(T8)討論 HtmlHelper 的 Label、Password、TextArea、Hidden、TextBox(For)、DropDownList(For)、(RadioButton,CheckBox,ListBox)List
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924
==========
(T8)討論 HtmlHelper 的 Label、Password、TextArea、Hidden、TextBox(For)、DropDownList(For)、(RadioButton,CheckBox,ListBox)List
(T8-1)討論 HtmlHelper 的 Label、Password、TextArea、Hidden、TextBox (1. to 4.3.)
(T8-2)討論 HtmlHelper 的 DropDownList (4.4.)
(T8-3)討論 HtmlHelper 的 TextBox、TextBoxFor、DropDownList、DropDownListFor (4.5. to 4.7.)
(T8-4)討論 HtmlHelper 的 RadiobuttonList (4.8. to 4.9.)
(T8-5)討論 HtmlHelper 的 CheckBoxList (4.10. to 4.11.)
(T8-6)討論 HtmlHelper 的 ListBoxList (4.12.)
==========

==========

0. Summary

In this tutorial, we will discuss
* MvcConventions
* AdoDotNetEntityDataModel
* EntityFramework
* Auto-Generate
* HtmlHelper

* @Html.Label
* @Html.Password
* @Html.TextArea
* @Html.Hidden
* @Html.Dropdownlist
* @Html.TextBox
* @Html.TextBoxFor
* @Html.Radiobuttonlist
* @Html.CheckBoxList
* @Html.ListBox

* 完全攻略 HTML Helper，蝦毀?不夠用?自己寫一個猴塞雷客製化的 HTML Helper 吧。為了(薪)部(水)落
* HTML Helper 包括 Label，Password，TextArea，Hidden，TextBox，TextBoxFor，DropDownList，DropDownListFor，CheckBoxList，ListBoxList。
* 蝦毀?不夠用?自己寫一個猴塞雷客製化的 HTML Helper 吧。為了(薪)部(水)落。

==========

補充 1:

來自社團討論:
https://www.facebook.com/groups/934567793358849/permalink/1455712907910999/

參考資料:
1.
[效能調教] 使用 Async / Await 非同步機制加快 Web API 回應時間
https://dotblogs.com.tw/wasichris/2017/06/08/101137
2.
async 與 await
https://www.huanlintalk.com/2016/01/async-and-await.html
-->
"原範例"程式碼 沒有發輝到"非同步"的優點
只能算是"假的非同步"
***我應該先把要跑的 Task 先準備好
***然後才一起 await
這樣才能發揮出非同步的作用
也感謝同學指正
我也**很喜歡**也很感謝**這種類型的討論
我一直都相信肯定會有比我的範例更好的程式碼
所以只要你有所懷疑, 記得一定要提出來討論
就算把我問倒, 我也會先記起來, 將來去研究, 實驗, 找答案給你
-->
這邊有兩個重點
資料來源:
https://www.huanlintalk.com/2016/01/async-and-await.html
1.
程式的控制流一開始進入非同步方法時，仍是以同步的方式執行，而且是執行於呼叫端所在的執行緒；直到碰到 await 敘述，控制流才會一分為二。基本上，await 之前的程式碼是一個同步執行的程式區塊，而 await 敘述之後的程式碼則為另一個同步執行的程式區塊；兩者分屬不同的控制流。前者即為本章開頭提到的先導工作，後者則是延續的工作——它會在 await 所等待的工作完成之後接著執行。
2.
一個以 async 關鍵字修飾的非同步方法裡面可以有一個或多個 await 敘述。按照先前的講法，若非同步方法中有兩個 await 敘述，即可以理解為該方法被切成三個控制流（三個各自同步執行的程式區塊）。若非同步方法中三個 await 敘述，則表示該方法被切成四個控制流。依此類推。
==========

1. 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

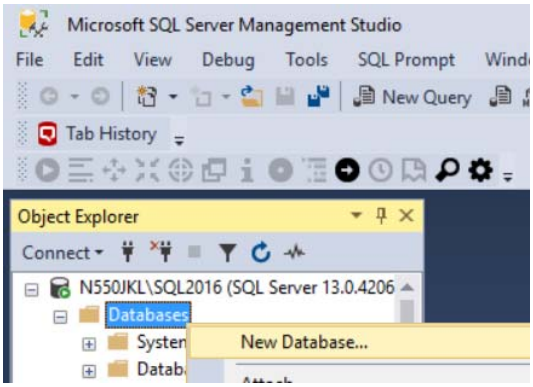
5.1. Views/ControllerName
5.2. Views/Shared
6. The extension name of view can be cshtml, vbhtml, aspx, or ascx.
7. Models can be anywhere, even can be in another project. However, it is better to put it in "Models" folder.
8. You may put Models in another project as business layer.
9. Shared folder stores shared views.
E.g. Master for aspx and Layout pages for cshtml

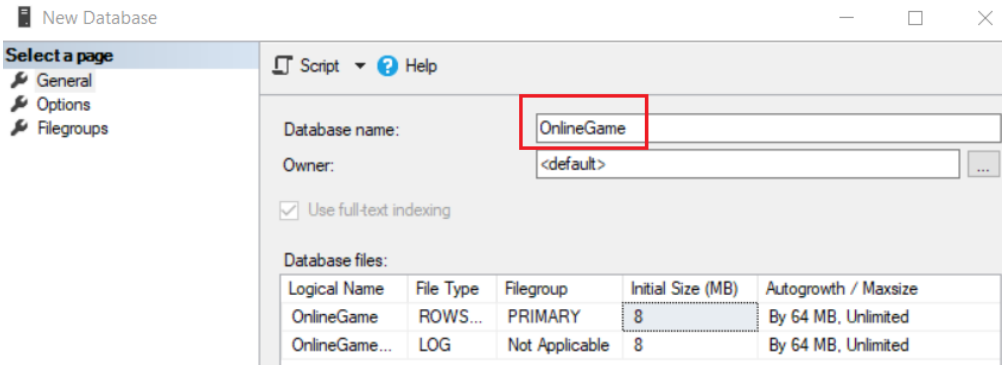
2. OnlineGame DB

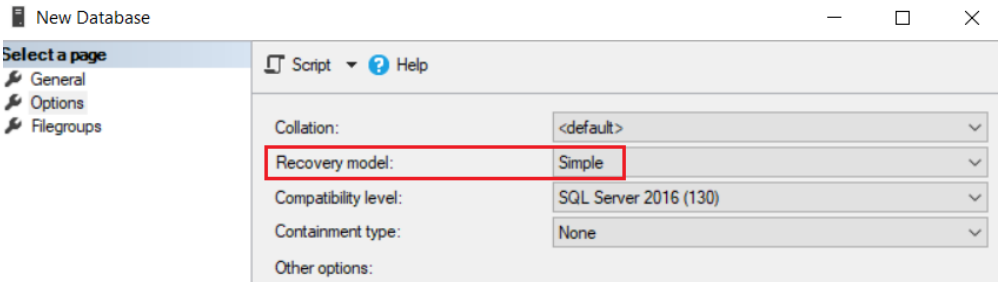
2.1. TSQL

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

In options Tab --> Recovery model : **Simple**








```
--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
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'SingleSelect' ) )
    BEGIN
        TRUNCATE TABLE SingleSelect;
        DROP TABLE SingleSelect;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'MultipleSelect' ) )
    BEGIN
        TRUNCATE TABLE MultipleSelect;
        DROP TABLE MultipleSelect;
    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
CREATE TABLE SingleSelect
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1)
```

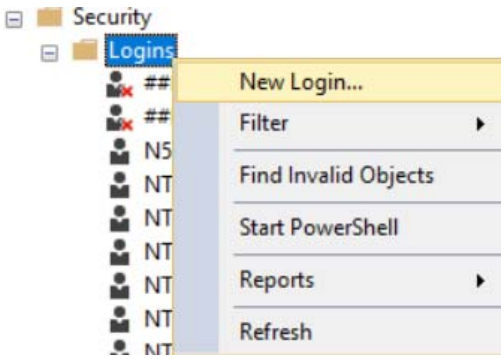
```
            NOT NULL ,
      [Name] NVARCHAR(100) NOT NULL ,
      IsSelected BIT NOT NULL
   );
GO -- Run the previous command and begins new batch
CREATE TABLE MultipleSelect
   (
      Id INT PRIMARY KEY
            IDENTITY(1, 1)
            NOT NULL ,
      [Name] NVARCHAR(100) NOT NULL ,
      IsSelected BIT NOT NULL
   );
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' );
GO -- Run the previous command and begins new batch
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
INSERT  SingleSelect
VALUES  ( N'SingleSelect Item A', 0);
INSERT  SingleSelect
VALUES  ( N'SingleSelect Item B', 1 );
INSERT  SingleSelect
VALUES  ( N'SingleSelect Item C', 0 );
GO -- Run the previous command and begins new batch
INSERT  MultipleSelect
VALUES  ( N'MultipleSelect Item A', 0);
INSERT  MultipleSelect
VALUES  ( N'MultipleSelect Item B', 0 );
INSERT  MultipleSelect
VALUES  ( N'MultipleSelect Item C', 0 );
INSERT  MultipleSelect
VALUES  ( N'MultipleSelect Item D', 0);
INSERT  MultipleSelect
```

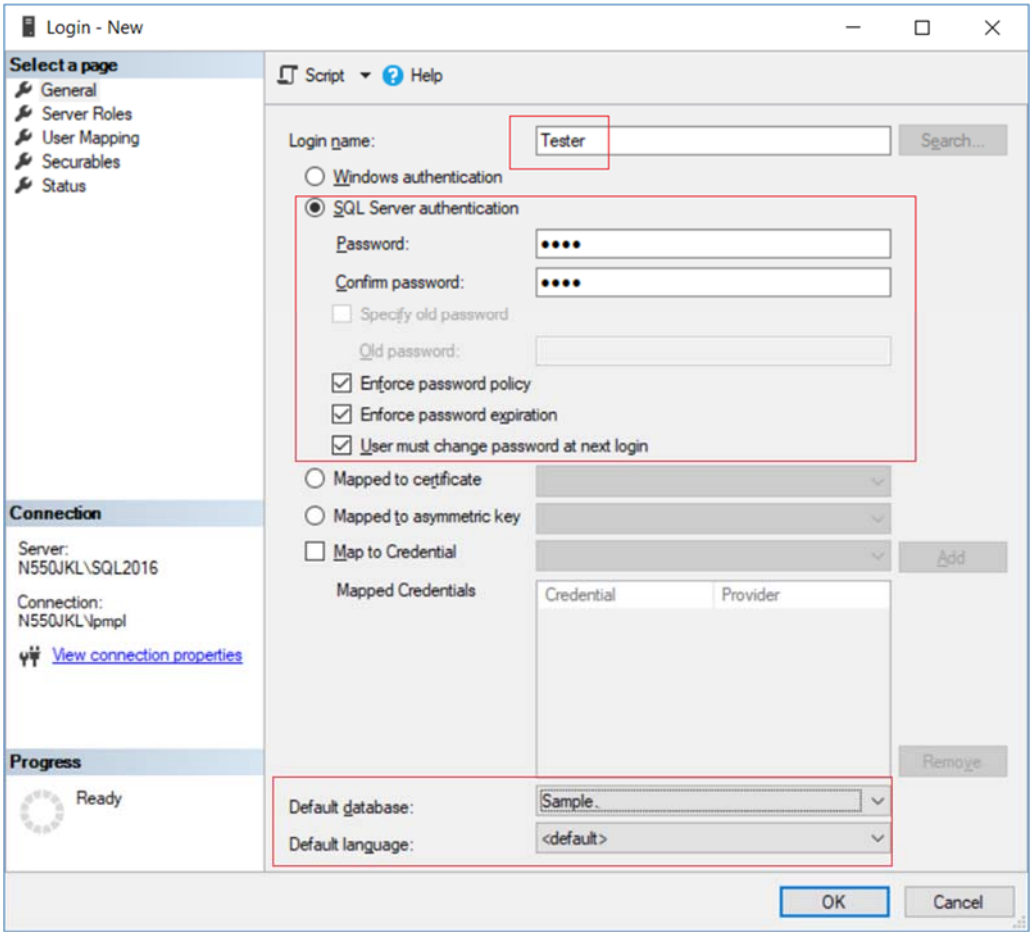
```
VALUES  ( N'MultipleSelect Item E', 0 );
INSERT  MultipleSelect
VALUES  ( N'MultipleSelect Item F', 0 );
GO -- Run the previous command and begins new batch
--4. SP
CREATE PROCEDURE spGetGamers
AS
    BEGIN
        SELECT  *
        FROM    Gamer;
    END;
GO -- Run the previous command and begins 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
```

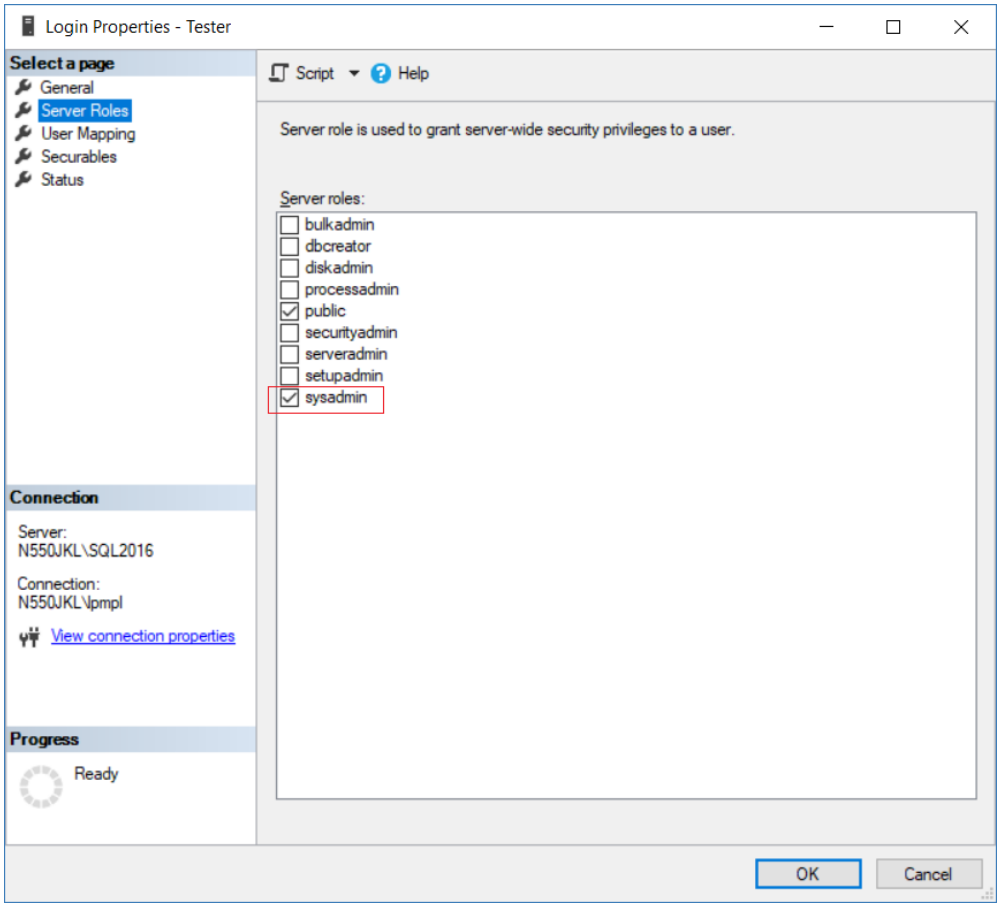
```
--GO -- Run the previous command and begins new batch
```

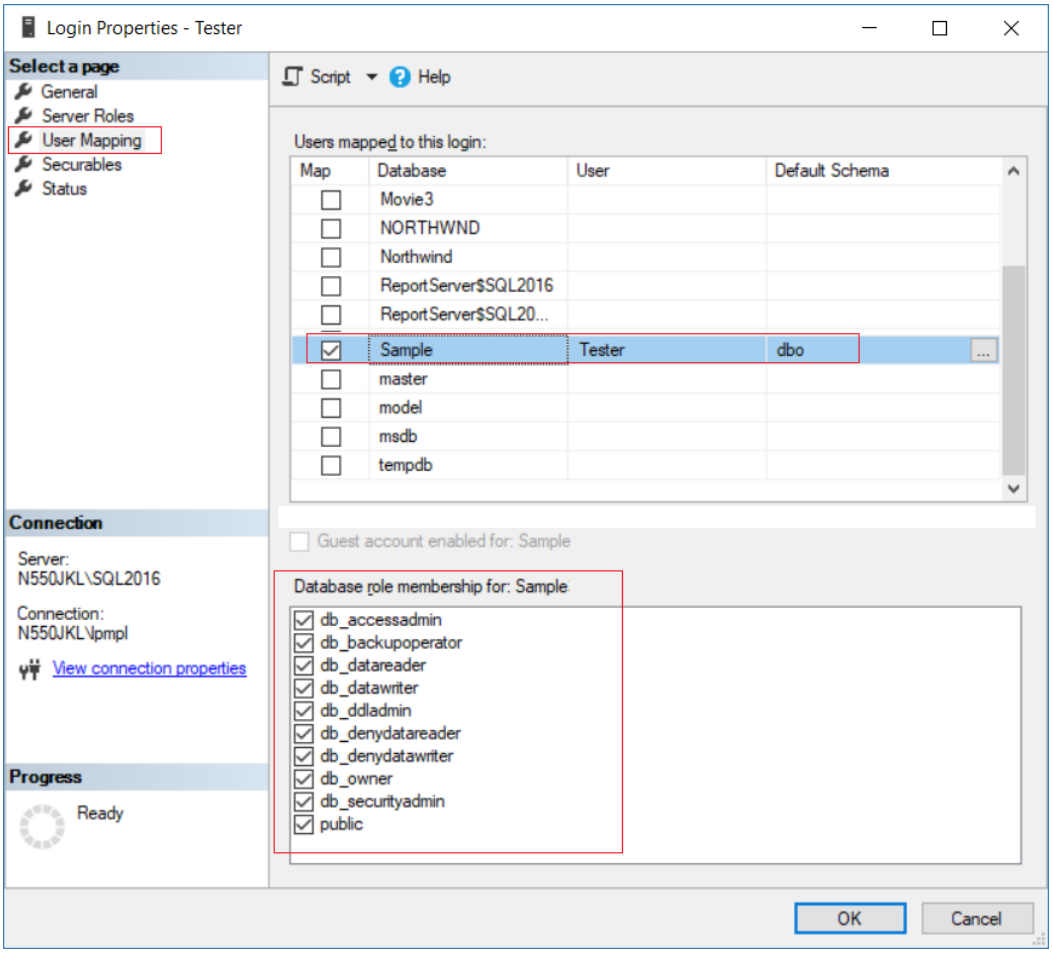
2.2. Security login

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.



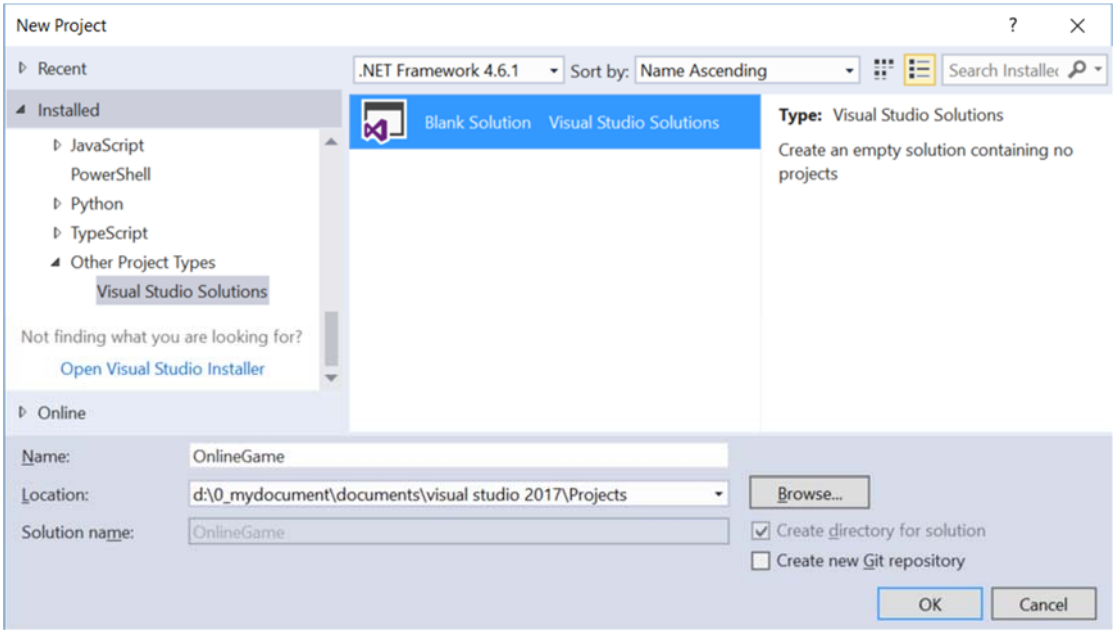

Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Tester
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
Sample
Default language:
<default>
Connection
Server:
N550JKL\SQL2016
Connection:
N550JKL\lpmpl
View connection properties
Progress
Ready
OK
Cancel


Login Properties - Tester
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Server role is used to grant server-wide security privileges to a user.
Server roles:
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connection
Server:
N550JKL\SQL2016
Connection:
N550JKL\lpmpl
View connection properties
Progress
Ready
OK
Cancel



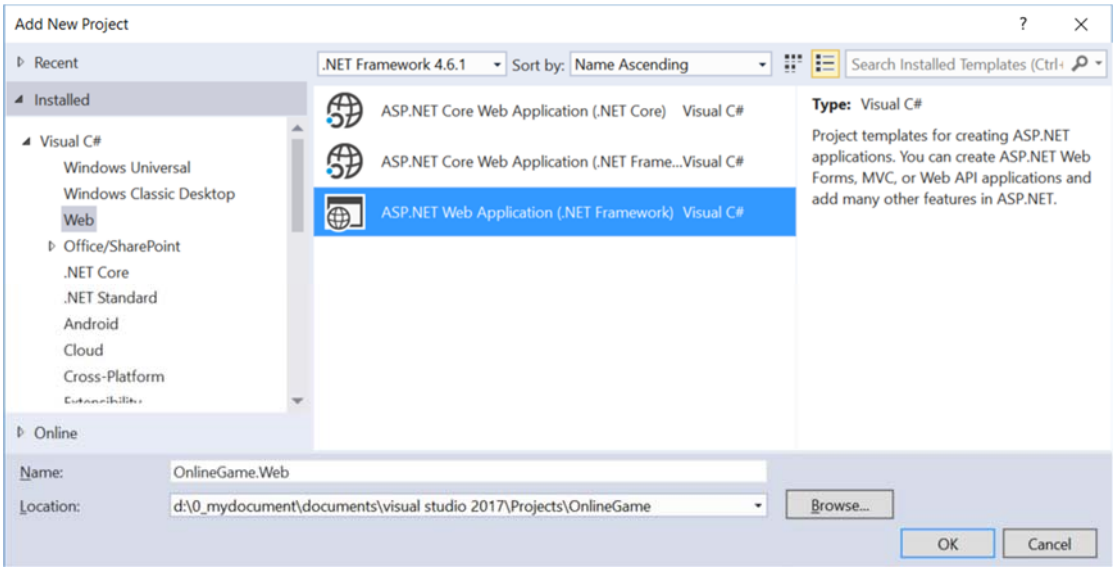
3. New Project - OnlineGame

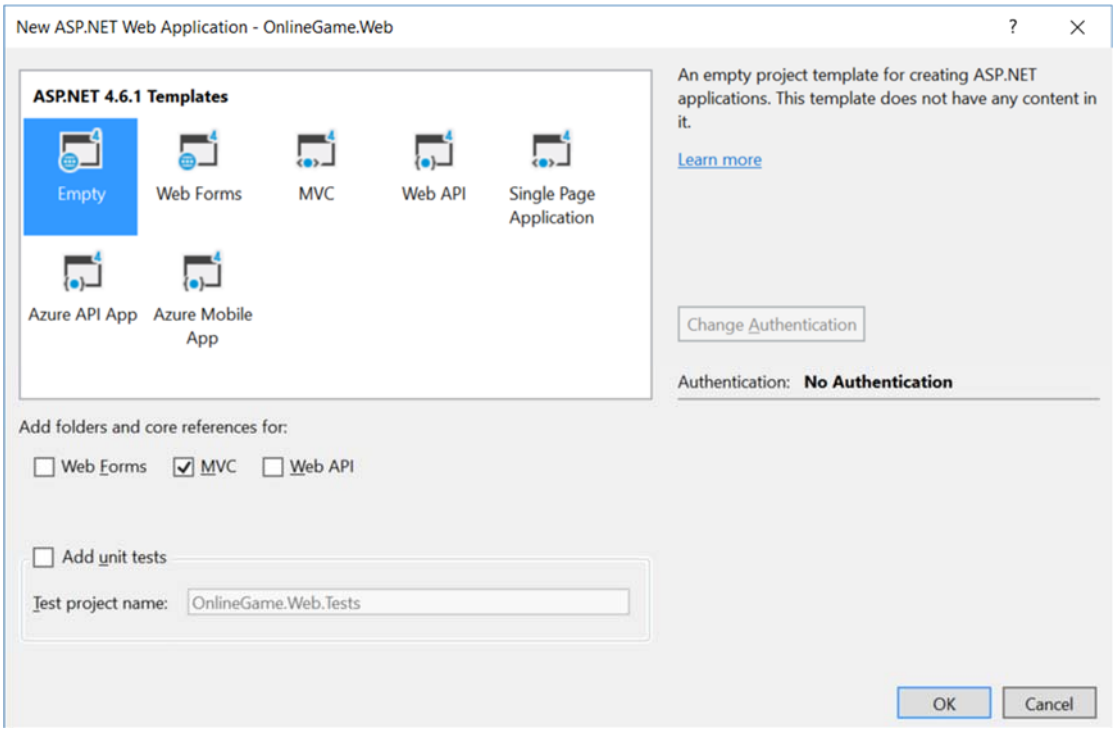
File --> New --> Project... -->
Other Project Types --> Visual Studio Solutions --> Blank Solution
-->
Name: **OnlineGame**



3.1. 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





3.1.1. 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);
        }
    }
}
```

3.1.2. 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/
```

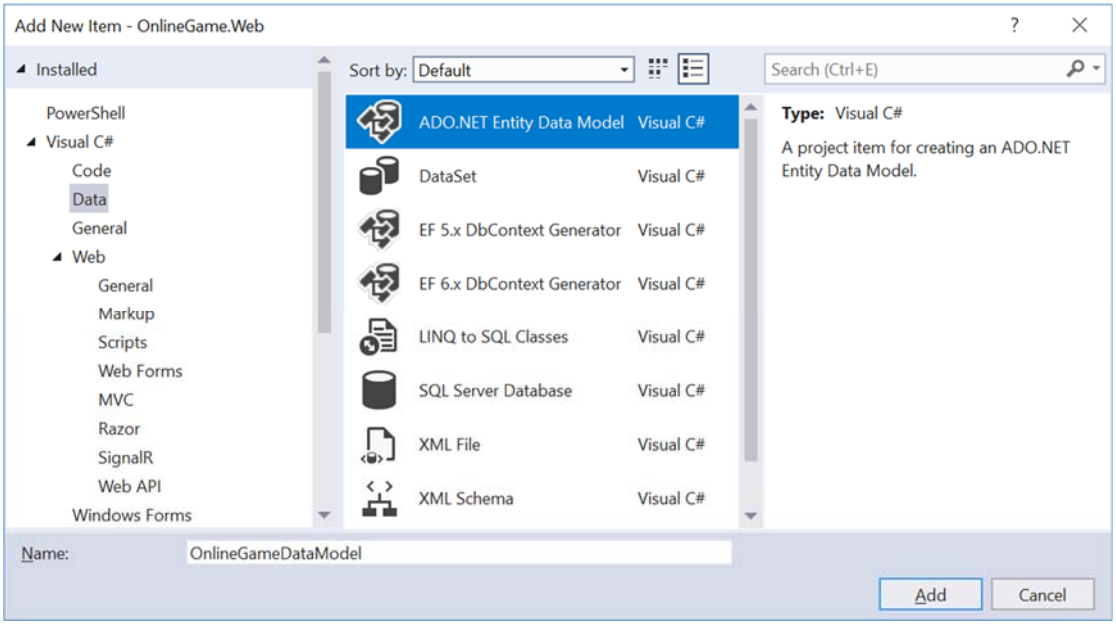
```
IIS Express will run
HomeController Index action.
It will map to Controllers/HomeController.cs
and   map to Index Method
1.2.
By convention in MVC.
All controllers will have Controller suffix.
This suffix is not required in the URL.
So, if you want to invoke Home controller,
you specify /Home and 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.
*/
```

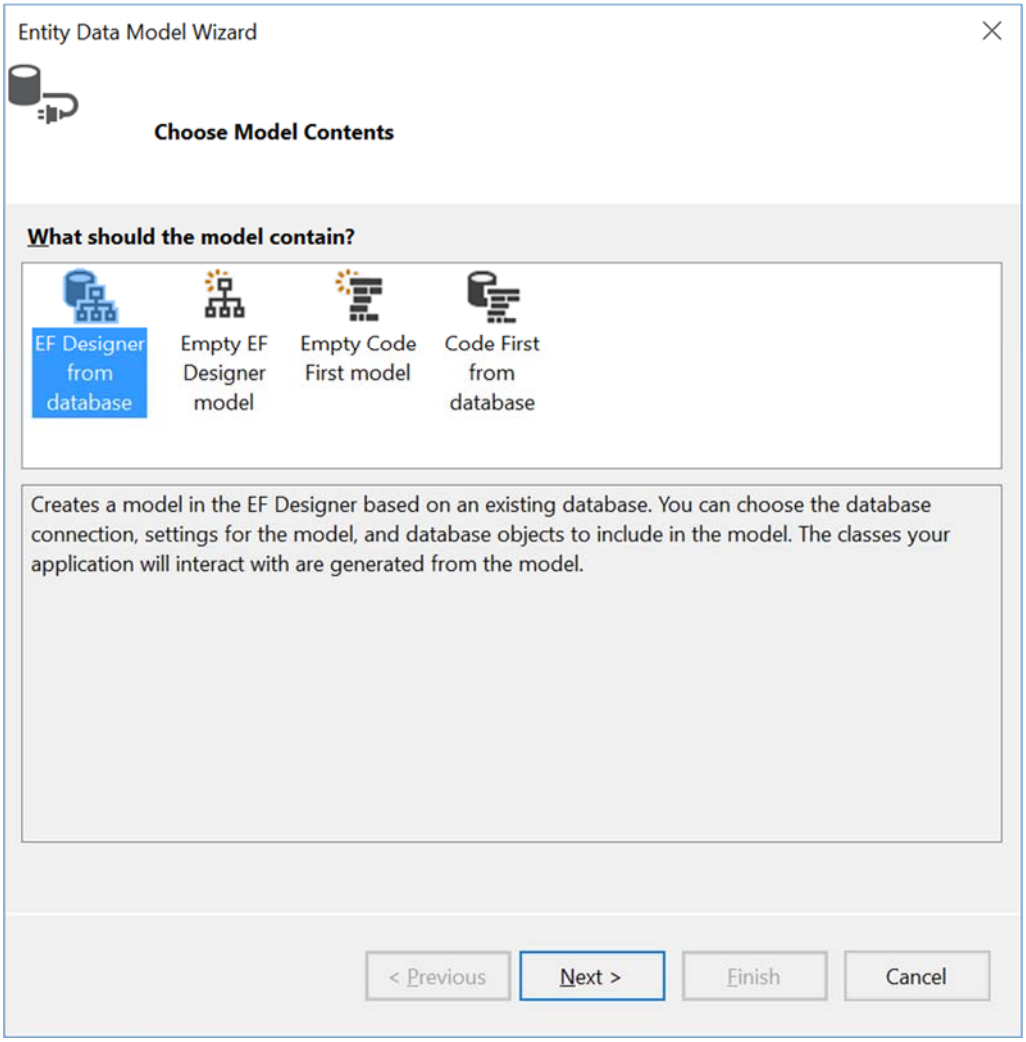
3.2. ADO.Net Entity Data Model - Entity Framework

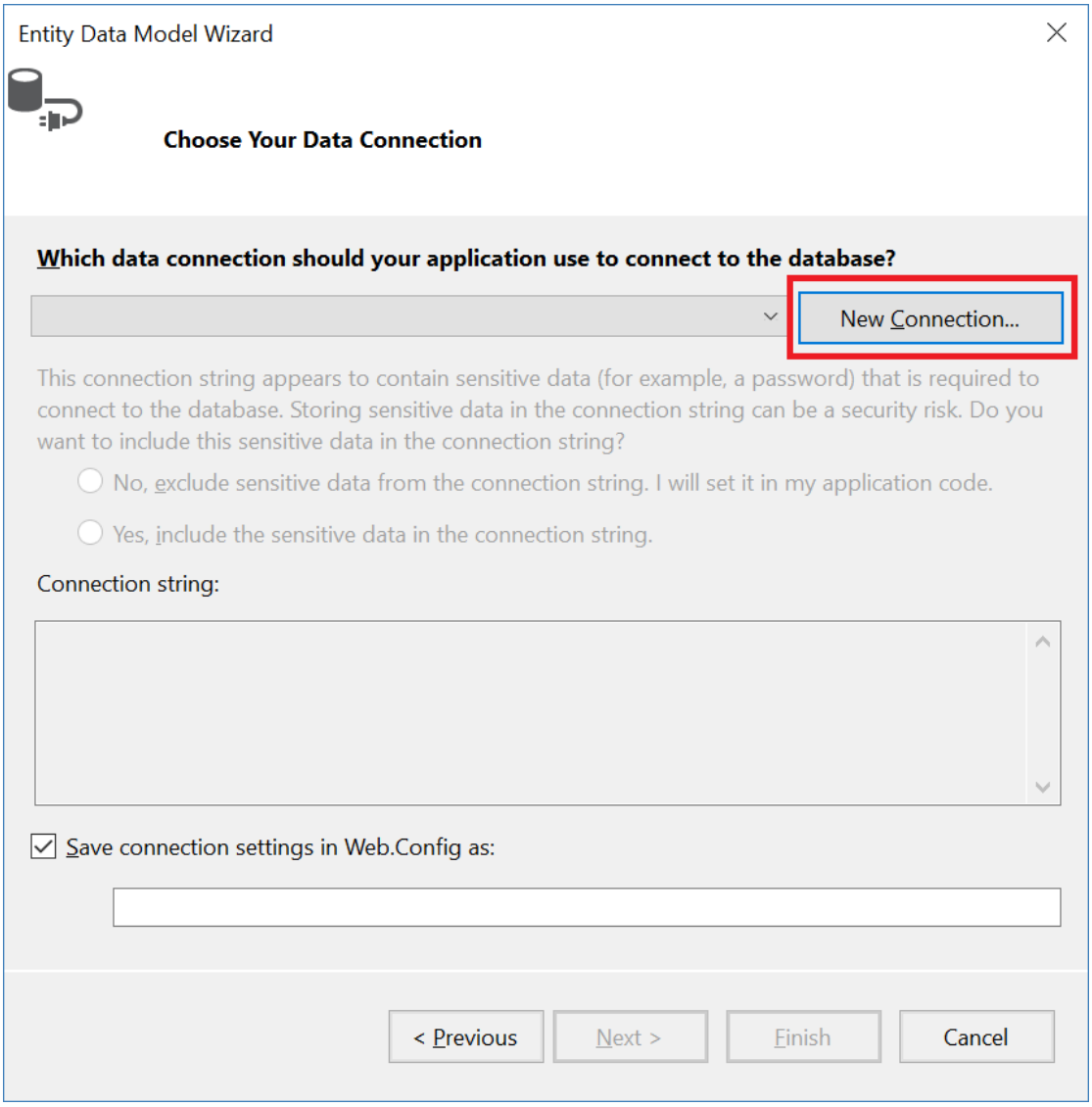
In Visual Studio 2017
Models folder --> Right Click --> Add --> New Item
--> Visual C# --> Data --> ADO.Net Entity Data Model
Name:
OnlineGameDataModel

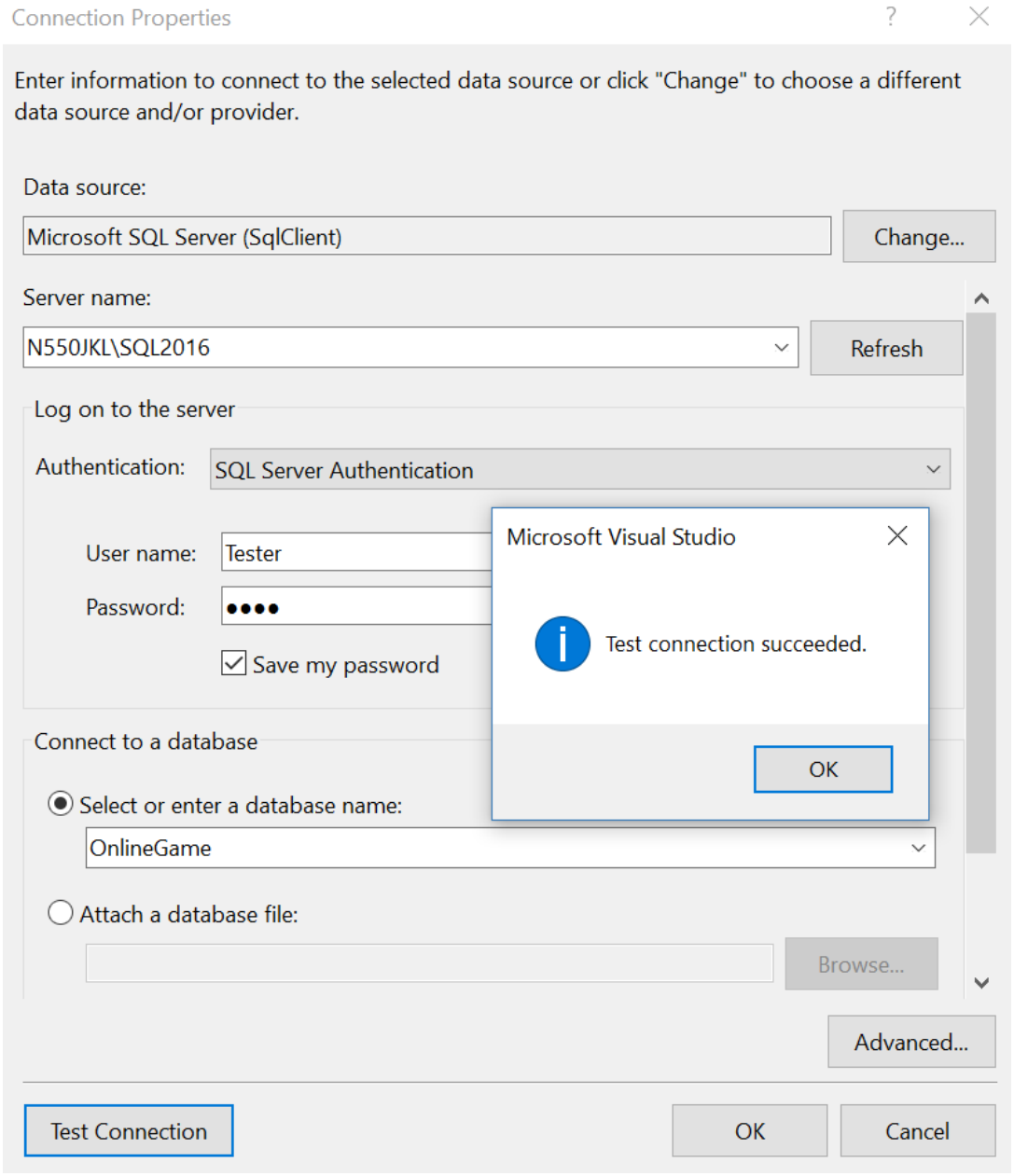
-->
EF Designer from database
....

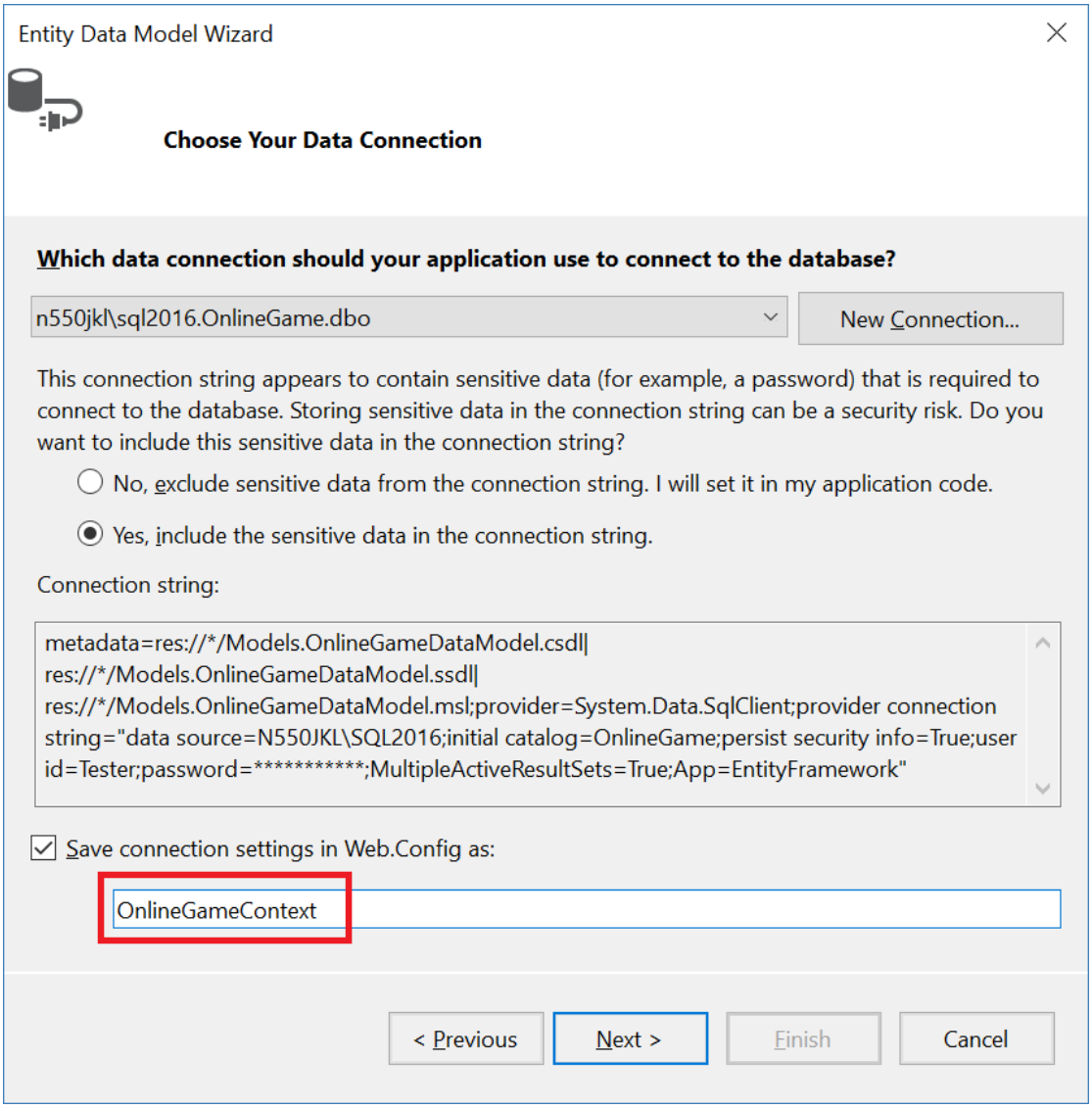
-->
Save Connection settings in Web.Config as:
OnlineGameContext

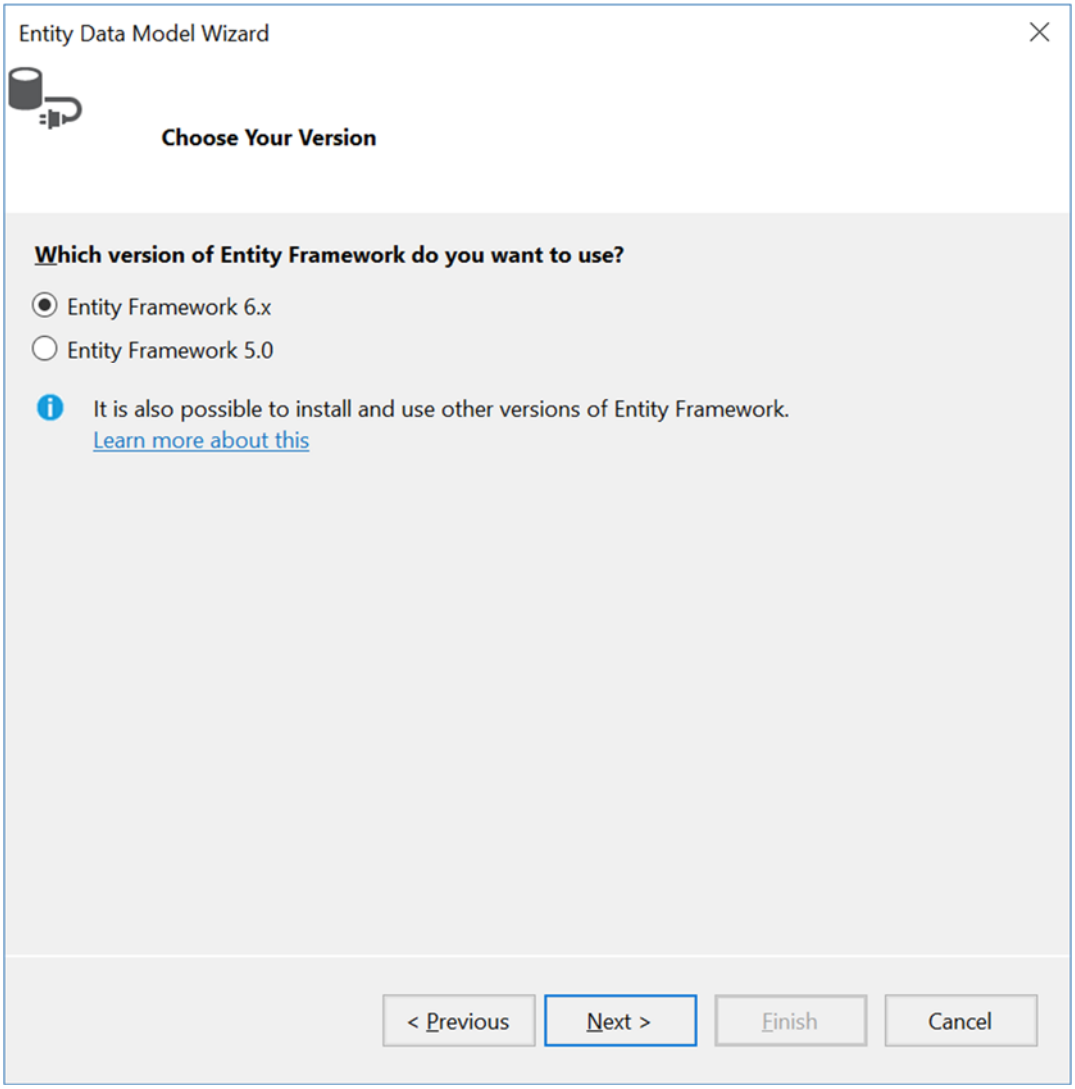

Add New Item - OnlineGame.Web
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Web API
Windows Forms
Sort by: Default
Search (Ctrl+E)
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
SQL Server Database Visual C#
XML File Visual C#
XML Schema Visual C#
Type: Visual C#
A project item for creating an ADO.NET Entity Data Model.
Name: OnlineGameDataModel
Add
Cancel

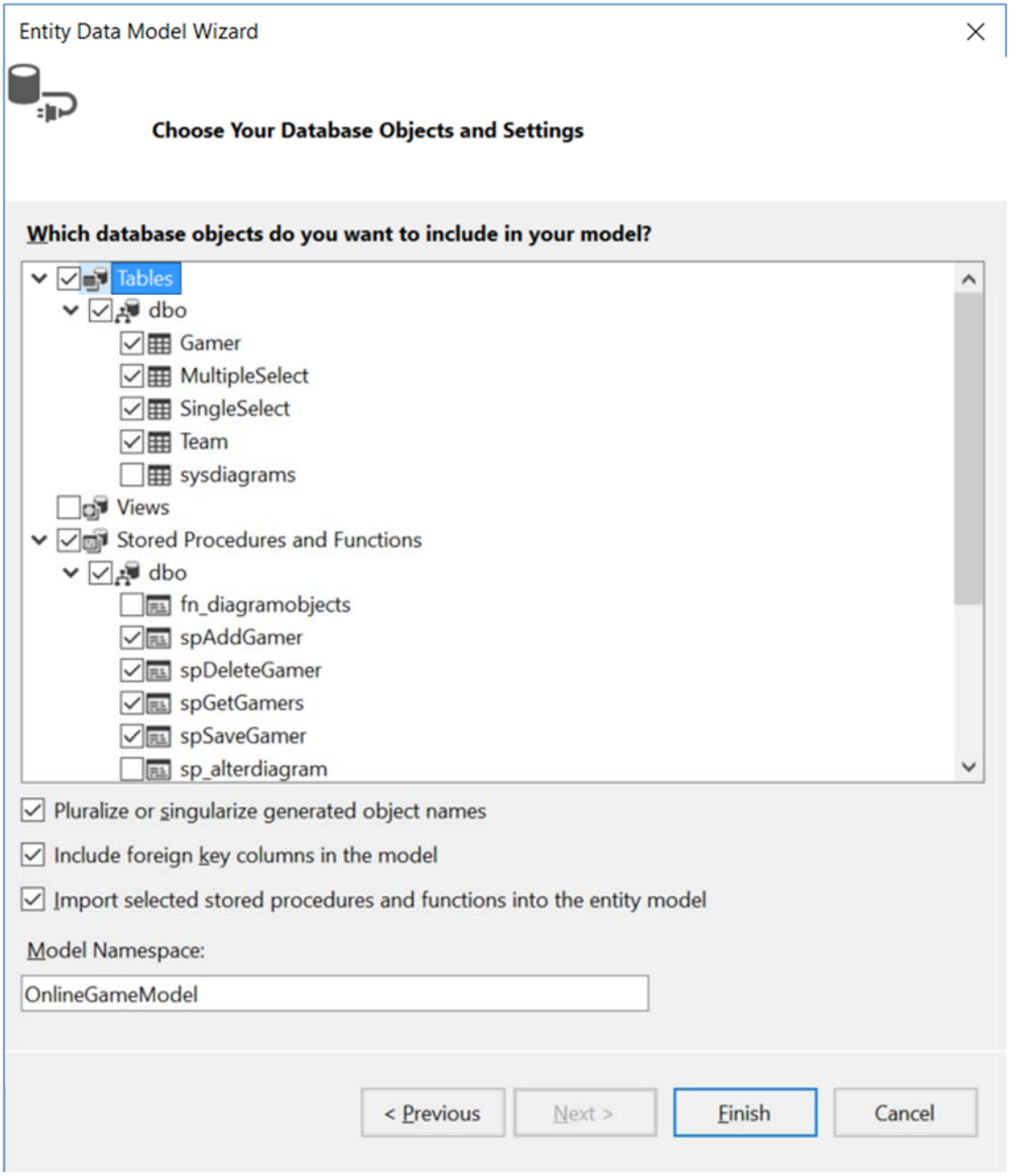

Entity Data Model Wizard
Choose Model Contents
What should the model contain?
EF Designer from database
Empty EF Designer model
Empty Code First model
Code First from database
Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.
< Previous
Next >
Finish
Cancel

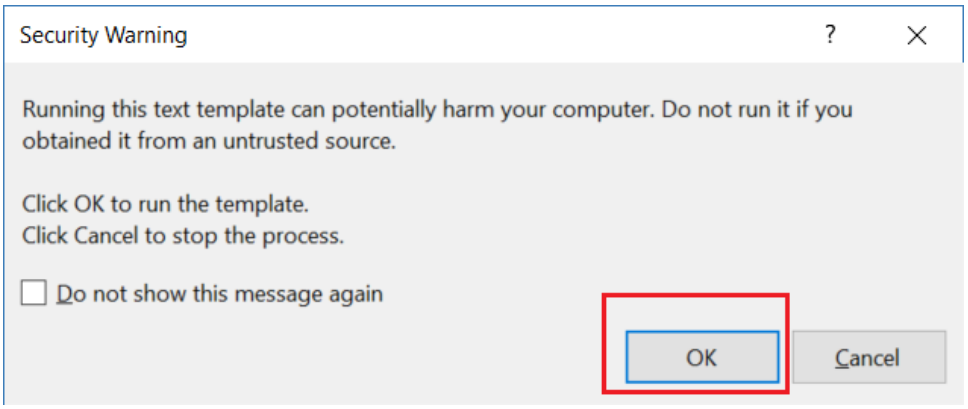

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in 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:
< Previous
Next >
Finish
Cancel

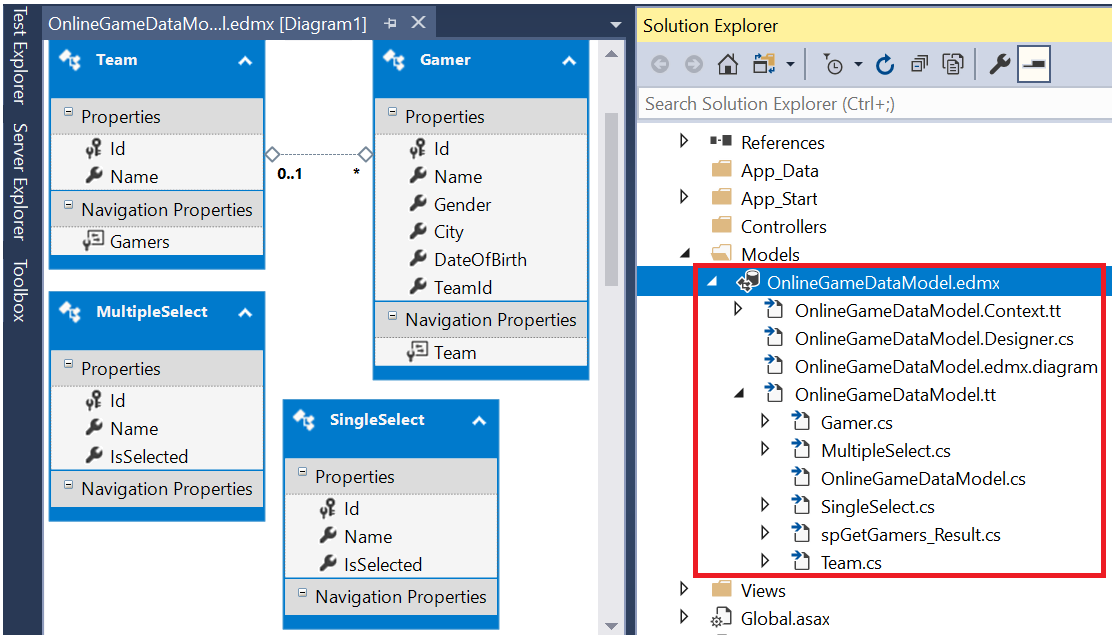

Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
N550JKL\SQL2016
Refresh
Log on to the server
Authentication:
SQL Server Authentication
User name:
Tester
Password:
Save my password
Microsoft Visual Studio
Test connection succeeded.
OK
Connect to a database
Select or enter a database name:
OnlineGame
Attach a database file:
Browse...
Advanced...
Test Connection
OK
Cancel


Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
n550jkl\sql2016.OnlineGame.dbo
New Connection...
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.OnlineGameDataModel.csdl|
res://*/Models.OnlineGameDataModel.ssdl|
res://*/Models.OnlineGameDataModel.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:
OnlineGameContext
< Previous
Next >
Finish
Cancel


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
< Previous
Next >
Finish
Cancel


Entity Data Model Wizard
Choose Your Database Objects and Settings
Which database objects do you want to include in your model?
Tables
dbo
Gamer
MultipleSelect
SingleSelect
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
< Previous
Next >
Finish
Cancel


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
OK
Cancel



4. OnlineGame.Web

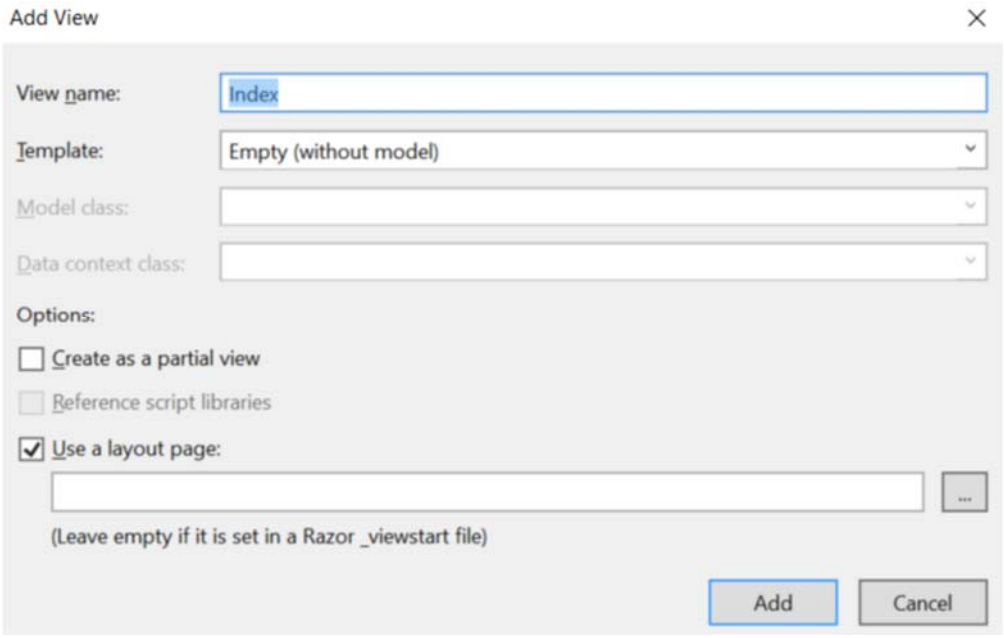
4.1. Controllers/GamersController.cs

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Mvc;
using OnlineGame.Web.Models;
namespace OnlineGame.Web.Controllers
{
    public class GamersController : Controller
    {
        private OnlineGameContext _dbContext = new OnlineGameContext();
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
        [HttpGet]
        public ActionResult HtmlHelpers()
        {
            return View();
        }
        [HttpGet]
        public async Task<ActionResult> Dropdownlist()
        {
            //Use the collection of teams as the parameter to create SelectList
            //which value is Team Id and the text is Team Name.
```

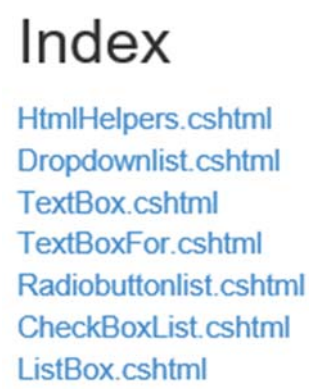
```
            //ViewBag.TeamId will bind this SelectList to View Model control components, TeamId1 and
TeamId2.
            ViewBag.TeamId1 = new SelectList(await _dbContext.Teams.ToListAsync(), "Id", "Name");
            ViewBag.TeamId2 = new SelectList(await _dbContext.Teams.ToListAsync(), "Id", "Name", 2);
            List<SelectListItem> selectListItems = new List<SelectListItem>();
            foreach (SingleSelect singleSelectItem in await _dbContext.SingleSelects.ToListAsync())
            {
                SelectListItem selectListItem = new SelectListItem
                {
                    Text = singleSelectItem.Name,
                    Value = singleSelectItem.Id.ToString(),
                    Selected = singleSelectItem.IsSelected
                };
                selectListItems.Add(selectListItem);
            }
            ViewBag.selectListItems1 = selectListItems;
            return View();
        }
        [HttpGet]
        public ActionResult TextBox()
        {
            Game game = new Game("GameA");
            ViewBag.GameName = game.Name;
            ViewBag.GameTeams = new SelectList(game.Teams, "Id", "Name");
            return View();
        }
        [HttpGet]
        public ActionResult TextBoxFor()
        {
            Game game = new Game("GameA");
            return View(game);
        }
        [HttpGet]
        public ActionResult Radiobuttonlist()
        {
            Game game = new Game("GameA");
            return View(game);
        }
        [HttpPost]
        public string Radiobuttonlist(Game game)
        {
            return string.IsNullOrEmpty(game.SelectedItemId)
                ? "Nothing is selected"
                : $"Selected Id == {game.SelectedItemId}";
            //return RedirectToAction("Index");
        }
        [HttpGet]
        public async Task<ActionResult> CheckBoxList()
        {
            List<MultipleSelect> multipleSelects =
                await _dbContext.MultipleSelects.ToListAsync();
            return View(multipleSelects);
        }
        [HttpPost]
        public string CheckBoxList(IEnumerable<MultipleSelect> multipleSelects)
        {
            IEnumerable<MultipleSelect> enumerable = multipleSelects as MultipleSelect[] ??
multipleSelects.ToArray();
            if (enumerable.Count(x => x.IsSelected) == 0)
            {
```

```
                return "Nothing is selected";
            }
            StringBuilder sb = new StringBuilder();
            sb.Append("Selected Items - ");
            foreach (MultipleSelect item in enumerable)
            {
                if (item.IsSelected)
                {
                    sb.Append($"{item.Name}, ");
                }
            }
            sb.Remove(sb.ToString().LastIndexOf(",", StringComparison.Ordinal), 1);
            return sb.ToString();
        }
        [HttpGet]
        public async Task<ActionResult> ListBox()
        {
            //Create List<SelectListItem> for ListBox
            //Retrive data from DB
            List<MultipleSelect> multipleSelects =
                await _dbContext.MultipleSelects.ToListAsync();
            List<SelectListItem> listSelectListItems =
                multipleSelects.Select(
                    item => new SelectListItem
                    {
                        Text = item.Name,
                        Value = item.Id.ToString(),
                        Selected = item.IsSelected
                    }).ToList();
            MultipleSelectViewModel multipleSelectViewModel = new MultipleSelectViewModel
            {
                MultipleSelectItems = listSelectListItems
            };
            return View(multipleSelectViewModel);
        }
        [HttpPost]
        public string ListBox(IEnumerable<string> selectedItemIds)
        {
            if (selectedItemIds == null)
            {
                return "No cities selected";
            }
            StringBuilder sb = new StringBuilder();
            sb.Append($"Selected ID - {string.Join(", ", selectedItemIds)}");
            return sb.ToString();
        }
    }
}
```

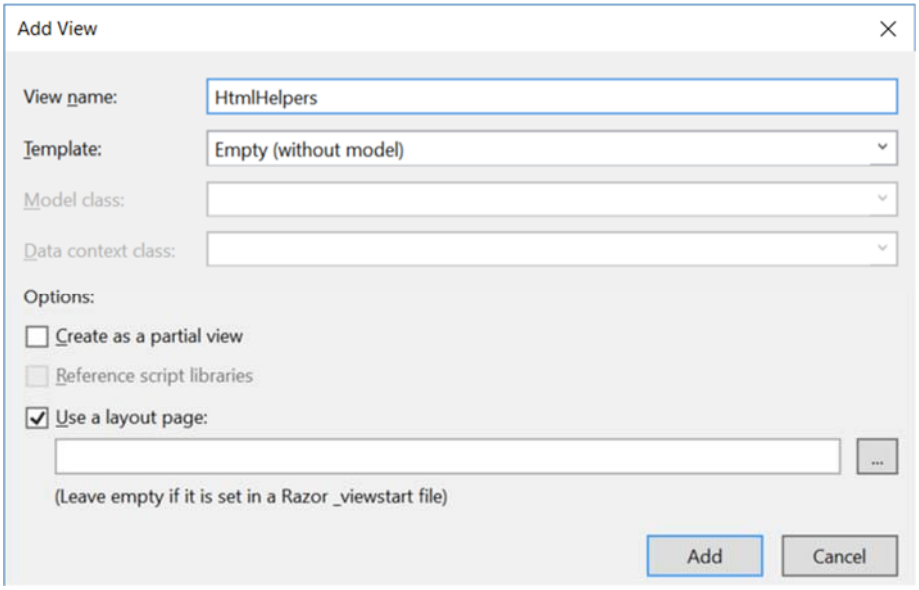
4.2. Views/Gamers/Index.cshtml




```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
@Html.ActionLink("HtmlHelpers.cshtml", "HtmlHelpers", "Gamers")
<br />
@Html.ActionLink("Dropdownlist.cshtml", "Dropdownlist", "Gamers")
<br />
@Html.ActionLink("TextBox.cshtml", "TextBox", "Gamers")
<br />
@Html.ActionLink("TextBoxFor.cshtml", "TextBoxFor", "Gamers")
<br />
@Html.ActionLink("Radiobuttonlist.cshtml", "Radiobuttonlist", "Gamers")
<br />
@Html.ActionLink("CheckBoxList.cshtml", "CheckBoxList", "Gamers")
<br />
@Html.ActionLink("ListBox.cshtml", "ListBox", "Gamers")
<br />
```



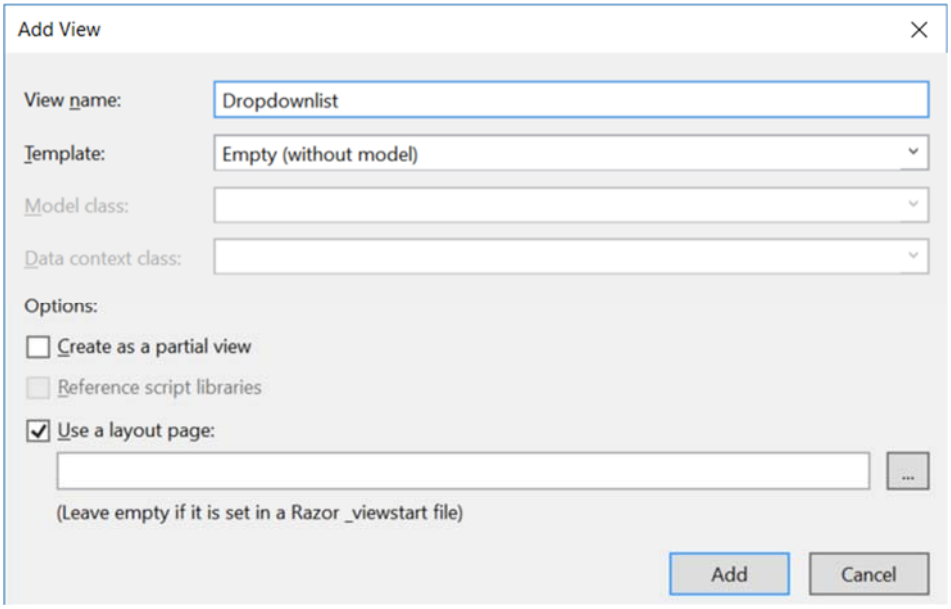
4.3. Views/Gamers/HtmlHelpers.cshtml




```
@{
    ViewBag.Title = "HtmlHelpers";
}
<h2>HtmlHelpers</h2>
@Html.TextBox("Name")
<br />
@Html.TextBox("Name2", "Name2Value")
<br />
@Html.TextBox("Name3", "Name3Value", new { style = "background-color:Gray; color:White; font-weight:bold",
title="Please enter Name3Value" })
<br />
@Html.TextBox("Name4", "Name4Value", new { @class = "ClassStyle", @readonly="true" })
<br />
@Html.Label("LastName", "Last Name")
<br />
@Html.Password("Password")
<br />
@Html.TextArea("Note", "", 10, 20, null)
<br />
@Html.Hidden("Id")
@*
1.
HtmlHelper contains a lot of extension methods to create form element.
You may use HTML to write a view,
but @HTML Helper can simplify the HTML code.
1.1.
//@Html.TextBox("Name")
It will create the following.
//<input id="Name" name="Name" type="text" value="">
1.2.
//@Html.TextBox("Name2", "Name2Value")
It will create the following.
//<input id="Name2" name="Name2" type="text" value="Name2Value">
1.3.
//@Html.TextBox("Name3", "Name3Value", new { style = "background-color:Gray; color:White; font-
weight:bold", title="Please enter Name3Value" })
It will create the following.
//<input id="Name3" name="Name3" style="background-color:Gray; color:White; font-weight:bold"
title="Please enter Name3Value" type="text" value="Name3Value">
1.4.
Some of HTMl attributes are reserved keywords.
E.g. Class, ReadOnly
```

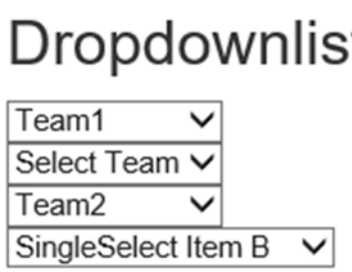
```
In this case, it needs @.
//@Html.TextBox("Name4", "Name4Value", new { @class = "ClassStyle", @readonly="true" })
It will create the following.
//<input class="ClassStyle" id="Name4" name="Name4" readonly="true" type="text" value="Name4Value">
1.5.
//@Html.Label("LastName", "Last Name")
It will create the following.
//<label for="LastName">Last Name</label>
1.6.
//@Html.Password("Password")
It will create the following.
//<input id="Password" name="Password" type="password">
1.7.
@Html.TextArea("Note", "", 10, 20, null)
It will create the following.
//<textarea cols="20" id="Note" name="Note" rows="10"></textarea>
1.8.
@Html.Hidden("Id")
It will create the following.
//<input id="Id" name="Id" type="hidden" value="">
*@
```

4.4. Views/Gamers/Dropdownlist.cshtml




```
@{
    ViewBag.Title = "Dropdownlist";
}
<h2>Dropdownlist</h2>
@Html.DropDownList("Teams", new List<SelectListItem>
{
new SelectListItem { Text = "Team1", Value = "1", Selected=true},
new SelectListItem { Text = "Team2", Value = "2"},
new SelectListItem { Text = "Team3", Value = "3"}
}, "Select Team")
<br />
@Html.DropDownList("TeamId1", "Select Team")
<br />
@Html.DropDownList("TeamId2", "Select Team")
<br />
@Html.DropDownList("selectListItems1", "Select singleList Item")
<br />
@*
1.
@Html.DropDownList will create a Select tag as dropdown list.
----------------------
1.1.
//@Html.DropDownList("Teams", new List<SelectListItem>
//{
//    new SelectListItem { Text = "Team1", Value = "1", Selected=true},
//    new SelectListItem { Text = "Team2", Value = "2"},
//    new SelectListItem { Text = "Team3", Value = "3"}
//}, "Select Team")
It will create the following.
//<select id="Teams" name="Teams"><option value="">Select Team</option>
//    <option selected="selected" value="1">Team1</option>
//    <option value="2">Team2</option>
//    <option value="3">Team3</option>
//</select>
----------------------
1.2.
//ViewBag.TeamId1 = new SelectList(_dbContext.Teams, "Id", "Name");
...
//@Html.DropDownList("TeamId1", "Select Team")
It will create the following.
//<select id="TeamId1" name="TeamId1"><option value="">Select Team</option>
```

```
//    <option value="1">Team1</option>
//    <option value="2">Team2</option>
//    <option value="3">Team3</option>
//</select>
---------------------
1.3.
//ViewBag.TeamId2 = new SelectList(_dbContext.Teams, "Id", "Name", 2);
...
//@Html.DropDownList("TeamId2", "Select Team")
It will create the following.
//<select id="TeamId2" name="TeamId2"><option value="">Select Team</option>
//    <option value="1">Team1</option>
//    <option selected="selected" value="2">Team2</option>
//    <option value="3">Team3</option>
//</select>
---------------------
1.4.
//List<SelectListItem> selectListItems = new List<SelectListItem>();
//foreach (SingleSelect singleSelectItem in _dbContext.SingleSelects)
//{
//    SelectListItem selectListItem = new SelectListItem
//    {
//        Text = singleSelectItem.Name,
//        Value = singleSelectItem.Id.ToString(),
//        Selected = singleSelectItem.IsSelected ?? false
//    };
//    selectListItems.Add(selectListItem);
//}
//ViewBag.selectListItems1 = selectListItems;
...
//@Html.DropDownList("selectListItems1", "Select singleList Item")
It will create the following.
//<select id="selectListItems1" name="selectListItems1"><option value="">Select singleList Item</option>
//    <option value="1">SingleSelect Item A</option>
//    <option selected="selected" value="2">SingleSelect Item B</option>
//    <option value="3">SingleSelect Item C</option>
//</select>
*@
```

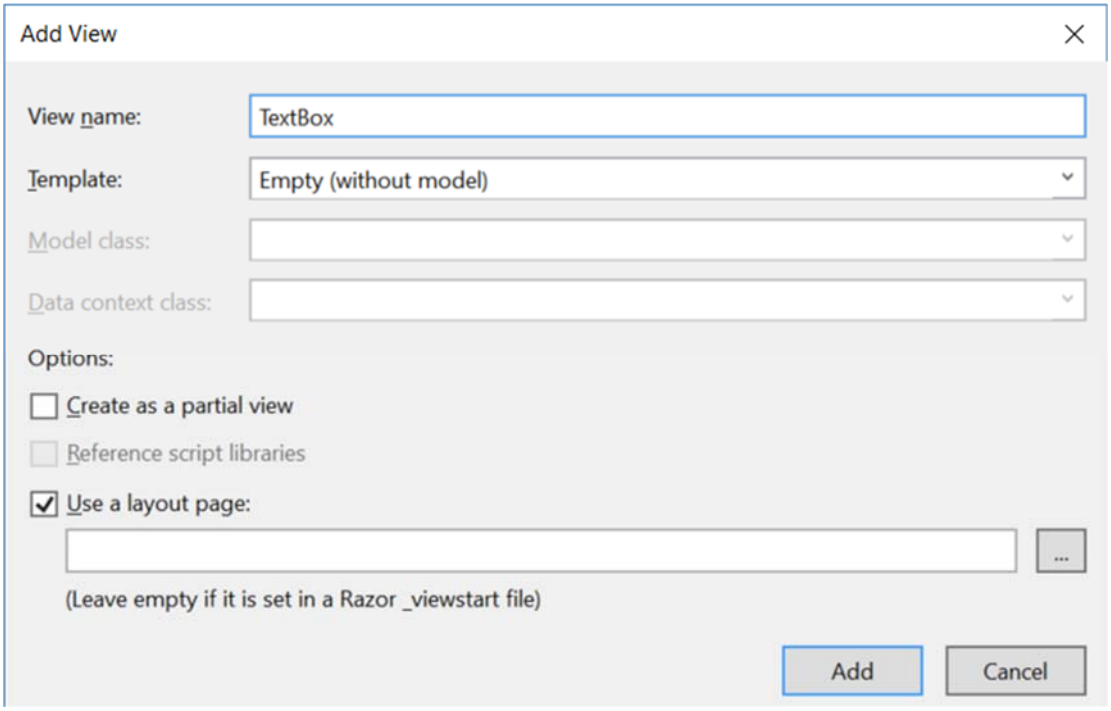


4.5. Models/Gamers/Game.cs

```
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
namespace OnlineGame.Web.Models
{
    public class Game
    {
        public string Name { get; set; }
        public SingleSelect SelectedItem { get; set; }
        public string SelectedItemId { get; set; }
        public List<Team> Teams
```

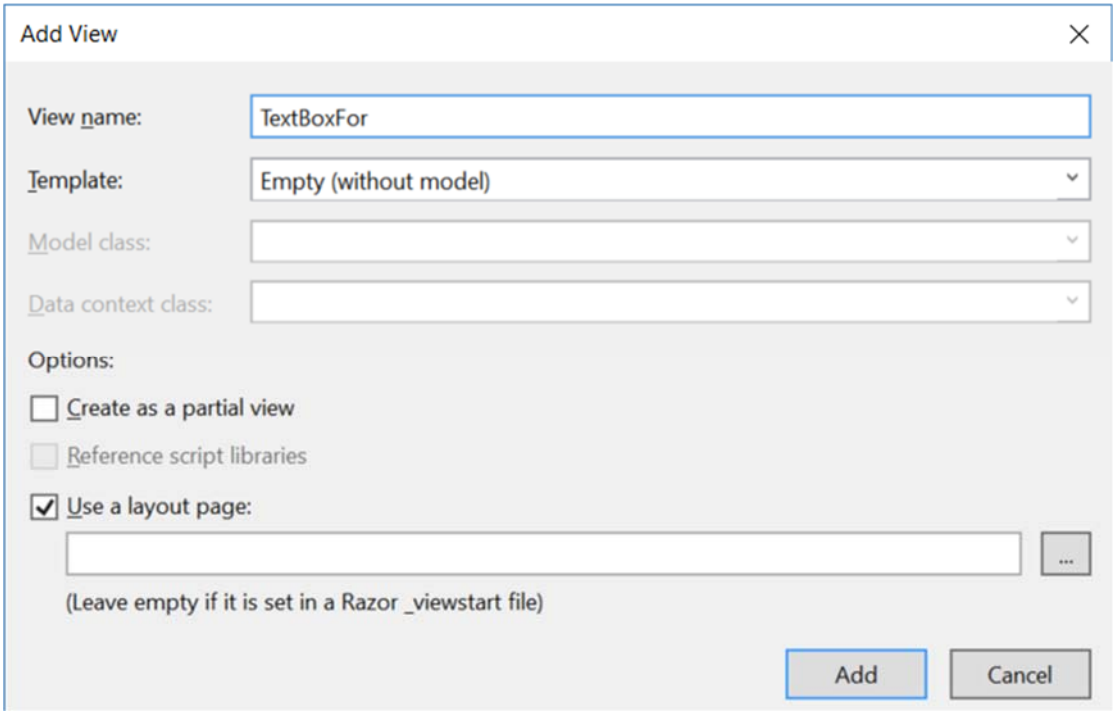
```
        {
            get
            {
                using (OnlineGameContext db = new OnlineGameContext())
                {
                    return db.Teams.ToListAsync().Result;
                }
            }
        }
        public List<SingleSelect> SingleSelectItems
        {
            get
            {
                using (OnlineGameContext db = new OnlineGameContext())
                {
                    //Get List
                    List<SingleSelect> singleSelectItems = db.SingleSelects.ToListAsync().Result;
                    //Set Property
                    SelectedItem = singleSelectItems.Single(item => item.IsSelected);
                    //Return List
                    return singleSelectItems;
                }
            }
        }
        //Constructor
        public Game(string name)
        {
            Name = name;
        }
        public Game()
        {
        }
    }
}
```

4.6. Views/Gamers/TextBox.cshtml



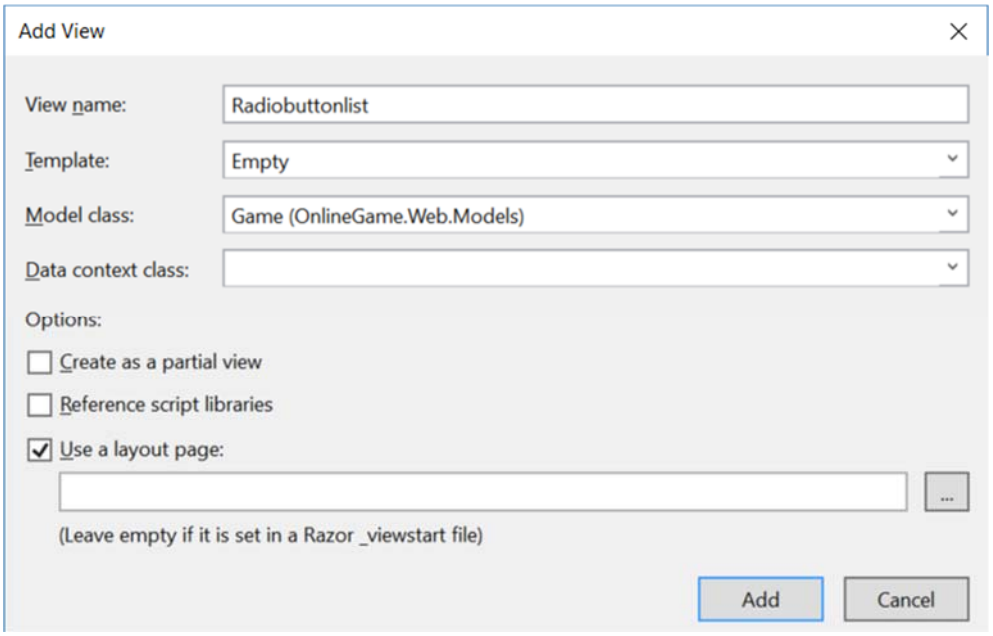

```
@{
    ViewBag.Title = "TextBox";
}
<h2>TextBox</h2>
@Html.TextBox("GameName", (string)ViewBag.GameName)
<br />
@Html.DropDownList("GameTeams", "Select Team")
@*
1.
//Game game = new Game("GameA");
//ViewBag.GameName = game.Name;
//ViewBag.GameTeams = new SelectList(game.Teams, "Id", "Name");
1.1.
//@Html.TextBox("GameName", (string)ViewBag.GameName)
It will create the following.
//<input id="Name" name="Name" type="text" value="GameA">
1.2.
//@Html.DropDownList("GameTeams", "Select Team")
It will create the following.
//<select id="Teams" name="Teams"><option value="">Select Team</option>
//    <option value="1">Team1</option>
//    <option value="2">Team2</option>
//    <option value="3">Team3</option>
//</select>
*@
```

4.7. Views/Gamers/TextBoxFor.cshtml

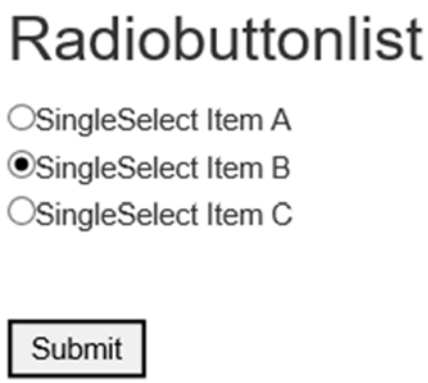



```
@using OnlineGame.Web.Models;
@model Game
@{
    ViewBag.Title = "TextBoxFor";
}
<h2>TextBoxFor</h2>
@Html.TextBoxFor(model => model.Name)
<br />
@Html.DropDownListFor(model => model.Teams, new SelectList(Model.Teams, "Id", "Name"), "Select Team")
@*
1.
@Html.TextBox and @Html.DropDownList are NOT strongly typed which does not need the view Model.
@Html.TextBoxFor and @Html.DropDownListFor are strongly typed which need the view Model.
The result of produced HTML is the same,
but Strongly typed HTML helpers support compile time error checking.
So Strongly typed HTML helpers are better.
1.1.
//@Html.TextBoxFor(model => model.Name)
It will create the following.
//<input id="Name" name="Name" type="text" value="GameA">
1.2.
//@Html.DropDownListFor(model => model.Teams, new SelectList(Model.Teams, "Id", "Name"), "Select Team")
It will create the following.
//<select id="Teams" name="Teams"><option value="">Select Team</option>
//    <option value="1">Team1</option>
//    <option value="2">Team2</option>
//    <option value="3">Team3</option>
//</select>
*@
```

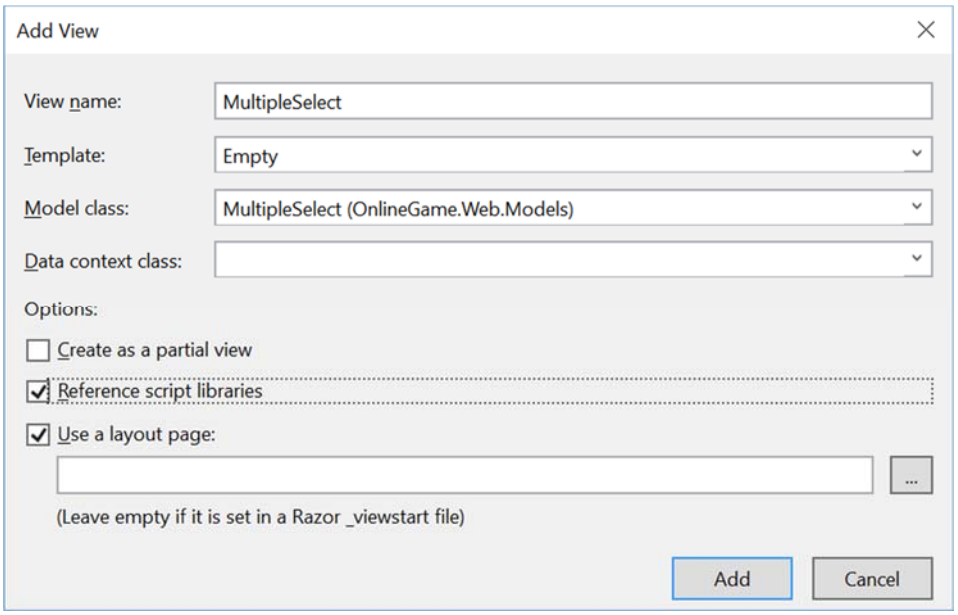
4.8. Views/Gamers/Radiobuttonlist.cshtml




```
@using OnlineGame.Web.Models
@model Game
@{
    ViewBag.Title = "Radiobuttonlist";
}
<h2>Radiobuttonlist</h2>
@using (Html.BeginForm("Radiobuttonlist", "Gamers"))
{
    foreach (SingleSelect item in Model.SingleSelectItems)
    {
        @(item.IsSelected ?
            Html.RadioButtonFor(model => model.SelectedItemId, item.Id, new {@checked = "checked"}) :
            Html.RadioButtonFor(model => model.SelectedItemId, item.Id))
        @*@Html.RadioButtonFor(model => model.SelectedItemId, item.Id)*@
        @item.Name<br />
    }
    <br />
    <br />
    <input type="submit" value="Submit" />
}
@*
//@(item.IsSelected ?
//    Html.RadioButtonFor(model => model.SelectedItemId, item.Id, new {@checked = "checked"}) :
//    Html.RadioButtonFor(model => model.SelectedItemId, item.Id))
It will create the following.
//<input id="SelectedItemId" name="SelectedItemId" type="radio" value="1">SingleSelect Item A<br>
//<input checked="checked" id="SelectedItemId" name="SelectedItemId" type="radio" value="2">SingleSelect
Item B<br>
//<input id="SelectedItemId" name="SelectedItemId" type="radio" value="3">SingleSelect Item C<br>
*@
```

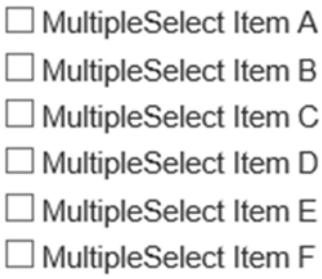


4.9. Views/Gamers/EditorTemplates/MultipleSelect.cshtml

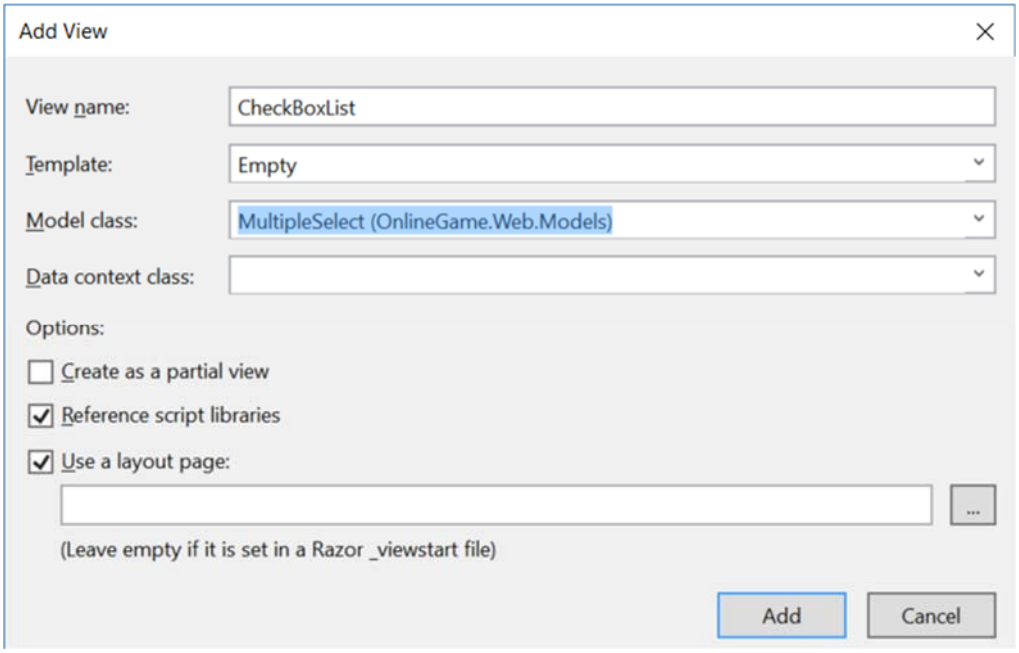



```
@using OnlineGame.Web.Models
@model MultipleSelect
@{
    ViewBag.Title = "MultipleSelect";
}
@Html.HiddenFor(model => model.Id)
@Html.HiddenFor(model => model.Name)
@Html.CheckBoxFor(model => model.IsSelected)
@Html.DisplayFor(model => model.Name)<br />
@*
1.
1.1.
//@Html.HiddenFor(model => model.Id)
It will create the following.
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[0].Id" type="hidden" value="1">
------------------
1.2.
//@Html.HiddenFor(model => model.Name)
It will create the following.
//<input name="[0].Name" type="hidden" value="MultipleSelect Item A">
------------------
1.3.
//@Html.CheckBoxFor(model => model.IsSelected)
```

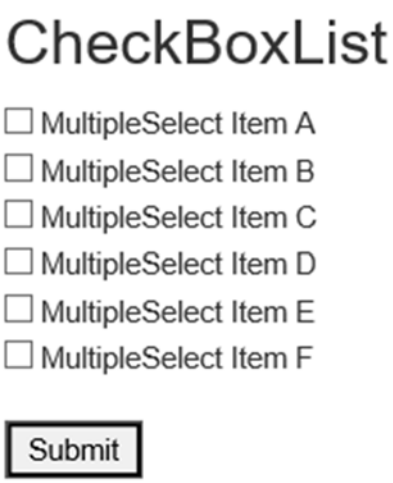
```
//@Html.DisplayFor(model => model.Name)<br />
It will create the following.
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[0].Id" type="hidden" value="1" />
//<input name="[0].Name" type="hidden" value="MultipleSelect Item A" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[0].IsSelected"
type="checkbox" value="true" /><input name="[0].IsSelected" type="hidden" value="false" />
//MultipleSelect Item A<br />
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[1].Id" type="hidden" value="2" />
//<input name="[1].Name" type="hidden" value="MultipleSelect Item B" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[1].IsSelected"
type="checkbox" value="true" /><input name="[1].IsSelected" type="hidden" value="false" />
//MultipleSelect Item B<br />
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[2].Id" type="hidden" value="3" />
//<input name="[2].Name" type="hidden" value="MultipleSelect Item C" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[2].IsSelected"
type="checkbox" value="true" /><input name="[2].IsSelected" type="hidden" value="false" />
//MultipleSelect Item C<br />
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[3].Id" type="hidden" value="4" />
//<input name="[3].Name" type="hidden" value="MultipleSelect Item D" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[3].IsSelected"
type="checkbox" value="true" /><input name="[3].IsSelected" type="hidden" value="false" />
//MultipleSelect Item D<br />
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[4].Id" type="hidden" value="5" />
//<input name="[4].Name" type="hidden" value="MultipleSelect Item E" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[4].IsSelected"
type="checkbox" value="true" /><input name="[4].IsSelected" type="hidden" value="false" />
//MultipleSelect Item E<br />
//<input data-val="true" data-val-number="The field Id must be a number." data-val-required="The Id field
is required." name="[5].Id" type="hidden" value="6" />
//<input name="[5].Name" type="hidden" value="MultipleSelect Item F" />
//<input data-val="true" data-val-required="The IsSelected field is required." name="[5].IsSelected"
type="checkbox" value="true" /><input name="[5].IsSelected" type="hidden" value="false" />
//MultipleSelect Item F<br />
*@
```



4.10. Views/Gamers/CheckBoxList.cshtml



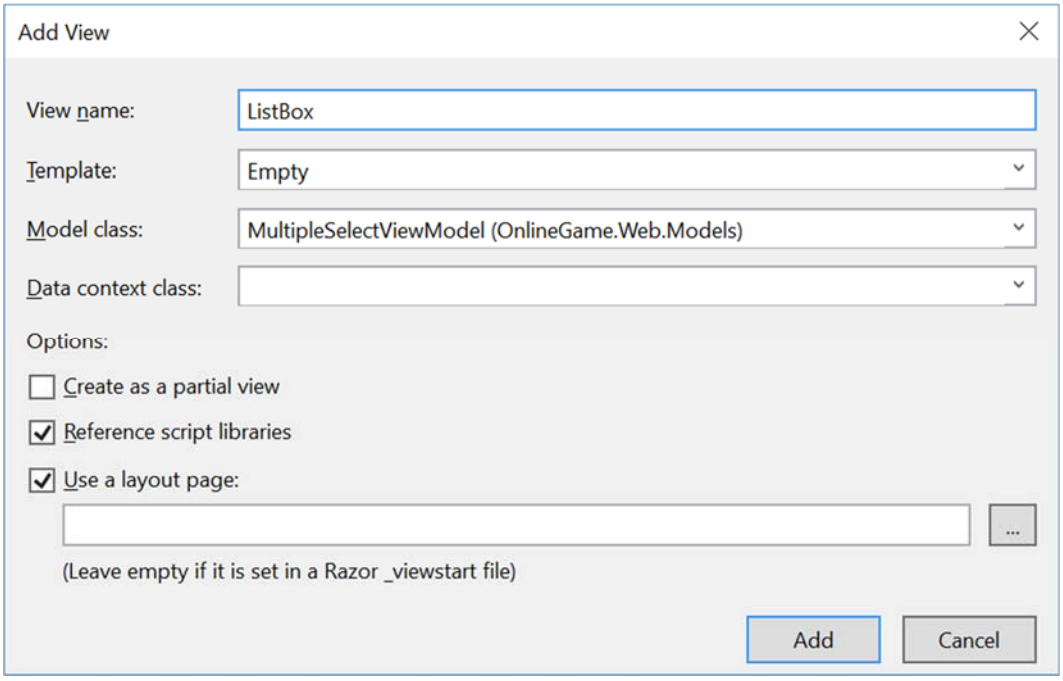

```
@using OnlineGame.Web.Models
@model List<MultipleSelect>
@{
    ViewBag.Title = "CheckBoxList";
}
<div style="font-family:Arial">
    <h2>CheckBoxList</h2>
    @using (Html.BeginForm("CheckBoxList", "Gamers"))
    {
        @Html.EditorForModel()
        <br />
        <input type="submit" value="Submit" />
    }
</div>
@*
By MVC convention,
@Html.EditorForModel() function is going to look for
a cshtml view template with the same name
as the current IEnumerable Model item type.
In this case, the Model is List<MultipleSelect>.
The IEnumerable Model item type is MultipleSelect.
The EditorForModel() will look for MultipleSelect.cshtml.
The Model of MultipleSelect.cshtml must be MultipleSelect type.
*@
```



4.11. Models/Gamers/MultipleSelectViewModel.cs

```
using System.Collections.Generic;
using System.Web.Mvc;
namespace OnlineGame.Web.Models
{
    public class MultipleSelectViewModel
    {
        public IEnumerable<string> SelectedItemIds { get; set; }
        public IEnumerable<SelectListItem> MultipleSelectItems { get; set; }
    }
}
```

4.12. Views/Gamers/ListBox.cshtml




```
@using OnlineGame.Web.Models
@model MultipleSelectViewModel
@{
    ViewBag.Title = "ListBox";
}
<h2>ListBox</h2>
@using (Html.BeginForm())
{
    @Html.ListBoxFor(model => model.SelectedItemIds, Model.MultipleSelectItems, new { size = 5 })
    <br />
    <input type="submit" value="Submit" />
}
@*
@Html.ListBoxFor(model => model.SelectedItemIds, Model.MultipleSelectItems, new { size = 5 })
It will create the following.
<select id="SelectedItemIds" multiple="multiple" name="SelectedItemIds" size="5">
```

```
    <option value="1">MultipleSelect Item A</option>
    <option value="2">MultipleSelect Item B</option>
    <option value="3">MultipleSelect Item C</option>
    <option value="4">MultipleSelect Item D</option>
    <option value="5">MultipleSelect Item E</option>
    <option value="6">MultipleSelect Item F</option>
</select>
*@
```