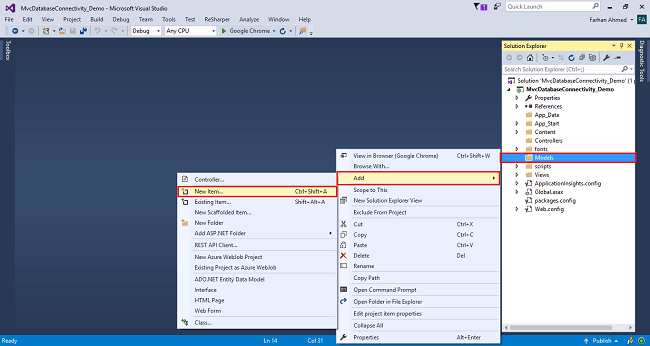
**Step 3**

Select "empty" template, check on MVC checkbox and click OK.

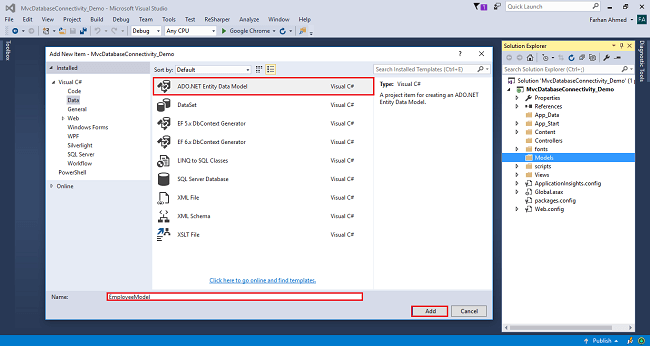


**Step 4**

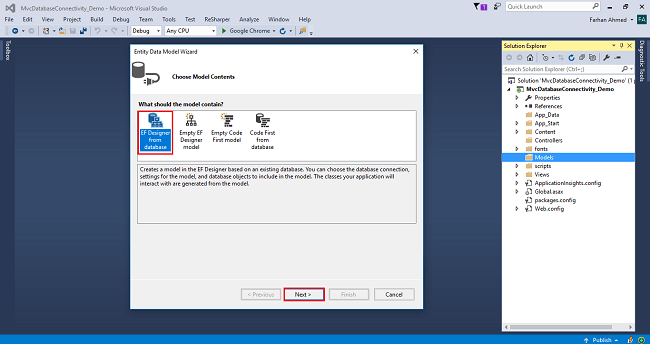
Right-click the Models folder and add a database model. Add Entity Framework now. For that, right-click on Models folder, select Add, then select "New Item".



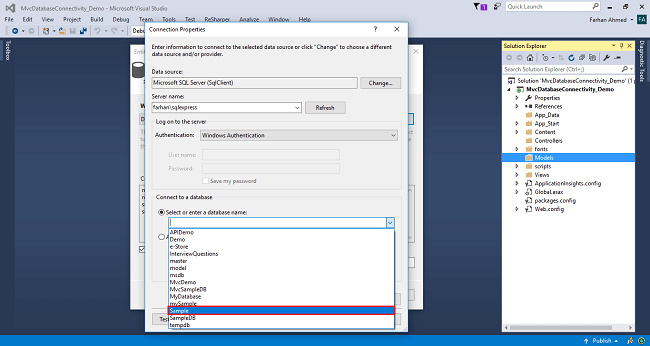
You will get a window; from there, select Data from the left panel and choose ADO.NET Entity Data Model, give it the name EmployeeModel (this name is not mandatory, you can give any name) and click "Add".



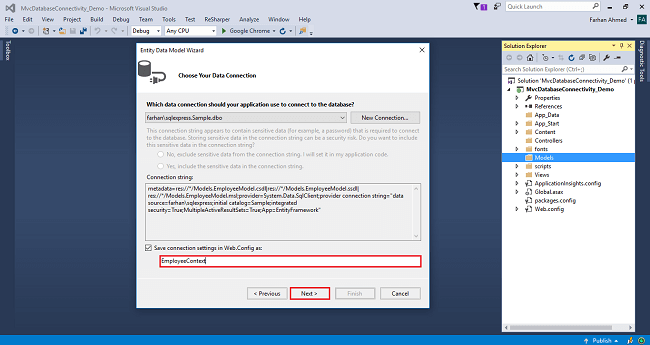
After you click on "Add a window", the wizard will open. Choose EF Designer from the database and click "Next".



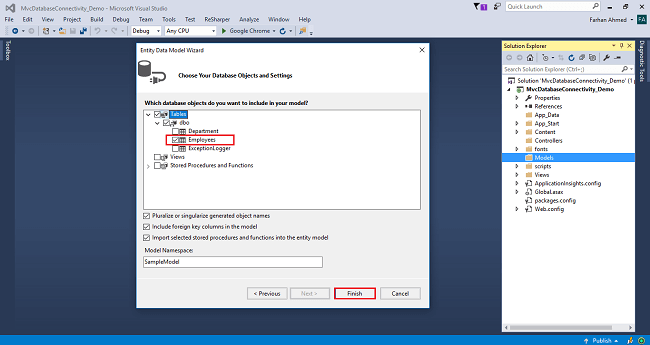
After clicking on "Next", a window will appear. Choose "New Connection". Another window will appear. Add your server name - if it is local, then enter a dot (.). Choose your database and click "OK".



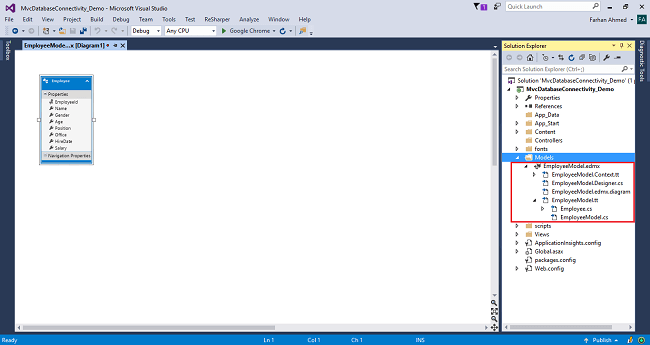
The connection will be added. If you wish, save the connection name as you want. You can change the name of your connection below. It will save the connection in the web config. Now, click "Next".



After clicking on NEXT, another window will appear. Choose the database table name as shown in the below screenshot and click "Finish".



Entity Framework gets added and the respective class gets generated under the Models folder.

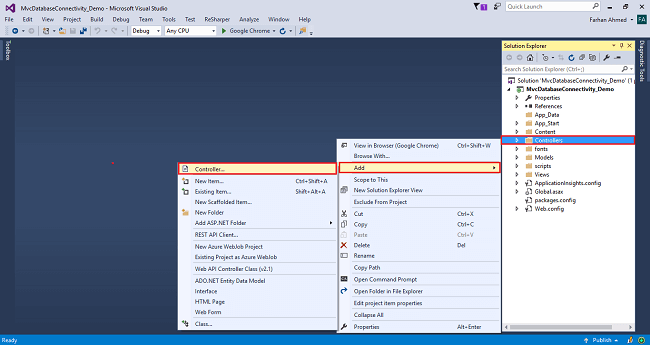


**Employee class**

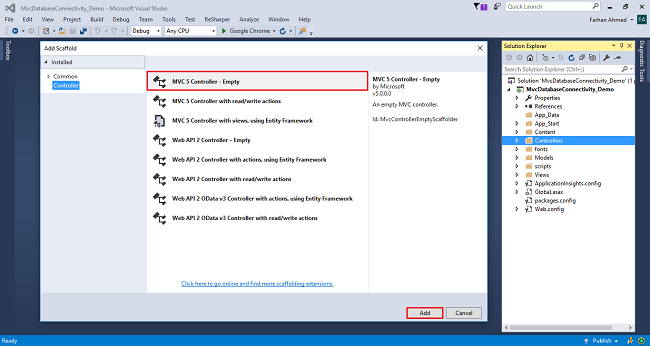
1. **namespace** MvcDatabaseConnectivity\_Demo.Models
2. {
3. **using** System;
5. **public** partial **class** Employee
6. {
7. **public** **int** EmployeeId { **get**; **set**; }
8. **public** **string** Name { **get**; **set**; }
9. **public** **string** Gender { **get**; **set**; }
10. **public** Nullable<**int**> Age { **get**; **set**; }
11. **public** **string** Position { **get**; **set**; }
12. **public** **string** Office { **get**; **set**; }
13. **public** Nullable<System.DateTime> HireDate { **get**; **set**; }
14. **public** Nullable<**int**> Salary { **get**; **set**; }
15. }
16. }

**Step 5**

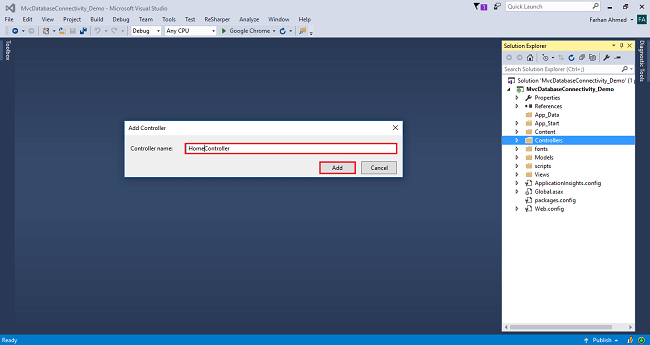
Right-click on Controllers folder and add a controller.



A window will appear. Choose MVC5 Controller-Empty and click "Add".



After clicking on "Add", another window will appear with DefaultController. Change the name to HomeController and click "Add". The HomeController will be added under the *Controllers* folder. Don’t change the Controller suffix for all controllers, change only the highlight, and instead of Default, just change Home,



**Complete controller code**

1. **using** MvcDatabaseConnectivity\_Demo.Models;
2. **using** System.Linq;
3. **using** System.Web.Mvc;
5. **namespace** MvcDatabaseConnectivity\_Demo.Controllers
6. {
7. **public** **class** HomeController : Controller
8. {
9. **private** **readonly** EmployeeContext \_dbContext = **new** EmployeeContext();
11. **public** ActionResult Index()
12. {
13. var employee = \_dbContext.Employees.ToList();
14. **return** View(employee);
15. }
16. }
17. }

**Step 6**

Right-click on the Index method in HomeController; the "Add View" window will appear with default index name checked (use a Layout page). Click on "Add.

1. @model IEnumerable<MvcDatabaseConnectivity\_Demo.Models.Employee>
2. @{
3. ViewBag.Title = "Index";
4. }
6. <h2>List of Employee</h2>
7. <table **class**="table table-bordered">
8. <thead>
9. <tr>
10. <th>@Html.DisplayNameFor(m => m.Name)</th>
11. <th>@Html.DisplayNameFor(m => m.Gender)</th>
12. <th>@Html.DisplayNameFor(m => m.Age)</th>
13. <th>@Html.DisplayNameFor(m => m.Position)</th>
14. <th>@Html.DisplayNameFor(m => m.Office)</th>
15. <th>@Html.DisplayNameFor(m => m.HireDate)</th>
16. <th>@Html.DisplayNameFor(m => m.Salary)</th>
17. </tr>
18. </thead>
19. <tbody>
20. @foreach (**var** emp **in** Model)
21. {
22. <tr>
23. <td>@emp.Name</td>
24. <td>@emp.Gender</td>
25. <td>@emp.Age</td>
26. <td>@emp.Position</td>
27. <td>@emp.Office</td>
28. <td>
29. @**if** (emp.HireDate != **null**)
30. {
31. @emp.HireDate
32. }
33. </td>
34. <td>@emp.Salary</td>
35. </tr>
36. }
37. </tbody>
38. </table>

**Step 7**

Build and run the project by pressing Ctrl+F5.

