(T7)討論 EfDbFirst 的 CRUD(Create,Read,Update,Delete)
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924

==

(T7)討論 EfDbFirst 的 CRUD(Create,Read,Update,Delete)

==

0. Summary
0.1. Summary
0.2. MVC conventions
----
1. OnlineGame DB
1.1. TSQL
1.2. Security login
----
2. New Project - OnlineGame
2.1. New Project - OnlineGame.Web
2.1.1. Global.asax.cs
2.1.2. App_Start/RouteConfig.cs
2.2. ADO.Net Entity Data Model - Entity Framework
2.3. Controllers/GamersController.cs
----
3. OnlineGame.Web
3.1. Models/Teams/Team.cs
3.2. Models/Teams/TeamMetaData.cs
3.3. Views/Gamers/Create.cshtml
3.4. Models/Gamers/Gamer.cs
3.5. Models/Gamers/GamerMetaData.cs
3.6. Views/Gamers/Edit.cshtml
3.7. Models/Teams/TeamTotals.cs
3.8. Controllers/GamersController.cs
3.9. Views/Gamers/GamersByTeam.cshtml

==

0. Summary

0.1. Summary

In this tutorial, we will discuss
* MvcConventions
* AdoDotNetEntityDataModel
* EntityFramework
* AutoGenerate Delete, Update, Insert, Read
Tutorial 7: 資料庫連結 3 - 自動生成程式碼秒殺 Entity Framework 以及新增/更新/移除資料
本課程一開始先帶你手把手完全手寫 Entity Framework 和 ADO.NET。等你熟悉後，教你如何使用自動生成的程式碼秒殺 Entity Framework 新增/更新/移除。因此本 MVC 的課程提供學生多種連接資料庫的選擇。之後的教學還討論當你的資料庫結構改變後，要如何 update 現有的 Entity Framework 程式碼!?如此完整的課程值得你的投資。

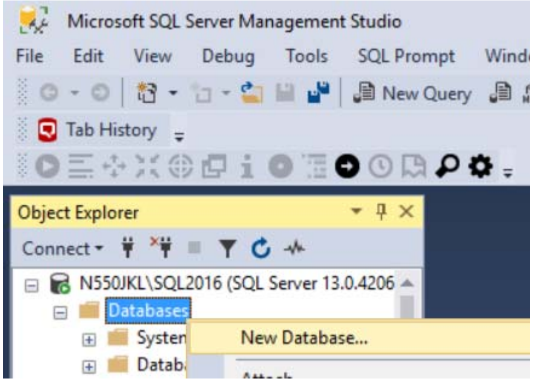
0.2. MVC conventions

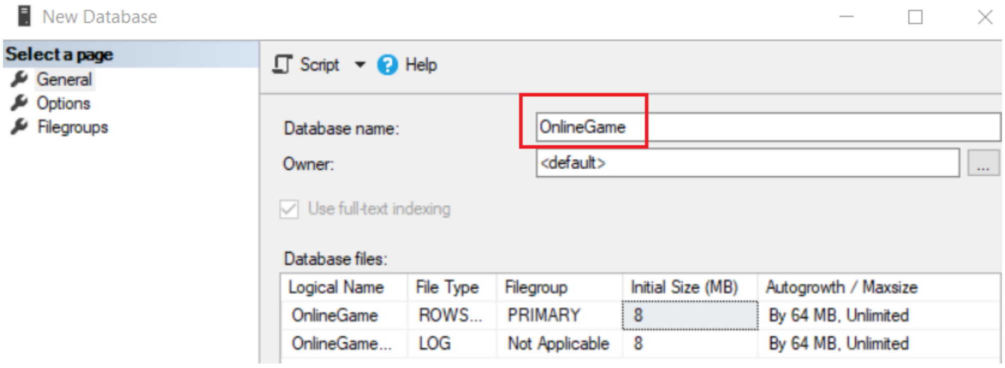
In MVC conventions,
1. Controllers must have the word "Controller" as the suffix and must extend "IController" interface.
2. A view must remain under "Views" folder.
3. If the view is for GamerController, then the view must remain under "Views/Gamer" folder.
4. In the "HomeController", when "Index" action "return View()", it will search the following files in order.
   4.1. ~/Views/Home/Index.aspx
   4.2. ~/Views/Home/Index.ascx
   4.3. ~/Views/Shared/Index.aspx
   4.4. ~/Views/Shared/Index.ascx
   4.5. ~/Views/Home/Index.cshtml
   4.6. ~/Views/Home/Index.vbhtml
   4.7. ~/Views/Shared/Index.cshtml
   4.8. ~/Views/Shared/Index.vbhtml
5. By MVC convention, MVC will look for the view in the following locations
   5.1. Views/ControllerName
   5.2. Views/Shared
6. The extension name of view can be cshtml, vbhtml, aspx, or ascx.
7. Models can be anywhere, even can be in another project. However, it is better to put it in "Models" folder.
8. You may put Models in another project as business layer.
9. Shared folder stores shared views.
   E.g. Master for aspx and Layout pages for cshtml

1. OnlineGame DB

1.1. TSQL

In SQL server Management Studio (SSMS)
Database --> Right Click --> New Database -->
In General Tab -->
Name: **OnlineGame**
In options Tab --> Recovery model : **Simple**





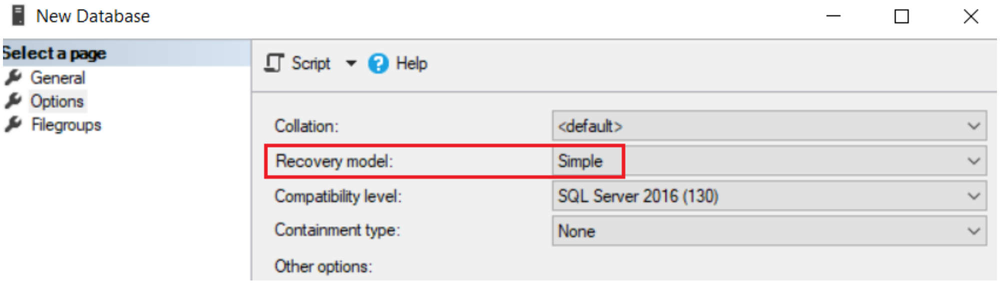
Database name: OnlineGame

Owner: <default>

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
OnlineGame	ROWS...	PRIMARY	8	By 64 MB, Unlimited
OnlineGame...	LOG	Not Applicable	8	By 64 MB, Unlimited



Collation: <default>

Recovery model: Simple

Compatibility level: SQL Server 2016 (130)

Containment type: None

Other options:

```
--1. Drop if it exists
--Drop Table if it exists.
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Gamer' ) )
    BEGIN
        TRUNCATE TABLE Gamer;
        DROP TABLE Gamer;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Team' ) )
    BEGIN
        TRUNCATE TABLE Team;
        DROP TABLE Team;
    END;
GO -- Run the previous command and begins new batch
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spSearchGamer') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spGetGamers' ) )
    BEGIN
        DROP PROCEDURE spGetGamers;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
```

```
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spAddGamer' ) )
    BEGIN
        DROP PROCEDURE spAddGamer;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spSaveGamer' ) )
    BEGIN
        DROP PROCEDURE spSaveGamer;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spDeleteGamer' ) )
    BEGIN
        DROP PROCEDURE spDeleteGamer;
    END;
GO -- Run the previous command and begins new batch
--2. Create Table
CREATE TABLE Team
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1)
             NOT NULL ,
      [Name] NVARCHAR(100) NULL
    );
GO -- Run the previous command and begins new batch
CREATE TABLE Gamer
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1)
             NOT NULL ,
      [Name] NVARCHAR(100) NULL ,
      Gender NVARCHAR(10) NULL ,
      City NVARCHAR(50) NULL ,
      DateOfBirth DATETIME NULL ,
      TeamId INT FOREIGN KEY REFERENCES Team ( Id )
    );
GO -- Run the previous command and begins new batch
--3. Insert Data
INSERT  Team
VALUES  ( N'Team1' );
INSERT  Team
VALUES  ( N'Team2' );
INSERT  Team
VALUES  ( N'Team3' );
INSERT  Gamer
VALUES  ( N'Name01 ABB', N'Male', N'City01', '1979/4/28', 1 );
```

```
INSERT  Gamer
VALUES  ( N'Name02 CDDE', N'Female', N'City03', '1981/7/24', 2 );
INSERT  Gamer
VALUES  ( N'Name03 FIJK', N'Female', N'City01', '1984/12/5', 3 );
INSERT  Gamer
VALUES  ( N'Name04 LMOPPQ', N'Male', N'City02', '1983/5/29', 1 );
INSERT  Gamer
VALUES  ( N'Name05 QRSTT', N'Male', N'City01', '1979/6/20', 3 );
INSERT  Gamer
VALUES  ( N'Name06 TUVVX', N'Female', N'City03', '1984/5/15', 3 );
INSERT  Gamer
VALUES  ( N'Name07 XYZZXX', N'Female', N'City01', '1986/4/29', 2 );
INSERT  Gamer
VALUES  ( N'Name08 ABBCDE', N'Male', N'City02', '1985/7/28', 1 );
INSERT  Gamer
VALUES  ( N'Name09 QRSTTUVXX', N'Male', N'City02', '1983/4/16', 1 );
GO -- Run the previous command and begins new batch
--4. SP
CREATE PROCEDURE spGetGamers
AS
    BEGIN
        SELECT  *
        FROM    Gamer;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spAddGamer
    (
      @Name NVARCHAR(50) ,
      @Gender NVARCHAR(10) ,
      @City NVARCHAR(50) ,
      @DateOfBirth DateTime ,
      @TeamId INT
    )
AS
    BEGIN
        INSERT  INTO Gamer
        VALUES  ( @Name, @Gender, @City, @DateOfBirth, @TeamId );
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spSaveGamer
    (
      @Id INT ,
      @Name NVARCHAR(50) ,
      @Gender NVARCHAR(10) ,
      @City NVARCHAR(50) ,
      @DateOfBirth DateTime ,
      @TeamId INT
    )
AS
    BEGIN
        UPDATE  dbo.Gamer
        SET     Name = @Name ,
                Gender = @Gender ,
                City = @City ,
                DateOfBirth = @DateOfBirth ,
                TeamId = @TeamId
        WHERE   Id = @Id;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spDeleteGamer ( @Id int )
AS
    BEGIN
        DELETE  FROM Gamer
        WHERE   Id = @Id;
    END;
GO -- Run the previous command and begins new batch
--EXEC spGetGamers
--GO -- Run the previous command and begins new batch
```

1.2. Security login

In SQL server

Object Explorer --> Security --> Logins --> New Logins

-->

General Tab

Login Name :

Tester

Password:

1234

Default Database:

OnlineGame

-->

Server Roles Tab

Select

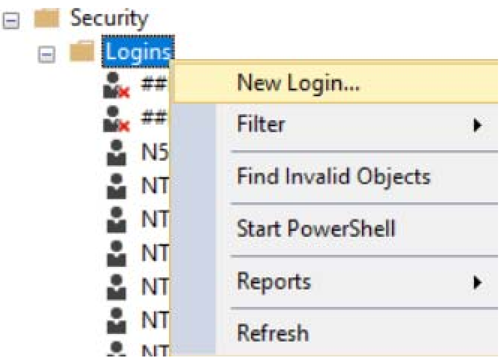
sysadmin

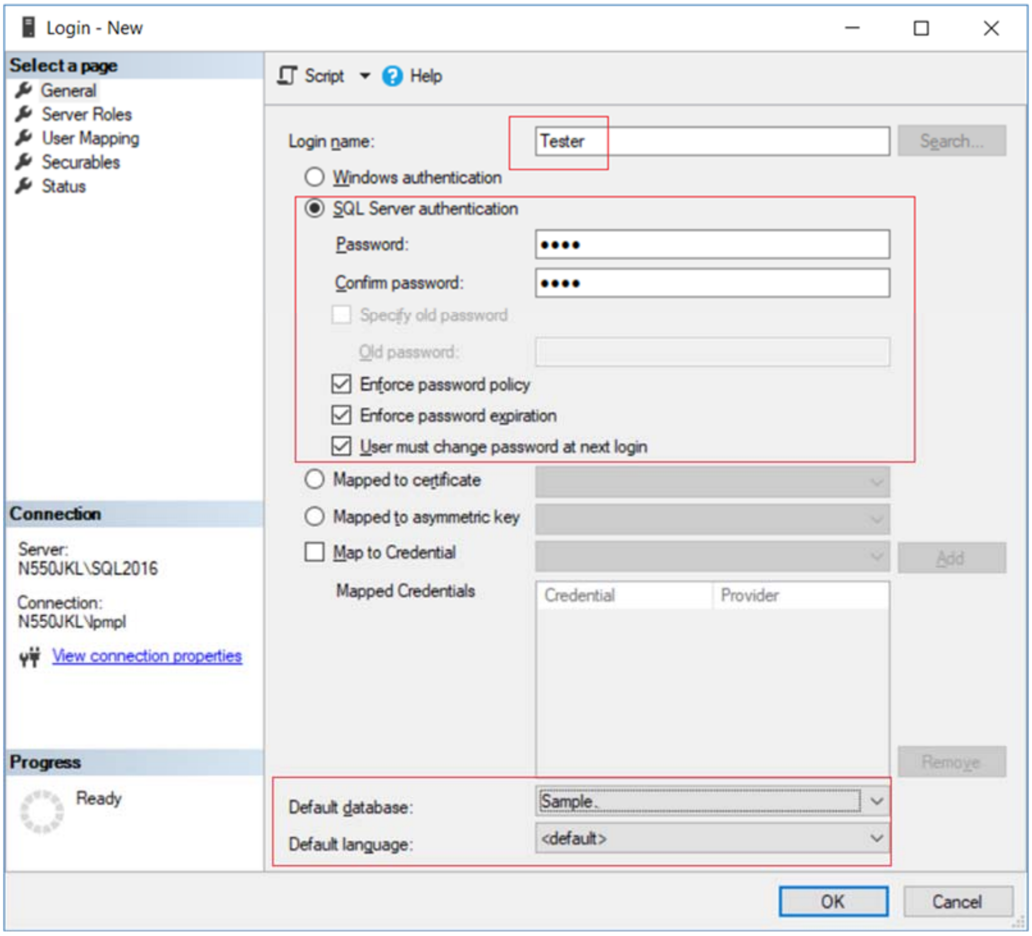
-->

User Mapping Tab

Select **OnlineGame**

Select every single role.





Login name: Tester

○ Windows authentication

● SQL Server authentication

Password: ••••

Confirm password: ••••

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Mapped Credentials

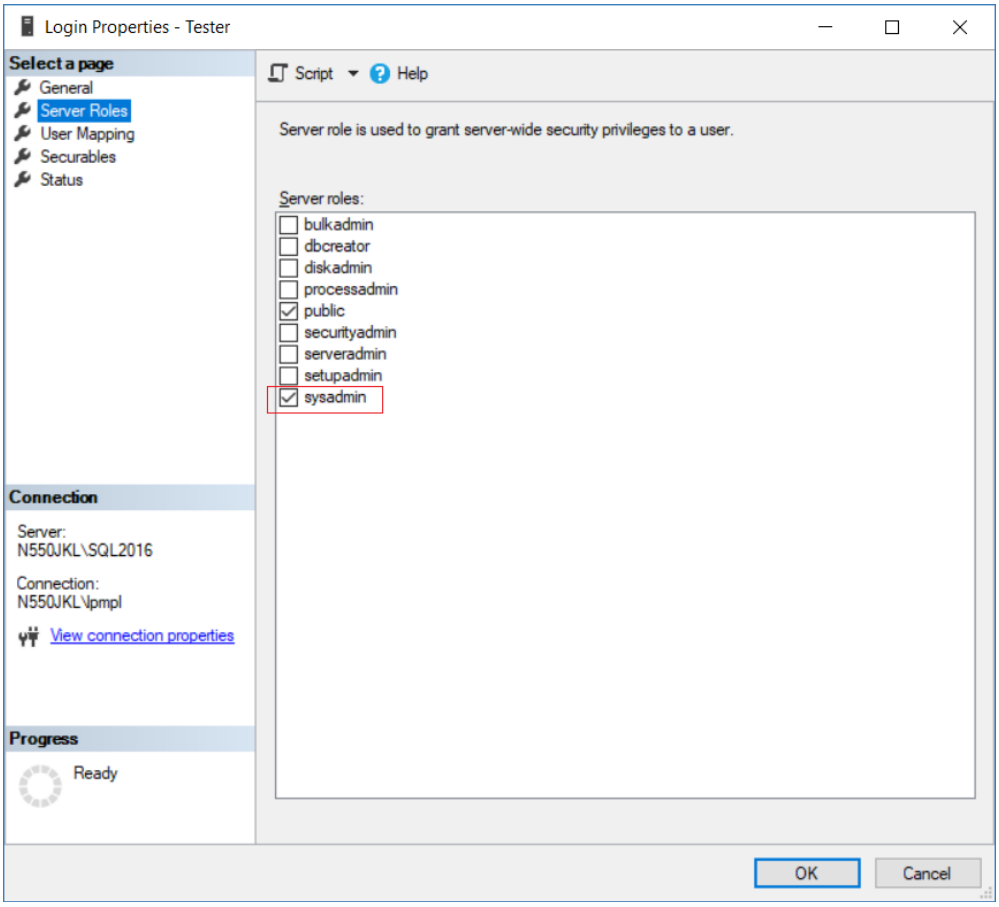
Credential	Provider

Default database: Sample

Default language: <default>

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl



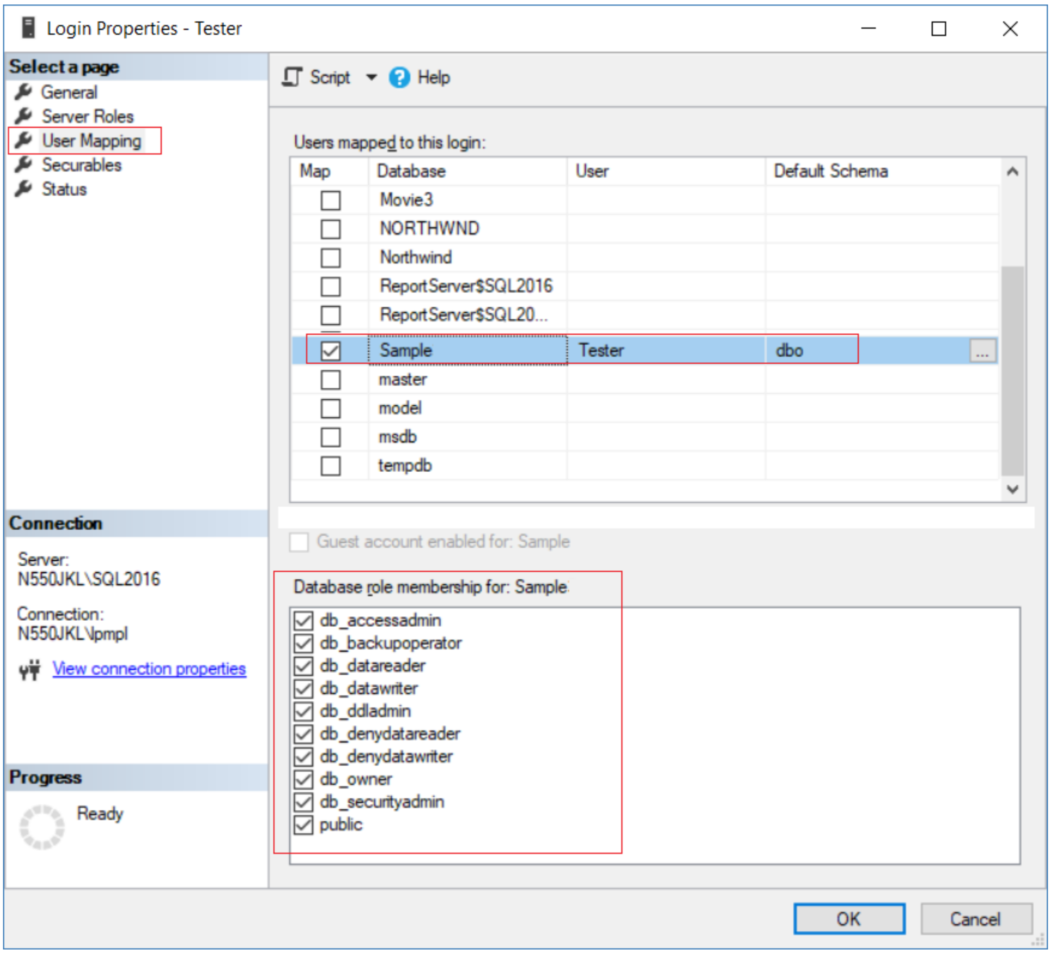
Server role is used to grant server-wide security privileges to a user.

Server roles:

- □ bulkadmin
- □ dbcreator
- □ diskadmin
- □ processadmin
- ☑ public
- □ securityadmin
- □ serveradmin
- □ setupadmin
- ☑ sysadmin

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl


Login Properties - Tester

Users mapped to this login:

Map	Database	User	Default Schema
□	Movie3		
□	NORTHWND		
□	Northwind		
□	ReportServer$SQL2016		
□	ReportServer$SQL20...		
☑	Sample	Tester	dbo
□	master		
□	model		
□	msdb		
□	tempdb		

□ Guest account enabled for: Sample

Database role membership for: Sample

- ☑ db_accessadmin
- ☑ db_backupoperator
- ☑ db_datareader
- ☑ db_datawriter
- ☑ db_ddladmin
- ☑ db_denydatareader
- ☑ db_denydatawriter
- ☑ db_owner
- ☑ db_securityadmin
- ☑ public

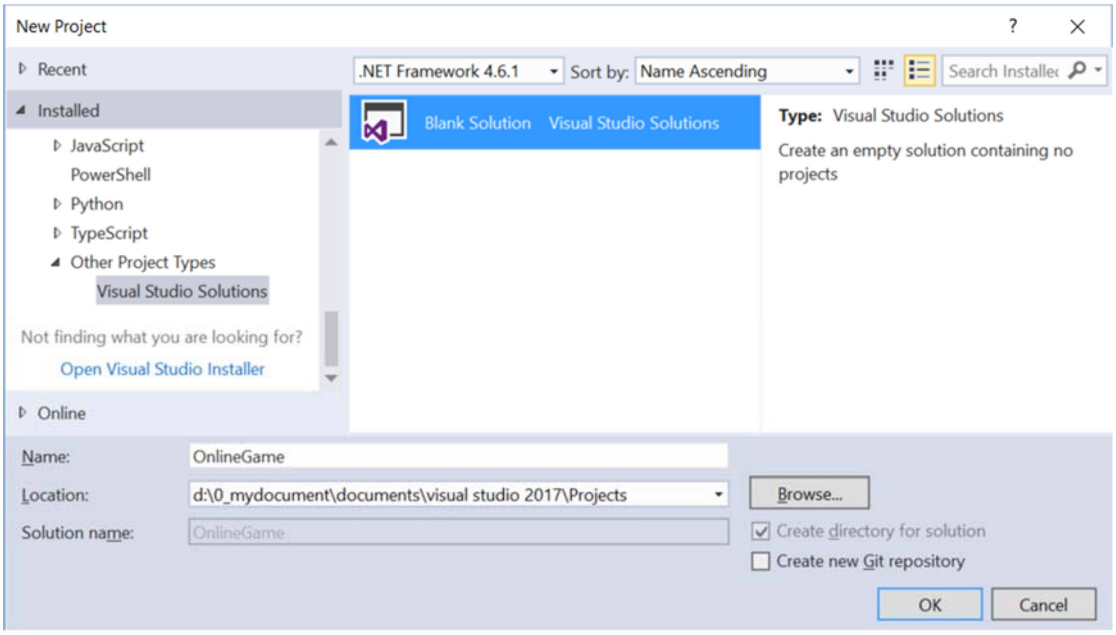
Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl

2. New Project - OnlineGame

File --> New --> Project... -->
Other Project Types --> Visual Studio Solutions --> Blank Solution
-->
Name: **OnlineGame**


New Project

Type: Visual Studio Solutions

Create an empty solution containing no projects

Name: OnlineGame

Location: d:\0_mydocument\documents\visual studio 2017\Projects

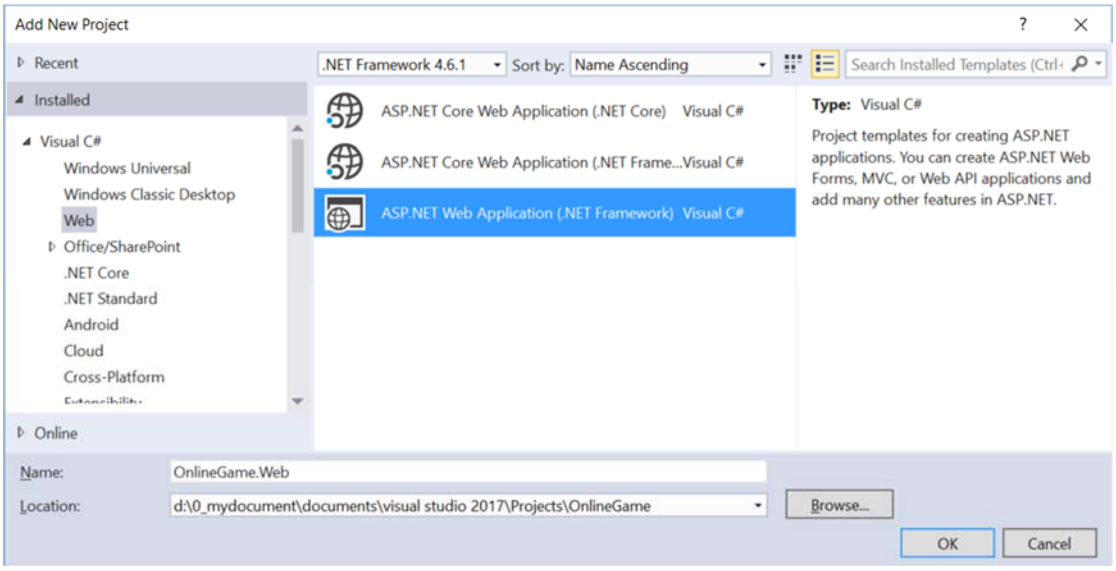
Solution name: OnlineGame

☑ Create directory for solution

□ Create new Git repository

2.1. New Project - OnlineGame.Web

Solutions Name --> Add --> New Project -->
Visual C# --> Web --> ASP.NET Web Application (.Net Framework)
-->
Name: **OnlineGame.Web**
Empty --> Select "MVC" --> OK



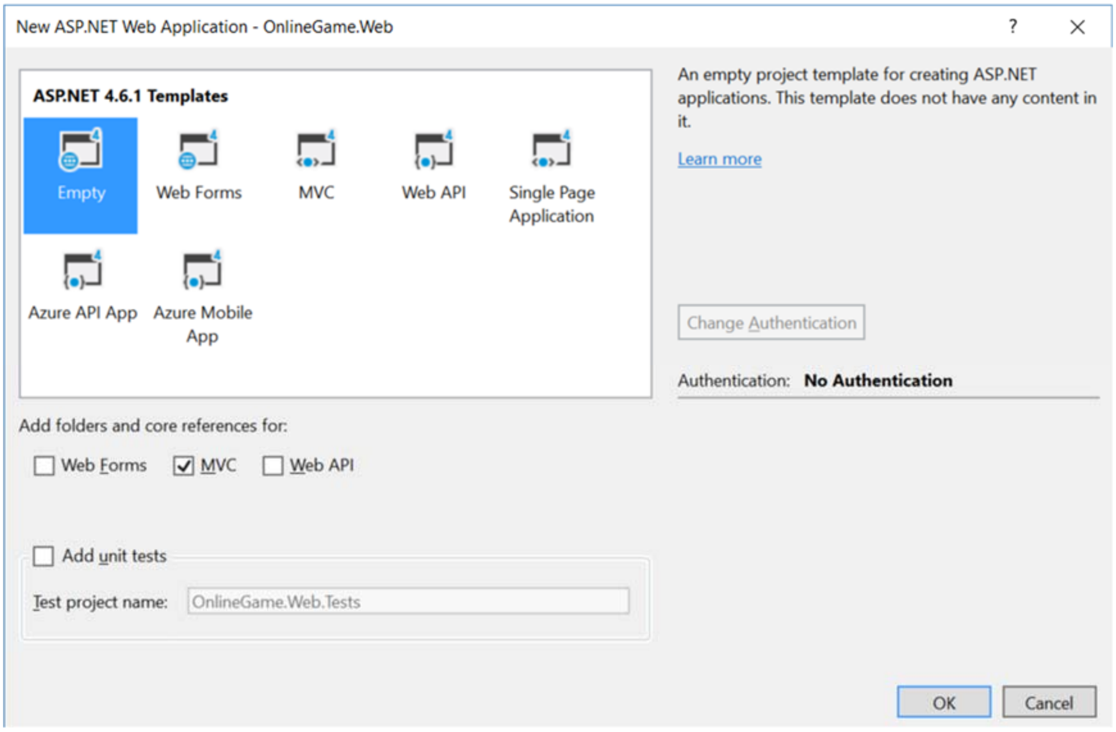
.NET Framework 4.6.1
Sort by: Name Ascending
Search Installed Templates (Ctrl+

ASP.NET Core Web Application (.NET Core) Visual C#
ASP.NET Core Web Application (.NET Frame... Visual C#
ASP.NET Web Application (.NET Framework) Visual C#

Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: OnlineGame.Web
Location: d:\0_mydocument\documents\visual studio 2017\Projects\OnlineGame



ASP.NET 4.6.1 Templates

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication: **No Authentication**

Add folders and core references for:

☐ Web Forms ☑ MVC ☐ Web API

☐ Add unit tests

Test project name: OnlineGame.Web.Tests

2.1.1. Global.asax.cs

```
using System.Web.Mvc;
using System.Web.Routing;
namespace OnlineGame.Web
```

```
{
    public class MvcApplication : System.Web.HttpApplication
    {
        //Application_Start() is the magic start point of this application
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            //1.
            //Register Route Configure in RouteConfig.cs
            //If you want to see route configuration,
            //you may find it in RouteConfig.cs
            //2.
            //System.Web.Routing.RouteCollection Routes { get; }
            //Gets a collection of objects that derive from the System.Web.Routing.RouteBase class.
            RouteConfig.RegisterRoutes(RouteTable.Routes);
        }
    }
}
```

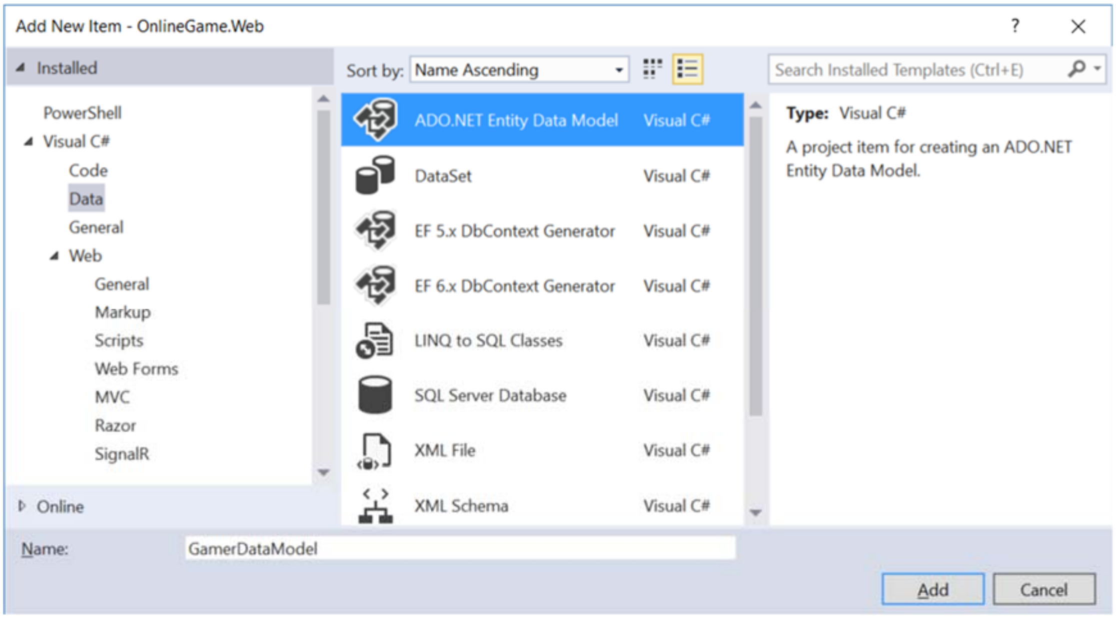
2.1.2. App_Start/RouteConfig.cs

```
using System.Web.Mvc;
using System.Web.Routing;
namespace OnlineGame.Web
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            //Handle the Route of the axd request file.
            //E.g. ASP.Net Tracing
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
            //Handle the Route called "Default".
            //The mapping URL is "{controller}/{action}/{id}"
            //Set the default value of Controller, action, and id.
            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Gamers", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
/*
1.
//routes.MapRoute(
//    name: "Default",
//    url: "{controller}/{action}/{id}",
//    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
//);
1.1.
When a request comes in,
it's trying to do a pattern match based on
all the templates it sees in these mapped routes.
A route is some instructions for
how to take a URI coming into a request
and map it to some code,
normally a controller.
In this case,
look at defaults parameter,
when user request http://localhost:PortNumber/
IIS Express will run
HomeController Index action.
It will map to Controllers/HomeController.cs
and   map to Index Method
1.2.
By convention in MVC.
All controllers will have Controller suffix.
This suffix is not required in the URL.
So, if you want to invoke Home controller,
you specify /Home and not /HomeController.
-----------------------------------
2.
//routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
2.1.
Reference:
https://stackoverflow.com/questions/9016650/what-is-routes-ignorerouteresource-axd-pathinfo
This line can handle the axd file request route,
E.g. trace.axd
.axd files don't exist physically.
ASP.NET uses URLs with .axd extensions
(ScriptResource.axd and WebResource.axd) internally,
and they are handled by an HttpHandler.
Therefore, you should keep this rule,
to prevent ASP.NET MVC from trying to handle the request
instead of letting the dedicated HttpHandler do it.
2.2.
trace.axd
Reference:
https://msdn.microsoft.com/en-us/library/wwh16c6c.aspx
trace.axd trace details for a specific request.
If you want to enable trace.axd,
then you have to go to Web.config
Add <trace enabled="true" pageOutput="false"/> under <system.web>
Then run the project, type the following URL
http://localhost/OnlineGame.Web/trace.axd
This will return ASP.NET trace, trace.axd.
If you do not have
// routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
then you can not enable the trace.axd.
*/
```

2.2. ADO.Net Entity Data Model - Entity Framework

In Visual Studio 2017
Models folder --> Right Click --> Add --> New Item
--> Visual C# --> Data --> ADO.Net Entity Data Model
Name:
GamerDataModel

-->
EF Designer from database

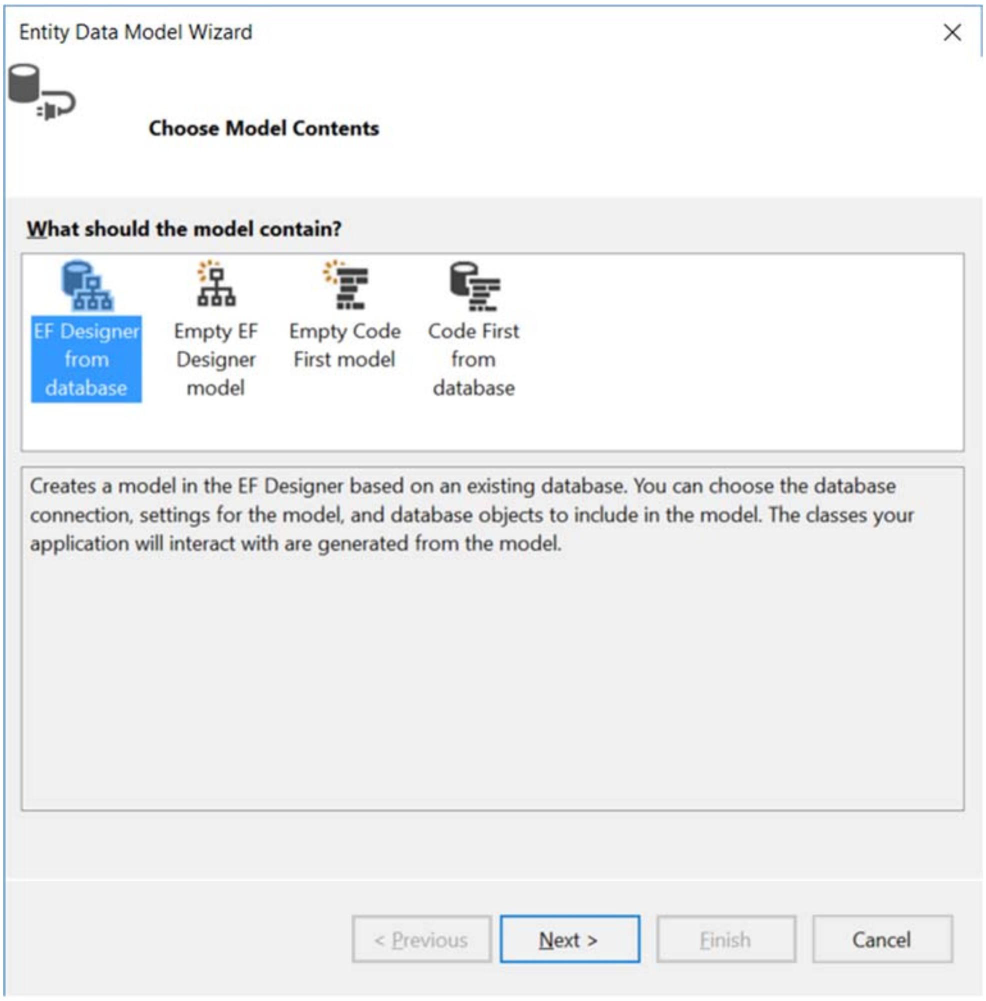

Add New Item - OnlineGame.Web
?
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Online
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
SQL Server Database Visual C#
XML File Visual C#
XML Schema Visual C#
Search Installed Templates (Ctrl+E)
Add
Cancel

Sort by: Name Ascending

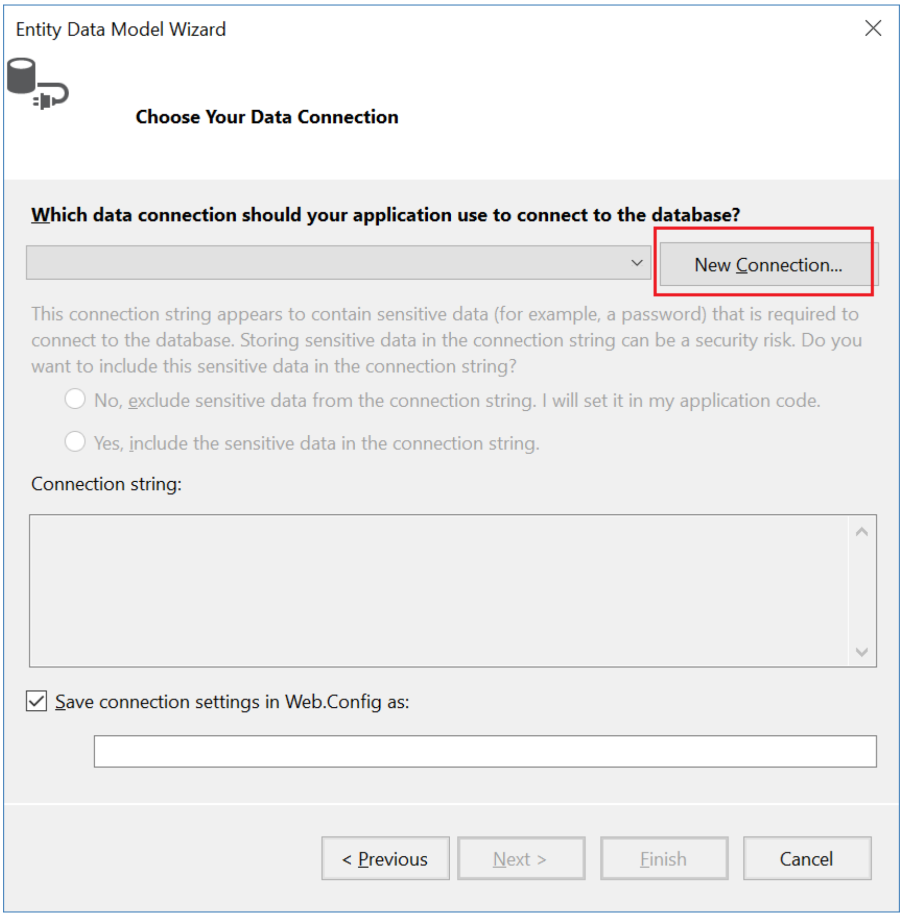
Type: Visual C#

A project item for creating an ADO.NET Entity Data Model.

Name: GamerDataModel



Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.



Choose Your Data Connection

Which data connection should your application use to connect to the database?

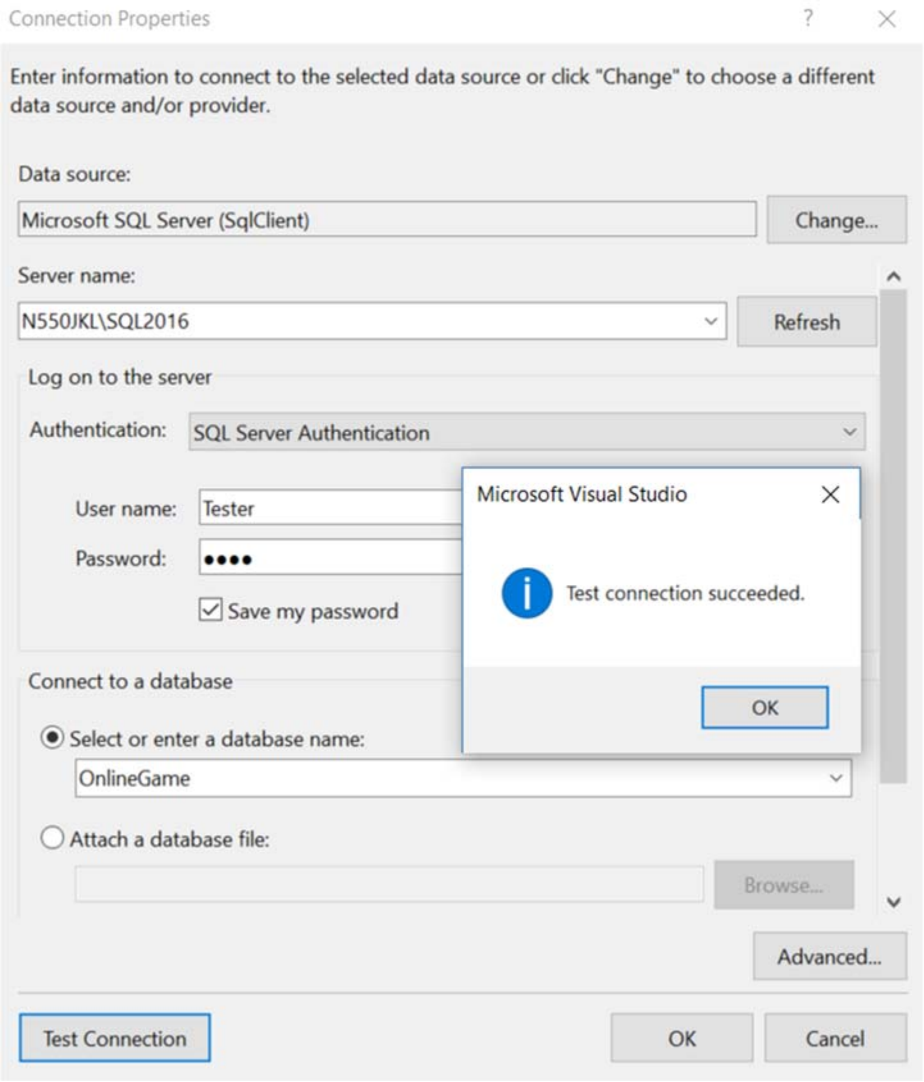
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

Connection string:

☑ Save connection settings in Web.Config as:



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: N550JKL\SQL2016

Log on to the server

Authentication: SQL Server Authentication

User name: Tester

Password: ●●●●

☑ Save my password

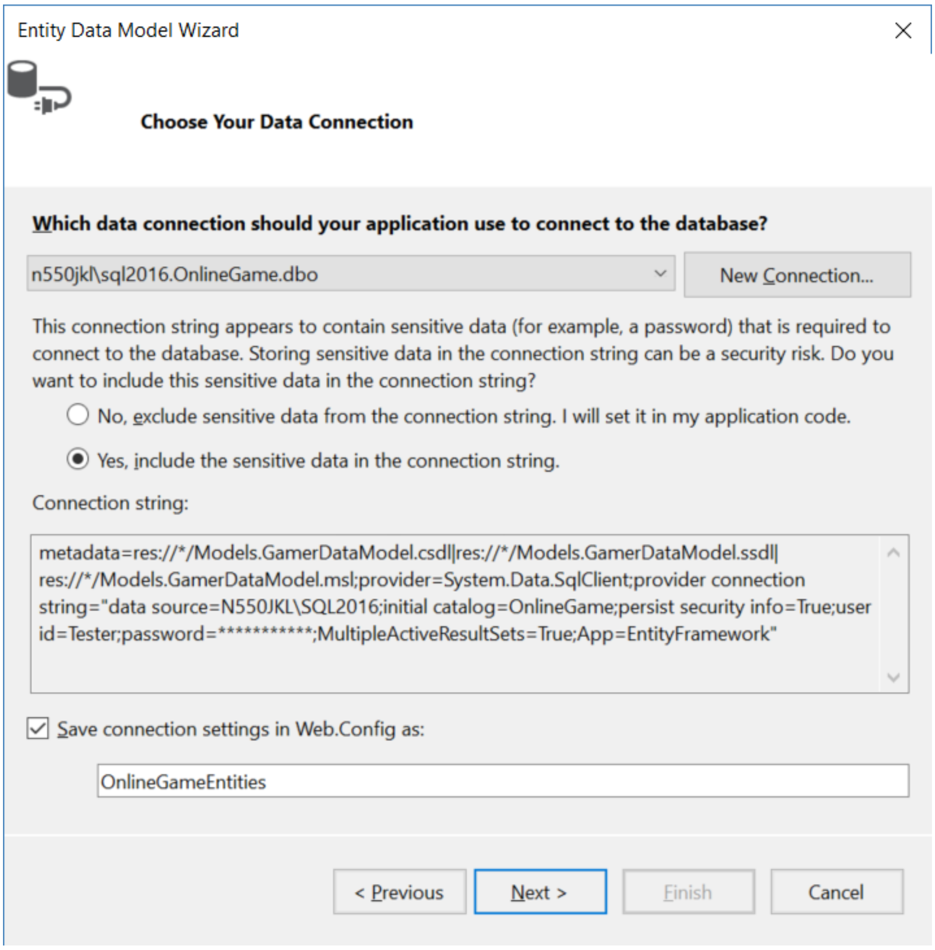
Microsoft Visual Studio

Test connection succeeded.

Connect to a database

● Select or enter a database name: OnlineGame

○ Attach a database file:



Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

n550jkl\sql2016.OnlineGame.dbo

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

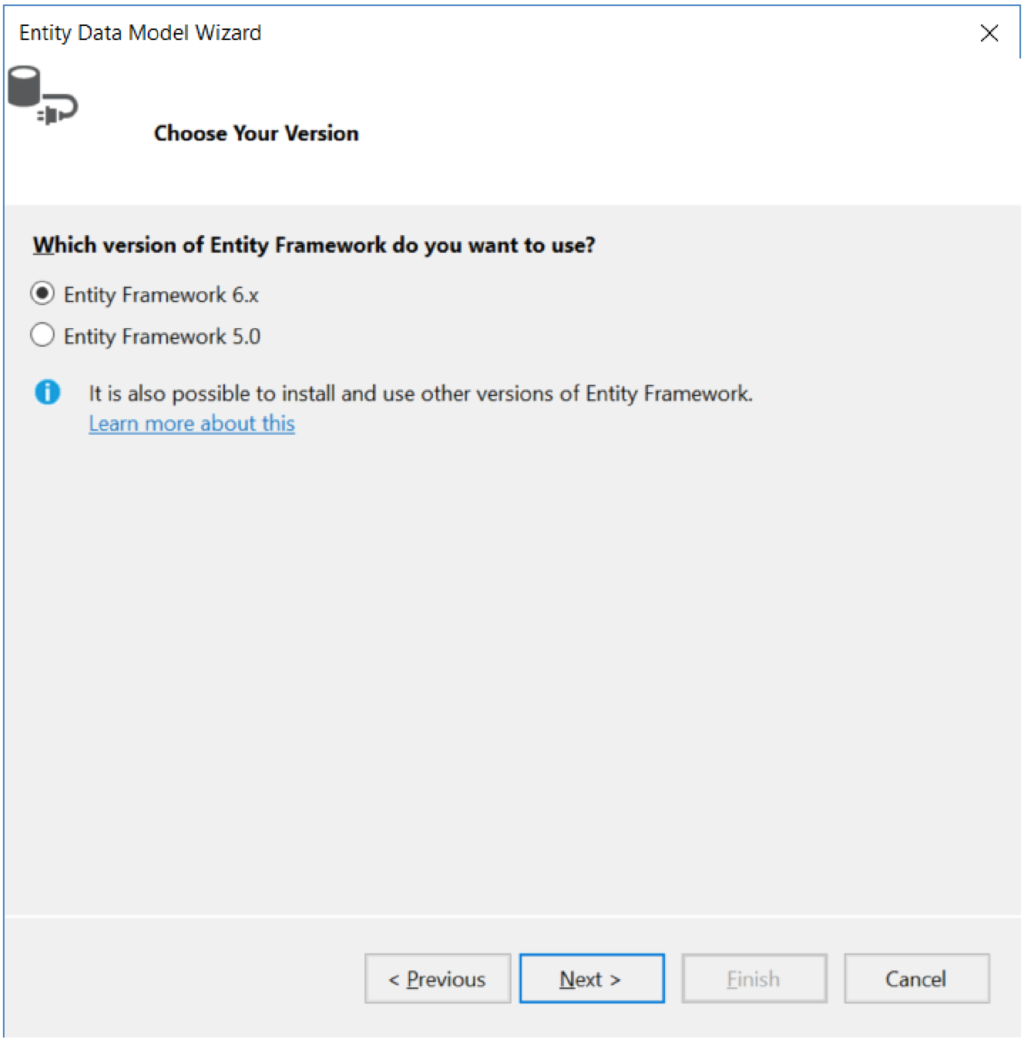
● Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/Models.GamerDataModel.csdl|res://*/Models.GamerDataModel.ssdl|
res://*/Models.GamerDataModel.msl;provider=System.Data.SqlClient;provider connection
string="data source=N550JKL\SQL2016;initial catalog=OnlineGame;persist security info=True;user
id=Tester;password=***********;MultipleActiveResultSets=True;App=EntityFramework"
```

☑ Save connection settings in Web.Config as:

OnlineGameEntities



Entity Data Model Wizard

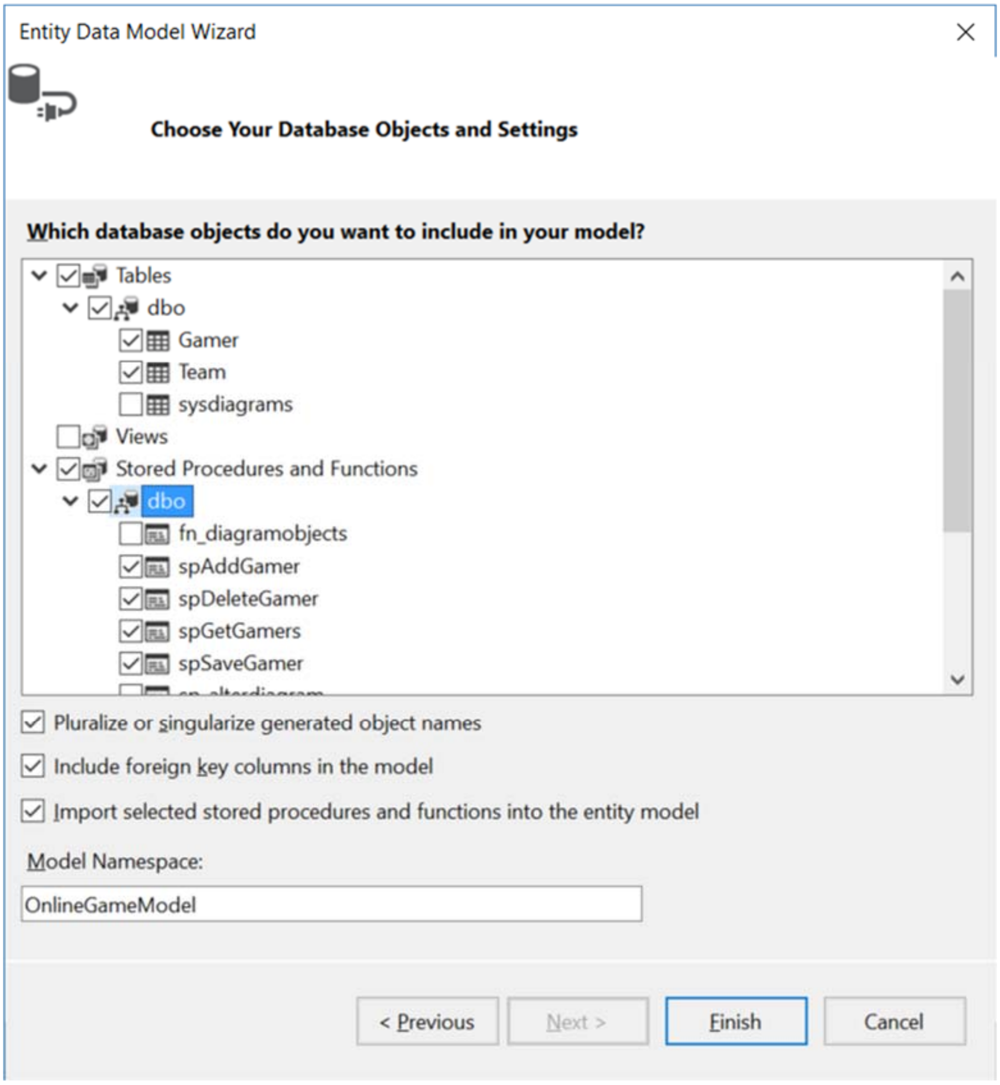
Choose Your Version

Which version of Entity Framework do you want to use?

● Entity Framework 6.x

○ Entity Framework 5.0

It is also possible to install and use other versions of Entity Framework.
Learn more about this



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

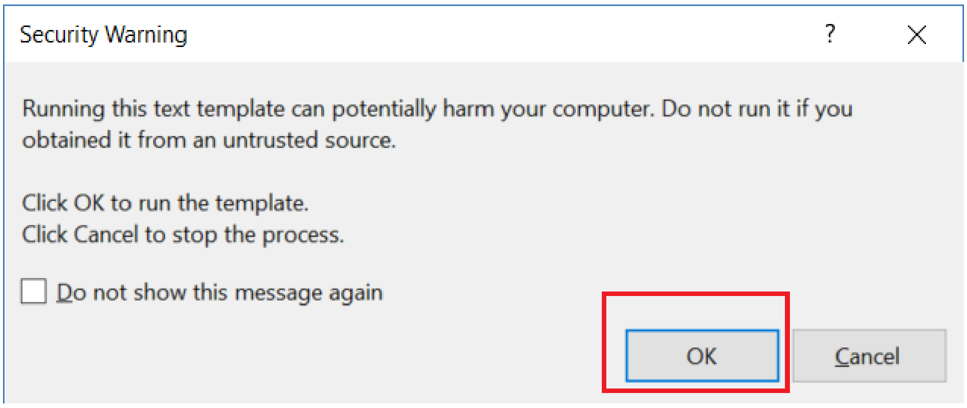
- ☑ Tables
  - ☑ dbo
    - ☑ Gamer
    - ☑ Team
    - □ sysdiagrams
- □ Views
- ☑ Stored Procedures and Functions
  - ☑ dbo
    - □ fn_diagramobjects
    - ☑ spAddGamer
    - ☑ spDeleteGamer
    - ☑ spGetGamers
    - ☑ spSaveGamer
    - □ sp_alterdiagram

☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

☑ Import selected stored procedures and functions into the entity model

Model Namespace: OnlineGameModel

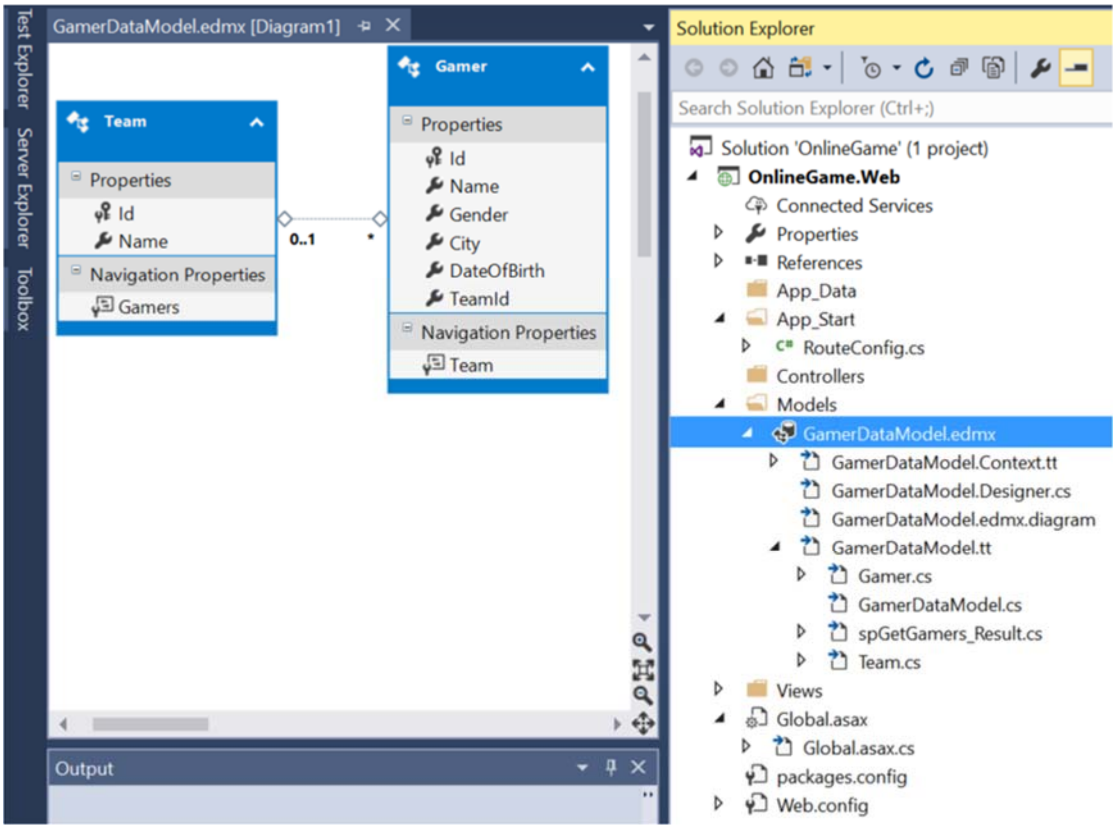


Security Warning

Running this text template can potentially harm your computer. Do not run it if you obtained it from an untrusted source.

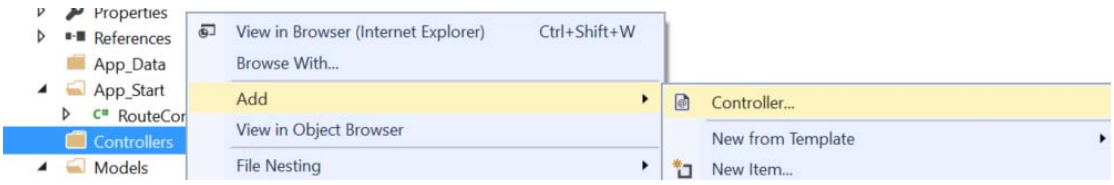
Click OK to run the template.
Click Cancel to stop the process.

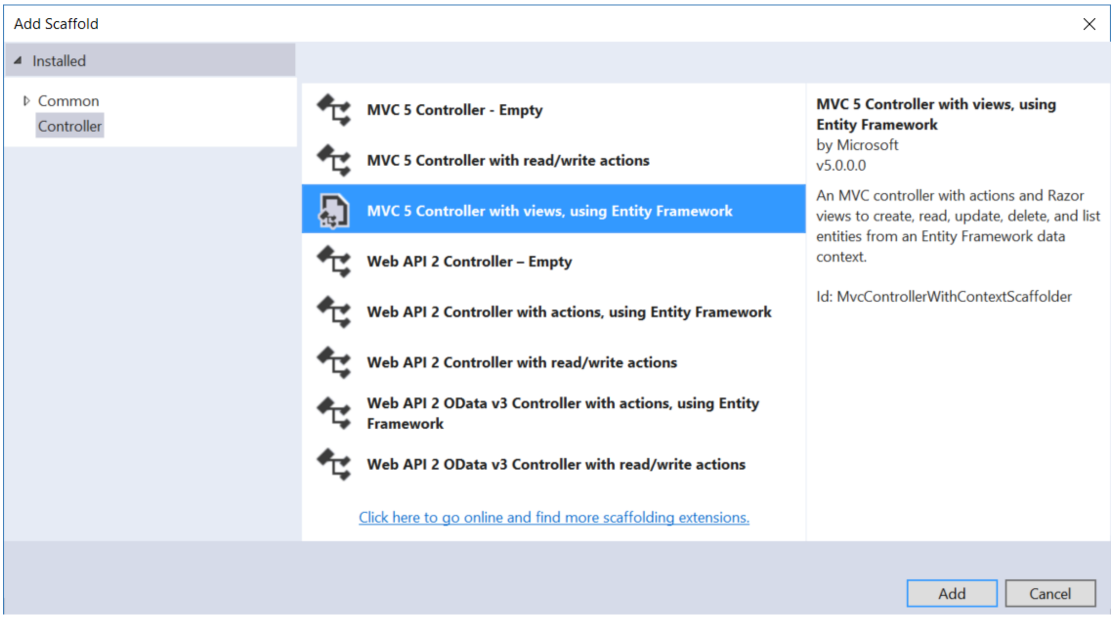
□ Do not show this message again



2.3. Controllers/GamersController.cs

Controllers --> Right click --> Add --> Controller
-->
MVC 5 Controller with views, using Entity Framework





Add Scaffold

- Installed
  - Common
  - Controller

- MVC 5 Controller - Empty
- MVC 5 Controller with read/write actions
- MVC 5 Controller with views, using Entity Framework
- Web API 2 Controller – Empty
- Web API 2 Controller with actions, using Entity Framework
- Web API 2 Controller with read/write actions
- Web API 2 OData v3 Controller with actions, using Entity Framework
- Web API 2 OData v3 Controller with read/write actions

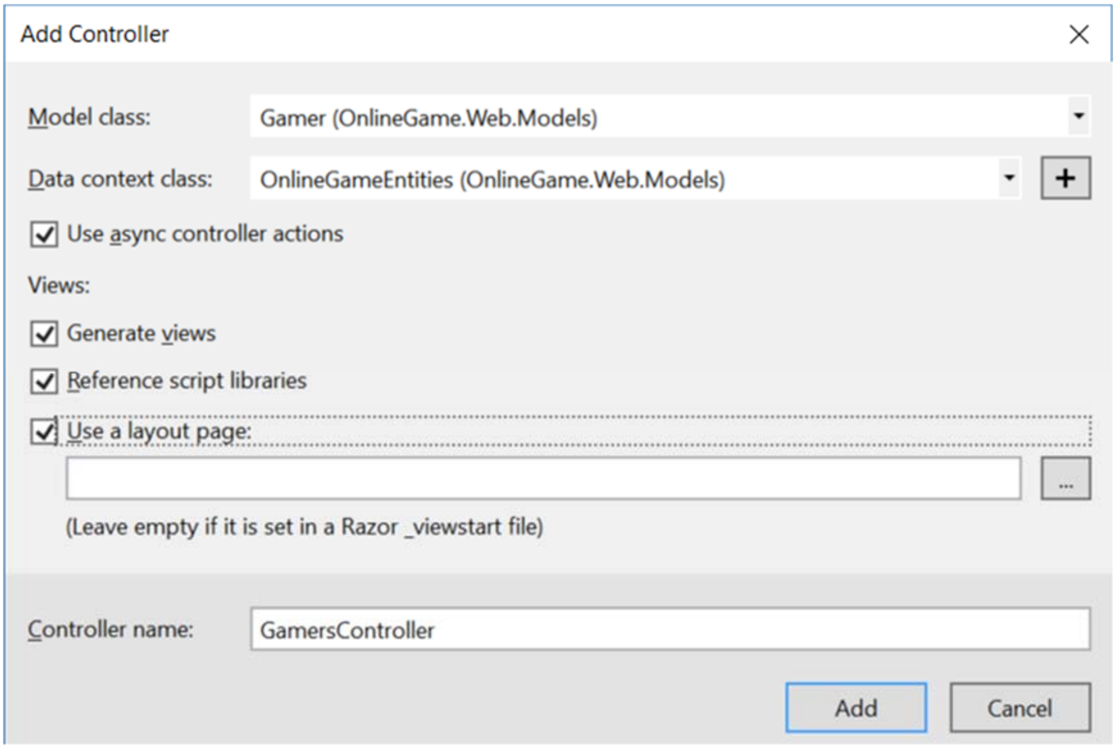
Click here to go online and find more scaffolding extensions.

MVC 5 Controller with views, using Entity Framework
by Microsoft
v5.0.0.0

An MVC controller with actions and Razor views to create, read, update, delete, and list entities from an Entity Framework data context.

Id: MvcControllerWithContextScaffolder

Add Cancel



Add Controller

Model class: Gamer (OnlineGame.Web.Models)

Data context class: OnlineGameEntities (OnlineGame.Web.Models)

☑ Use async controller actions

Views:

☑ Generate views

☑ Reference script libraries

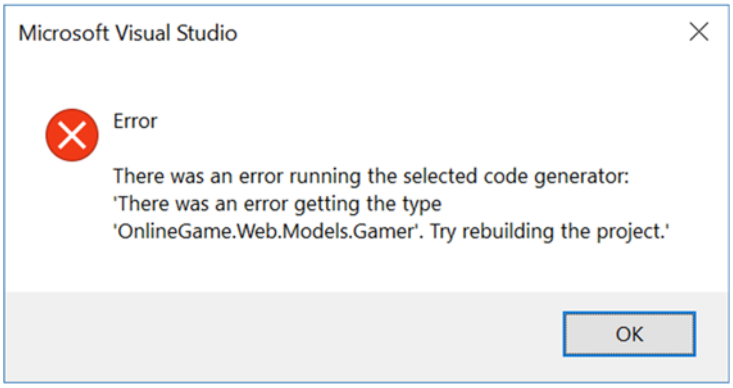
☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Controller name: GamersController

Add Cancel

If you see the following error message, then you have to re-build solution before you create the controller.

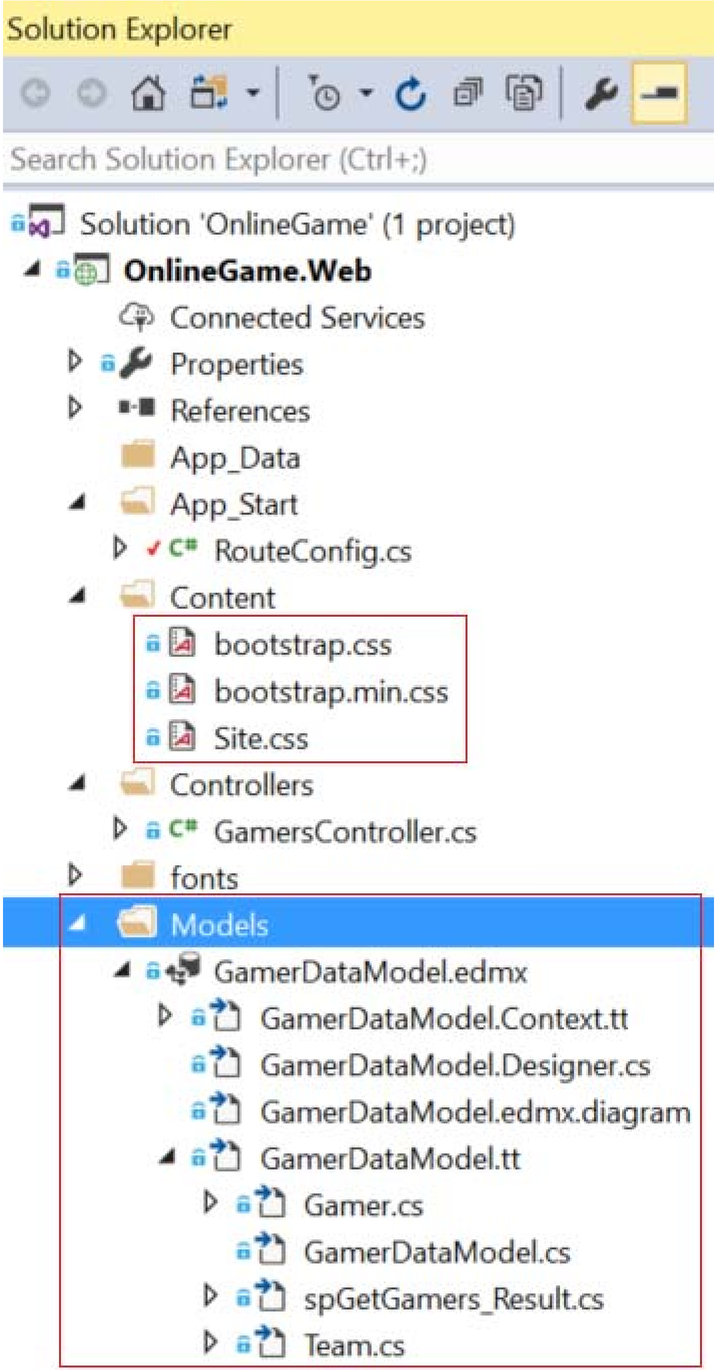


Microsoft Visual Studio

Error

There was an error running the selected code generator: 'There was an error getting the type 'OnlineGame.Web.Models.Gamer'. Try rebuilding the project.'

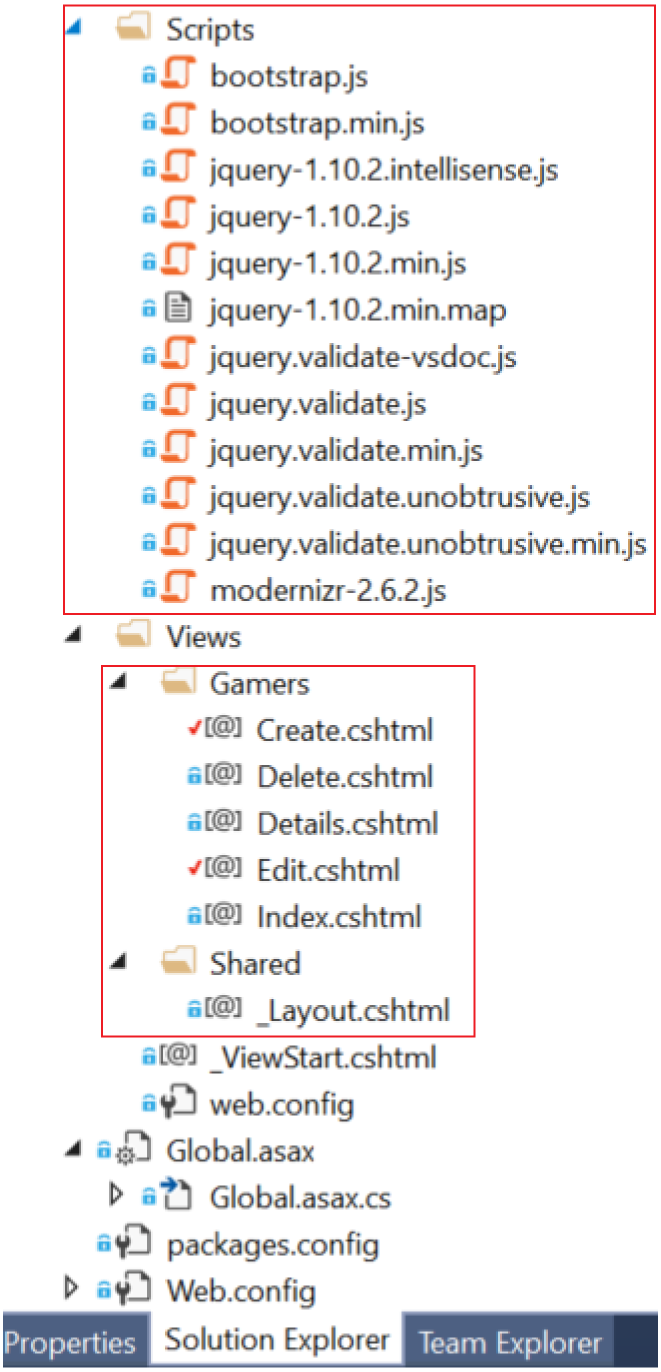
It will automatically generate the controller, views, and several javascript and css files.



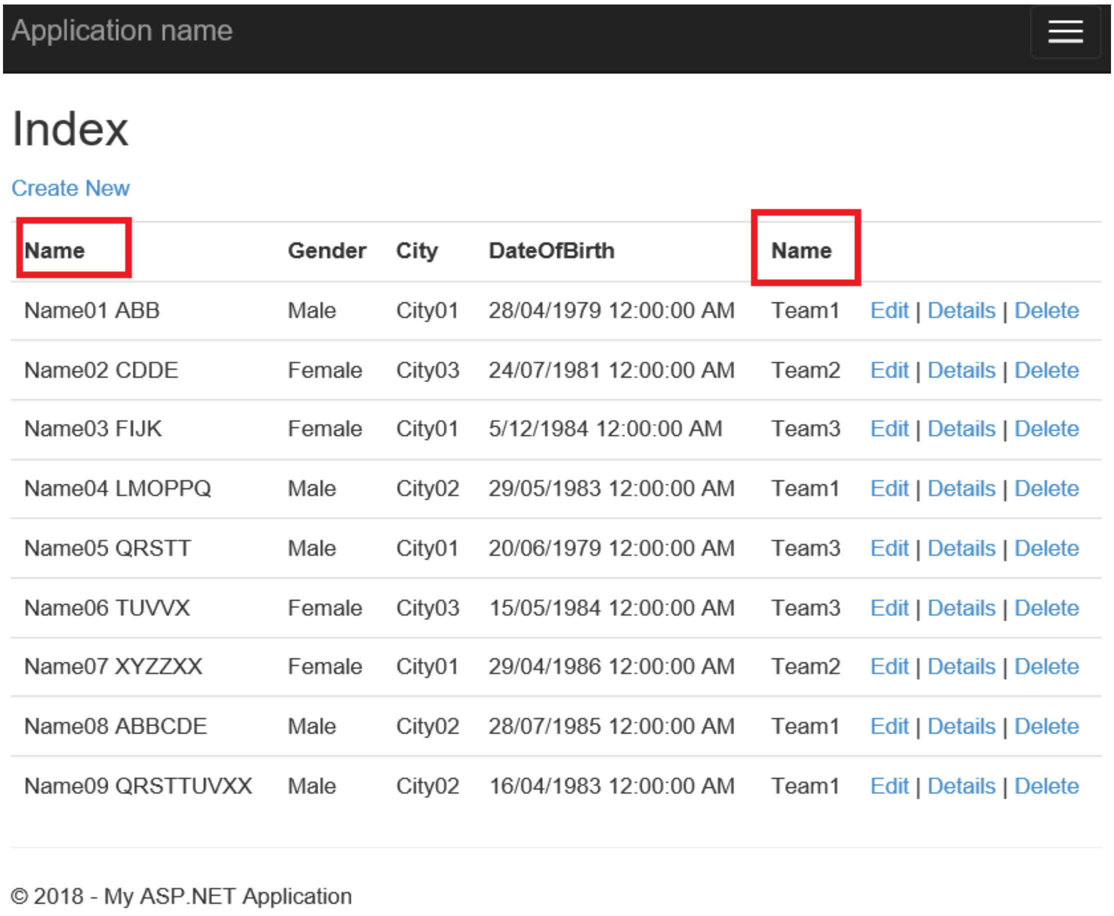
Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'OnlineGame' (1 project)
  OnlineGame.Web
    Connected Services
    Properties
    References
    App_Data
    App_Start
      RouteConfig.cs
    Content
      bootstrap.css
      bootstrap.min.css
      Site.css
    Controllers
      GamersController.cs
    fonts
    Models
      GamerDataModel.edmx
        GamerDataModel.Context.tt
        GamerDataModel.Designer.cs
        GamerDataModel.edmx.diagram
        GamerDataModel.tt
          Gamer.cs
          GamerDataModel.cs
          spGetGamers_Result.cs
          Team.cs
```



```
◢ Scripts
    bootstrap.js
    bootstrap.min.js
    jquery-1.10.2.intellisense.js
    jquery-1.10.2.js
    jquery-1.10.2.min.js
    jquery-1.10.2.min.map
    jquery.validate-vsdoc.js
    jquery.validate.js
    jquery.validate.min.js
    jquery.validate.unobtrusive.js
    jquery.validate.unobtrusive.min.js
    modernizr-2.6.2.js
◢ Views
  ◢ Gamers
      ✓ Create.cshtml
        Delete.cshtml
        Details.cshtml
      ✓ Edit.cshtml
        Index.cshtml
  ◢ Shared
        _Layout.cshtml
    _ViewStart.cshtml
    web.config
◢ Global.asax
  ▷ Global.asax.cs
  packages.config
▷ Web.config
```



Index

Create New

Name	Gender	City	DateOfBirth	Name	
Name01 ABB	Male	City01	28/04/1979 12:00:00 AM	Team1	Edit | Details | Delete
Name02 CDDE	Female	City03	24/07/1981 12:00:00 AM	Team2	Edit | Details | Delete
Name03 FIJK	Female	City01	5/12/1984 12:00:00 AM	Team3	Edit | Details | Delete
Name04 LMOPPQ	Male	City02	29/05/1983 12:00:00 AM	Team1	Edit | Details | Delete
Name05 QRSTT	Male	City01	20/06/1979 12:00:00 AM	Team3	Edit | Details | Delete
Name06 TUVVX	Female	City03	15/05/1984 12:00:00 AM	Team3	Edit | Details | Delete
Name07 XYZZXX	Female	City01	29/04/1986 12:00:00 AM	Team2	Edit | Details | Delete
Name08 ABBCDE	Male	City02	28/07/1985 12:00:00 AM	Team1	Edit | Details | Delete
Name09 QRSTTUVXX	Male	City02	16/04/1983 12:00:00 AM	Team1	Edit | Details | Delete

© 2018 - My ASP.NET Application

3. OnlineGame.Web

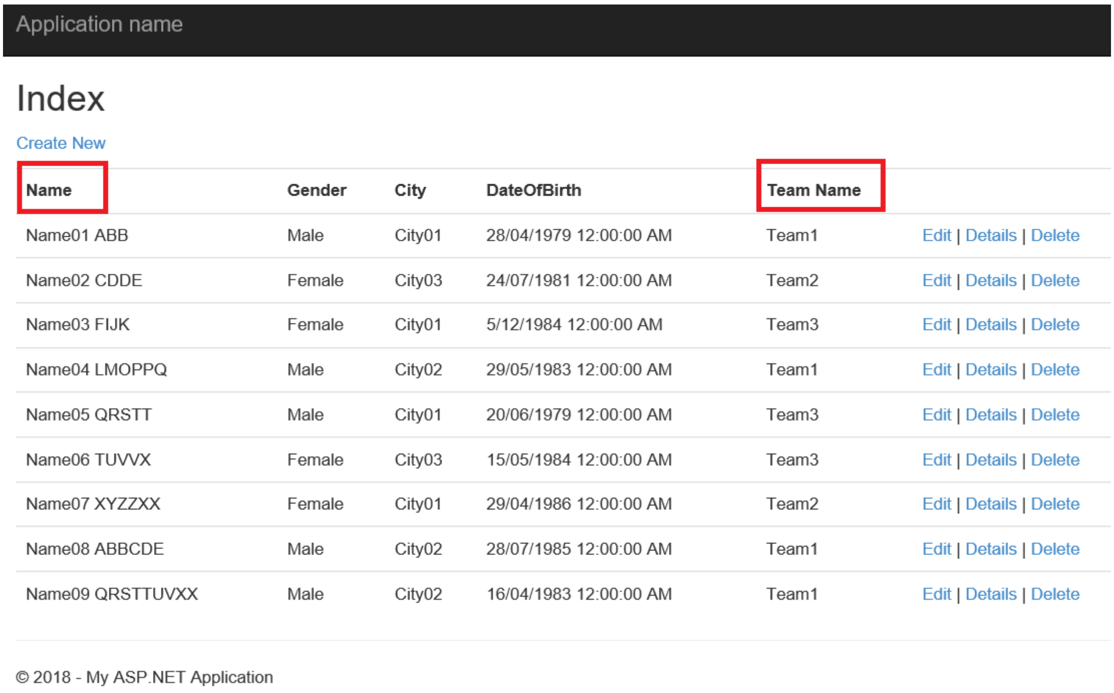
3.1. Models/Teams/Team.cs

```
using System.ComponentModel.DataAnnotations;
namespace OnlineGame.Web.Models
{
    [MetadataType(typeof(TeamMetaData))]
    public partial class Team
    {
        //[Display(Name = "Team Name")]
        //public string Name { get; set; }
        //// Error!!
        //// Memeber with the same name is aready declared in other auto-generated partial class.
        //// Thus, you need MetadataType to add extra code for the Property.
        //// E.g. tadataType(typeof(TeamMetaData))]
        //// In this case, you may add some extra code for the Property in MetadataType class
    }
}
```

3.2. Models/Teams/TeamMetaData.cs

```
using System.ComponentModel.DataAnnotations;
namespace OnlineGame.Web.Models
```

```
{
    public class TeamMetaData
     {
         [Display(Name = "Team Name")]
        public string Name { get; set; }
        // Here is the place you may add some extra code for the property
        // which is already in the auto-generate partail class.
     }
}
```



Application name

Index

Create New

Name	Gender	City	DateOfBirth	Team Name	
Name01 ABB	Male	City01	28/04/1979 12:00:00 AM	Team1	Edit | Details | Delete
Name02 CDDE	Female	City03	24/07/1981 12:00:00 AM	Team2	Edit | Details | Delete
Name03 FIJK	Female	City01	5/12/1984 12:00:00 AM	Team3	Edit | Details | Delete
Name04 LMOPPQ	Male	City02	29/05/1983 12:00:00 AM	Team1	Edit | Details | Delete
Name05 QRSTT	Male	City01	20/06/1979 12:00:00 AM	Team3	Edit | Details | Delete
Name06 TUVVX	Female	City03	15/05/1984 12:00:00 AM	Team3	Edit | Details | Delete
Name07 XYZZXX	Female	City01	29/04/1986 12:00:00 AM	Team2	Edit | Details | Delete
Name08 ABBCDE	Male	City02	28/07/1985 12:00:00 AM	Team1	Edit | Details | Delete
Name09 QRSTTUVXX	Male	City02	16/04/1983 12:00:00 AM	Team1	Edit | Details | Delete

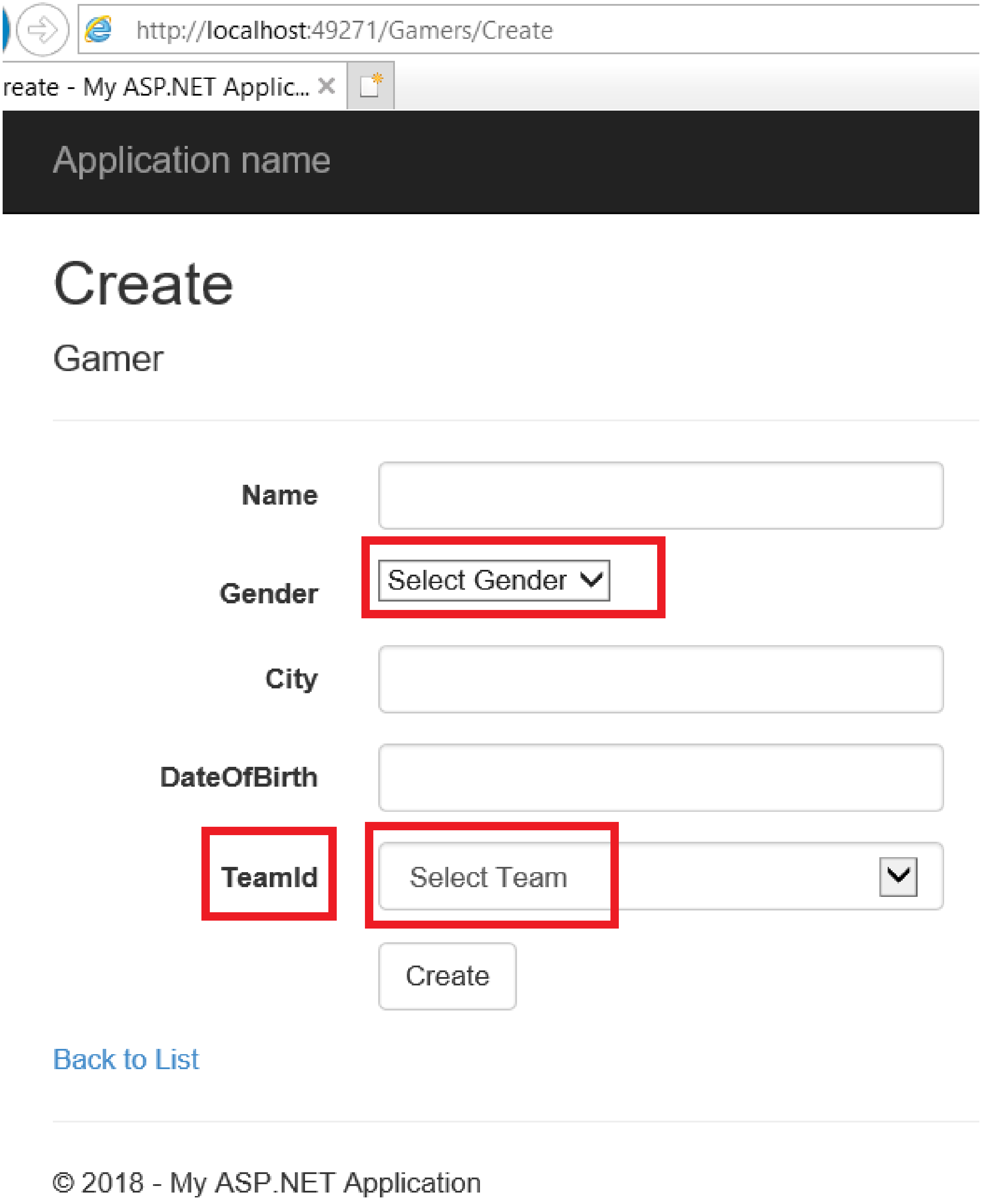
© 2018 - My ASP.NET Application

3.3. Views/Gamers/Create.cshtml

```
@model OnlineGame.Web.Models.Gamer
@{
    ViewBag.Title = "Create";
}
<h2>Create</h2>
@*@using (Html.BeginForm("Create", "Gamer"))*@
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Gamer</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Gender, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @*@Html.EditorFor(model => model.Gender, new { htmlAttributes = new { @class = "form-
control" } })*@
                @*@Html.DropDownList("Gender",new List<SelectListItem>
                    {
                        new SelectListItem{Text = "Male", Value = "Male"},
                        new SelectListItem{Text = "Female", Value = "Female"}
                    })*@
                @Html.DropDownList("Gender", new List<SelectListItem>
                    {
                        new SelectListItem{Text = "Male", Value = "Male"},
                        new SelectListItem{Text = "Female", Value = "Female"}
                    }, "Select Gender")
                @Html.ValidationMessageFor(model => model.Gender, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.City, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.TeamId, "TeamId", new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("TeamId", null, "Select Team", new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.TeamId, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

```
@*
1.
1.1.
Select <form>....</form>
We are using Web Essentials extension
Select part of HTML text, then press Shift + Alt + W
it will surround those pieces of text with <div>
1.2.
// <div class="row">
class="row" in bootstrap means this is the container for the bootstrap 12 columns grid system
1.3.
// <div class="col-md-6">
1.3.1.
.col-lg-XX means the screen size is >= 1200px, lg means large
.col-md-XX means the screen size is >= 992px, means 992px to 1199px, md means medium
.col-sm-XX means the screen size is >= 768px, means 768px to 991px, sm means small
.col-xs-XX means the screen size is < 768px, xs means extra small
1.3.2.
// <div class="col-md-6">
The md in   class="col-md-6"  means the screen size is 992px to 1199px
12 columns grid system divide this screen to 12 columns,
The 6 in   class="col-md-6" means this div occupy 6 columns out of 12 columns grid system.
Therefore,
// <div class="col-md-6">
//      ...
// </div>
// <div class="col-md-6">
//      <h2>The Map</h2>
// </div>
It will become left half and right half when it is full screen.
However, when reducing  screen size less than 11 columns,
then it will become top and bottom.
1.3.3.
If we do this
// <div class="col-md-6 col-xs-8">
//      ...
// </div>
// <div class="col-md-6 col-xs-4">
//      <h2>The Map</h2>
// </div>
The  .col-xs-XX   will override  .col-md-XX
However, we actually only need big one, .col-md-XX
because for small screen device like phone,
we want it automatically align vertically.
so let's only use   .col-md-XX   and delete   .col-xs-XX
2.
// <div class="form-group">
//    <label>Date</label>
//    <input class="form-control"/>
// </div>
// ...
// <input type="submit" value="Add" class="btn btn-success"/>
2.1.
.form-group in bootstrap normally used for group label and input.
2.2.
.form-control in bootstrap will occupy the 100% width
2.3.
.btn-success in bootstrap will become green btn
2.4.
btn-danger in bootstrap will become red btn
2.5.
btn-warning in bootstrap will become orange btn
2.6.
btn-default in bootstrap will become gray btn
3.
3.1.
```

```
//<input type="submit" value="Create" class="btn btn-default" />
It is the submit button of the form.
3.2.
//@using (Html.BeginForm())
//@using (Html.BeginForm("Create", "Gamer"))
These two using will create
//<form action="/Gamer/Create3" method="post" novalidate="novalidate">
It will run the "Gamer" controller and "Create" HttpPost action.
//@using (Html.BeginForm())
It does not specify any controller or any action.
Thus, it will use the default controller and default HttpPost action.
It is Create.cshtml in Views/Gamer folder.
Thus, it will run "Gamer" controller and "Create" HttpPost action
3.3.
//@using (Html.BeginForm("Create2", "Gamer"))
It will create
//<form action="/Gamer/Create2" method="post" novalidate="novalidate">
It will run "Gamer" controller and "Create2" HttpPost action.
4.
4.1.
//@Html.LabelFor(model => model.Name, new { @class = "control-label col-md-2" })
It will create
//<label class="control-label col-md-2" for="Name">Name</label>
4.2.
//@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
It will create
//<input class="form-control text-box single-line" id="Name" name="Name" type="text" value="">
4.3.
//@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
It will create
//<span class="field-validation-valid text-danger" data-valmsg-for="Name" data-valmsg-
replace="true"></span>
The cshtml also use the following JS
//<script src="~/Scripts/jquery-1.10.2.min.js"></script>
//<script src="~/Scripts/jquery.validate.min.js"></script>
//<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
Thus, the span will display the error message of Name input.
4.4.
//@Html.DropDownList("Gender",new List<SelectListItem>
//{
//    new SelectListItem{Text = "Male", Value = "Male"},
//    new SelectListItem{Text = "Female", Value = "Female"}
//})
It will create
//<select id="Gender" name="Gender">
//    <option value="Male">Male</option>
//    <option value="Female">Female</option>
//</select>
4.5.
//@Html.DropDownList("Gender", new List<SelectListItem>
//{
//    new SelectListItem{Text = "Male", Value = "Male"},
//    new SelectListItem{Text = "Female", Value = "Female"}
//}, "Select Gender")
It will create
//<select id="Gender" name="Gender">
//    <option value="">Select Gender</option>
//    <option value="Male">Male</option>
//    <option value="Female">Female</option>
//</select>
4.6.
//@Html.ActionLink("Back to List", "Index2")
It will create
//<a href="/Gamer/Index2">Back to List</a>
*@
```



http://localhost:49271/Gamers/Create
reate - My ASP.NET Applic...
Application name

Create

Gamer

Name

Gender: Select Gender

City

DateOfBirth

TeamId: Select Team

© 2018 - My ASP.NET Application

3.4. Models/Gamers/Gamer.cs

```
using System.ComponentModel.DataAnnotations;

namespace OnlineGame.Web.Models
{
    [MetadataType(typeof(GamerMetaData))]
    public partial class Gamer
    {
        //[Required]
        //public string Name { get; set; }
        //[Required]
        //public string Gender { get; set; }
        //[Required]
        //public string City { get; set; }
        //[Required]
        //[Display(Name = "Team")]
        //public int TeamId { get; set; }
        //// Error!!
        //// Memeber with the same name is aready declared in other auto-generated partial class.
        //// Thus, you need MetadataType to add extra code for the Property.
        //// E.g. [MetadataType(typeof(GamerMetaData))
        //// In this case, you may add some extra code for the Property in MetadataType class
    }
}
```

3.5. Models/Gamers/GamerMetaData.cs

```
using System;
using System.ComponentModel.DataAnnotations;
namespace OnlineGame.Web.Models
{
    public class GamerMetaData
    {
        // Here is the place you may add some extra code for the property
        // which is already in the auto-generate partail class.
        //[Required]
        public string Name { get; set; }
         [Required]
        public string Gender { get; set; }
         [Required]
        public string City { get; set; }
         [Required]
         [Display(Name = "Team")]
        public Nullable<int> TeamId { get; set; }
        //In the database, TeamId is Nullable,
        //so the [Required] attibure here will not affect any thing.
        //If the TeamId in database is not Nullable,
        //then without [Display(Name = "Team")] attibute,
        //the validation message will display "TeamId is required".
        //if it is with [Display(Name = "Team")] attibute,
        //then validation message will display "Team is required".
    }
}
```

3.6. Views/Gamers/Edit.cshtml

```
@model OnlineGame.Web.Models.Gamer
@{
    ViewBag.Title = "Edit";
}
<h2>Edit</h2>
@using (Html.BeginForm())
```

```
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Gamer</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)
        <div class="form-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @*@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })*@

                @Html.HiddenFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
                @Html.DisplayFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Gender, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @*@Html.EditorFor(model => model.Gender, new { htmlAttributes = new { @class = "form-control" } })*@
                @*@Html.DropDownList("Gender",new List<SelectListItem>
                    {
                        new SelectListItem{Text = "Male", Value = "Male"},
                        new SelectListItem{Text = "Female", Value = "Female"}
                    })*@
                @Html.DropDownList("Gender", new List<SelectListItem>
                    {
                        new SelectListItem{Text = "Male", Value = "Male"},
                        new SelectListItem{Text = "Female", Value = "Female"}
                    }, "Select Gender")
                @Html.ValidationMessageFor(model => model.Gender, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.City, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.TeamId, "TeamId", new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("TeamId", null, "Select Team", new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.TeamId, "", new {@class = "text-danger"})
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
@*
1.
1.1.
//@Html.HiddenFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
It will create the following.
//<input data-val="true" data-val-required="The Name field is required." htmlattributes="{ class = form-control }" id="Name" name="Name" type="hidden" value="Name01 ABB">
1.2.
//@Html.DisplayFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
It will create the following.
//Name01 ABB
1.3.
//@Html.EditorFor(model => model.Name, new {htmlAttributes = new {@class = "form-control"}})
It will create the following.
//<input class="form-control text-box single-line valid" id="Name" name="Name" type="text" value="Name01 ABB">
*@
```

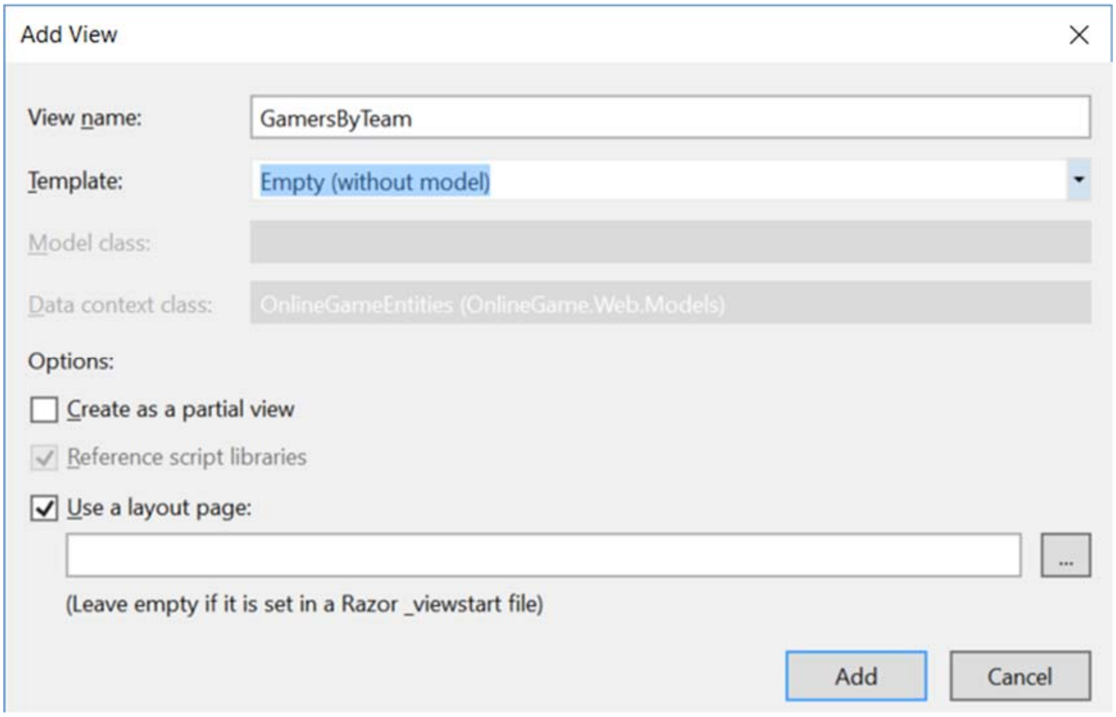
3.7. Models/Teams/TeamTotals.cs

```
namespace OnlineGame.Web.Models
{
    public class TeamTotals
     {
        public string Name { get; set; }
        public int Total { get; set; }
     }
}
```

3.8. Controllers/GamersController.cs

```
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
usingSystem.Net;
using System.Web.Mvc;
using OnlineGame.Web.Models;
namespace OnlineGame.Web.Controllers
{
    public class GamersController : Controller
    {
        private OnlineGameEntities db = new OnlineGameEntities();
        public ActionResult GamersByTeam()
        {
            ////db.Gamers.Include("Team")
            //Retrive the Gamers with their Team data.
            List<TeamTotals> teamTotals =
                db.Gamers.Include("Team")
                .GroupBy(g => g.Team.Name)
                .Select(gamer => new TeamTotals
                {
                    Name = gamer.Key,
                    Total = gamer.Count()
                }).ToList();
            return View(teamTotals);
        }
        // GET: Gamers
        public async Task<ActionResult> Index()
        {
            IQueryable<Gamer> gamers = db.Gamers.Include(g => g.Team);
            //return View(await gamers.ToListAsync());  //~/Views/Gamers/Index.cshtml
            //return View("Index", await gamers.ToListAsync());    //~/Views/Gamers/Index.cshtml
            //return View("Index.cshtml", await gamers.ToListAsync());    //  Error
            return View(await gamers.ToListAsync());
        }
        // GET: Gamers/Details/5
        public async Task<ActionResult> Details(int? id)
        {
            if (id == null)
            {
                //return BadRequest code.
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Gamer gamer = await db.Gamers.FindAsync(id);
            if (gamer == null)
            {
                //return HttpNotFound code.
                return HttpNotFound();
            }
            return View(gamer);
        }
        // GET: Gamers/Create
        public ActionResult Create()
        {
            //Use the collection of teams as the parameter to create SelectList
            //which value is Team Id and the text is Team Name.
            //ViewBag.TeamId will bind this SelectList to View Model TeamId property.
            ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name");
            return View();
        }
        // POST: Gamers/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind to, 
for 
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Create([Bind(Include 
= "Id,Name,Gender,City,DateOfBirth,TeamId")] Gamer gamer)
        {
            // We don't allow Fiddler to compose the Post body to change Name property,
            //so we don't use [Required] attribute on Name property.
            //However, in Create mode, we want to set Name is required property.
            //Thus, we have to dynamically add the ModelState.AddModelError in Create action
            if (string.IsNullOrEmpty(gamer.Name))
            {
                ModelState.AddModelError("Name", "Name is required.");
            }
            if (ModelState.IsValid)
            {
                db.Gamers.Add(gamer);
                await db.SaveChangesAsync();
                return RedirectToAction("Index");
            }
            //Use the collection of teams as the parameter to create SelectList
            //which value is Team Id and the text is Team Name.
            //ViewBag.TeamId will bind this SelectList to View Model TeamId property.
            ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name", gamer.TeamId);
            return View(gamer);
        }
        // GET: Gamers/Edit/5
        public async Task<ActionResult> Edit(int? id)
        {
            if (id == null)
            {
                //return BadRequest code.
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Gamer gamer = await db.Gamers.FindAsync(id);
            if (gamer == null)
            {
                //return HttpNotFound code.
                return HttpNotFound();
            }
            //Use the collection of teams as the parameter to create SelectList
            //which value is Team Id and the text is Team Name.
            //ViewBag.TeamId will bind this SelectList to View Model TeamId property.
            ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name", gamer.TeamId);
            return View(gamer);
        }
        //1.
        // POST: Gamers/Edit/5
        // To protect from overposting attacks, please enable the specific properties you want to bind to, 
for 
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        //2.
        ////public async Task<ActionResult> Edit([Bind(Include = "Id,Gender,City,DateOfBirth,TeamId")]
Gamer gamer)
        //Only update properties in the list, and ignore rest of properties.
        //In this case, update will exclude the Name property.
        //Thus, The post body generated by Fiddler can not update Name property.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Edit([Bind(Include
= "Id,Gender,City,DateOfBirth,TeamId")] Gamer gamer)
        {
            //Get the gamer
            Gamer gamerFromDb = db.Gamers.Single(g => g.Id == gamer.Id);
            //Update the gamerFromDb
            gamerFromDb.Id = gamer.Id;
            gamerFromDb.Gender = gamer.Gender;
            gamerFromDb.City = gamer.City;
            gamerFromDb.TeamId = gamer.TeamId;
            //In the beginning, gamer.Name is null.
            //In order to pass ModelState.IsValid,
            //we need to set value for gamer.Name
            gamer.Name = gamerFromDb.Name;
            if (ModelState.IsValid)
            {
                //Update the entity by gamerFromDb, and set state as EntityState.Modified
                db.Entry(gamerFromDb).State = EntityState.Modified;
                await db.SaveChangesAsync();    //Save changes.
                return RedirectToAction("Index");
            }
            //1.
            //if validation is failed, then stay in the same page.
            //2.
            //Use the collection of teams as the parameter to create SelectList
            //which value is Team Id and the text is Team Name.
            //ViewBag.TeamId will bind this SelectList to View Model TeamId property.
            ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name", gamer.TeamId);
            return View(gamer);
        }
        // GET: Gamers/Delete/5
        public async Task<ActionResult> Delete(int? id)
        {
            if (id == null)
            {
                // bad request.
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            //Get the gamers
            Gamer gamer = await db.Gamers.FindAsync(id);
            if (gamer == null)
            {
                //return HttpNotFound code.
                return HttpNotFound();
            }
            return View(gamer);
        }
        // POST: Gamers/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> DeleteConfirmed(int id)
        {
            Gamer gamer = await db.Gamers.FindAsync(id);
            if (gamer != null) db.Gamers.Remove(gamer);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

3.9. Views/Gamers/GamersByTeam.cshtml


Add View

View name: GamersByTeam

Template: Empty (without model)

Model class:

Data context class: OnlineGameEntities (OnlineGame.Web.Models)

Options:

☐ Create as a partial view

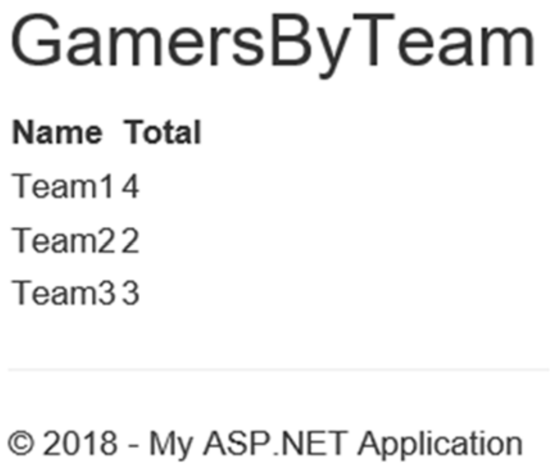
☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

```
@using OnlineGame.Web.Models
@model IEnumerable<TeamTotals>
@{
    ViewBag.Title = "GamersByTeam";
}
<h2>GamersByTeam</h2>
<table>
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Total)
        </th>
    </tr>
    @foreach (TeamTotals item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Total)
            </td>
        </tr>
    }
</table>
```

http://localhost:49271/Gamers/GamersByTeam



GamersByTeam

Name	Total
Team1	4
Team2	2
Team3	3

© 2018 - My ASP.NET Application