(T13)實做類似 GridView、SearchBar(搜尋引擎)、Pagging(分頁)、Sorting(排序)、DeleteMultipleRows(執行多筆移除)、SelectAll,UnSelectAll(全選,全不選)
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924

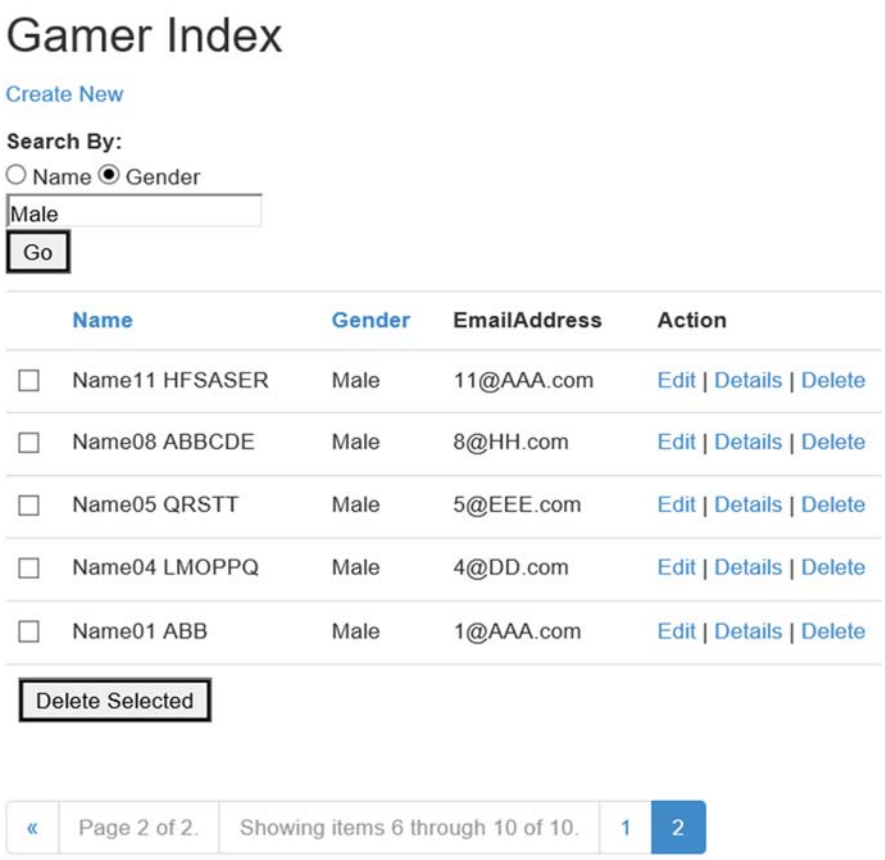
(T13)實做類似 GridView、SearchBar(搜尋引擎)、Pagging(分頁)、Sorting(排序)、DeleteMultipleRows(執行多筆移除)、SelectAll,UnSelectAll(全選,全不選)
(T13-1)實做類似 GridView、SearchBar(搜尋引擎) (1. to 4.)
(T13-2)實做類似 GridView、Pagging(分頁) (5.)
(T13-3)實做類似 GridView、Sorting(排序) (6.)
(T13-4)實做類似 GridView、DeleteMultipleRows(執行多筆移除) (7.)
(T13-5)實做類似 GridView、SelectAll,UnSelectAll(全選,全不選) (8.)

0. What to learn
---
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
2.1.3. Web.config
---
3. OnlineGame.Web
3.1. ADO.Net Entity Data Model - Entity Framework
3.2. Controllers/GamersController.cs
---
4. OnlineGame.Web - Search Bar
4.1. Views/Gamer/Index.cshtml
4.2. Controllers/GamerController.cs
4.3. Run Search Bar
---
5. OnlineGame.Web - PagedList, PagedList.Mvc
5.1. Install NuGet Package
5.2. Controllers/GamerController.cs
5.3. Views/Gamer/Index.cshtml
5.4. Run Search Bar with paging
---
6. OnlineGame.Web - Sorting
6.1. Controllers/GamerController.cs
6.2. Views/Gamer/Index.cshtml
6.3. Run Search Bar, paging, and Sorting
---
7. OnlineGame.Web - Check box delete All
7.1. Controllers/GamerController.cs
7.2. Views/Gamer/Index.cshtml
7.3. Run Search Bar, paging, and Sorting
---
8. OnlineGame.Web - Check box delete All
8.1. Views/Gamer/Index.cshtml

0. What to learn

* 哪尼?手寫搜尋引擎 Search Bar 搭配 MVC 實現有分頁的 GridView，還可 Sorting，可全選取 On/Off 並執行多筆移除。
* 完全手寫一個 Search Bar 搜尋引擎。
* 使用 MVC 實現像是 Web Form 的無敵 GridView。
* 實現 GridView 分業，排序，可全選 On/Off，並執行多筆移除。

其他參考資料
MVC 這裡我教了分頁
http://kevintsengtw.blogspot.com/2014/11/pagedlistmvc-pager.html
請把這個當作補充教材，這邊是把 pagelist 中文化



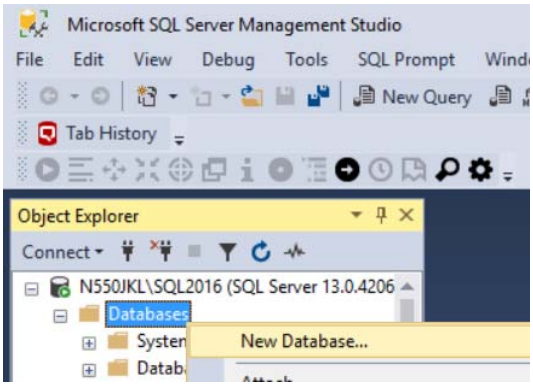
1. OnlineGame DB

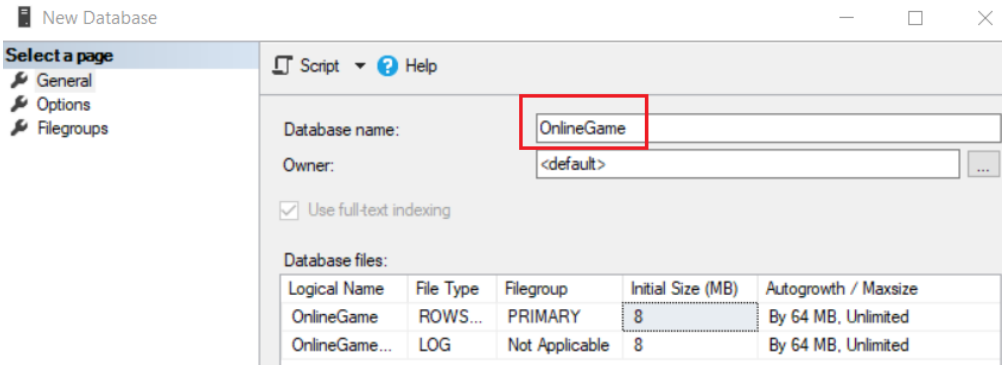
1.1. TSQL

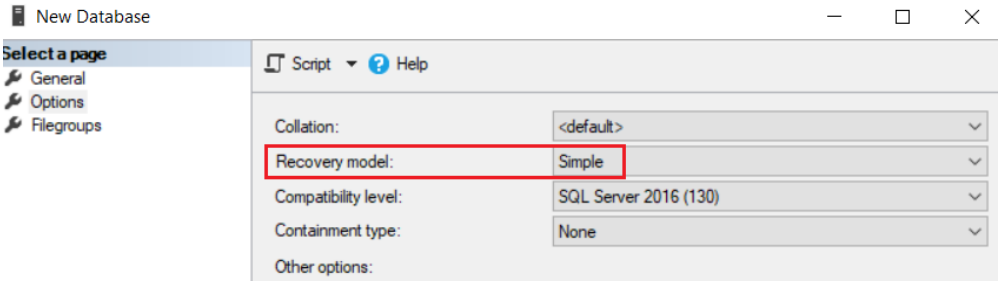
In SQL server Management Studio (SSMS)
Database --> Right Click --> New Database -->

In General Tab -->
Name: **OnlineGame**

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








```
--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
--2. Create Table
CREATE TABLE Gamer
(
   Id INT PRIMARY KEY
            IDENTITY(1, 1)
```

```
            NOT NULL ,
    [Name] NVARCHAR(100) NOT NULL ,
    Gender NVARCHAR(10) NOT NULL,
    EmailAddress nvarchar(100) NOT NULL,
)
--3. Insert Data
INSERT  Gamer
VALUES  ( N'Name01 ABB', N'Male', '1@AAA.com');
INSERT  Gamer
VALUES  ( N'Name02 CDDE', N'Female', '2@BBB.com');
INSERT  Gamer
VALUES  ( N'Name03 FIJK', N'Female', '3@CCCC.com');
INSERT  Gamer
VALUES  ( N'Name04 LMOPPQ', N'Male', '4@DD.com');
INSERT  Gamer
VALUES  ( N'Name05 QRSTT', N'Male', '5@EEE.com');
INSERT  Gamer
VALUES  ( N'Name06 TUVVX', N'Female', '6@FF.com');
INSERT  Gamer
VALUES  ( N'Name07 XYZZXX', N'Female', '7@GGGG.com');
INSERT  Gamer
VALUES  ( N'Name08 ABBCDE', N'Male', '8@HH.com');
INSERT  Gamer
VALUES  ( N'Name09 QRSTTUVXX', N'Male', '9@IIII.com');
INSERT  Gamer
VALUES  ( N'Name10 GGAAEE', N'Male', '10@XXWFFS.com');
INSERT  Gamer
VALUES  ( N'Name11 HFSASER', N'Male', '11@AAA.com');
INSERT  Gamer
VALUES  ( N'Name12 ESVSADC', N'Female', '12@BBB.com');
INSERT  Gamer
VALUES  ( N'Name13 REDSVF', N'Female', '13@CCCC.com');
INSERT  Gamer
VALUES  ( N'Name14 BBGVDD', N'Male', '14@DD.com');
INSERT  Gamer
VALUES  ( N'Name15 WWVFSSQ', N'Male', '15@EEE.com');
INSERT  Gamer
VALUES  ( N'Name16 TTVSS', N'Female', '16@FF.com');
INSERT  Gamer
VALUES  ( N'Name17 AAQERR', N'Female', '17@GGGG.com');
INSERT  Gamer
VALUES  ( N'Name18 BBFSAQ', N'Male', '18@HH.com');
INSERT  Gamer
VALUES  ( N'Name19 QRSTTUVXX', N'Male', '19@IIII.com');
INSERT  Gamer
VALUES  ( N'Name20 HHFWSWQ', N'Male', '20@XXWFFS.com');
GO -- Run the previous command and begins new batch
```

1.2. Security login

In SQL server

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

-->

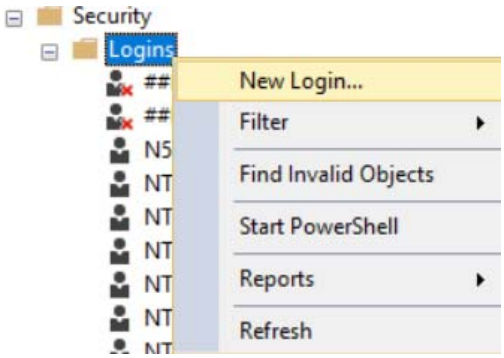
General Tab

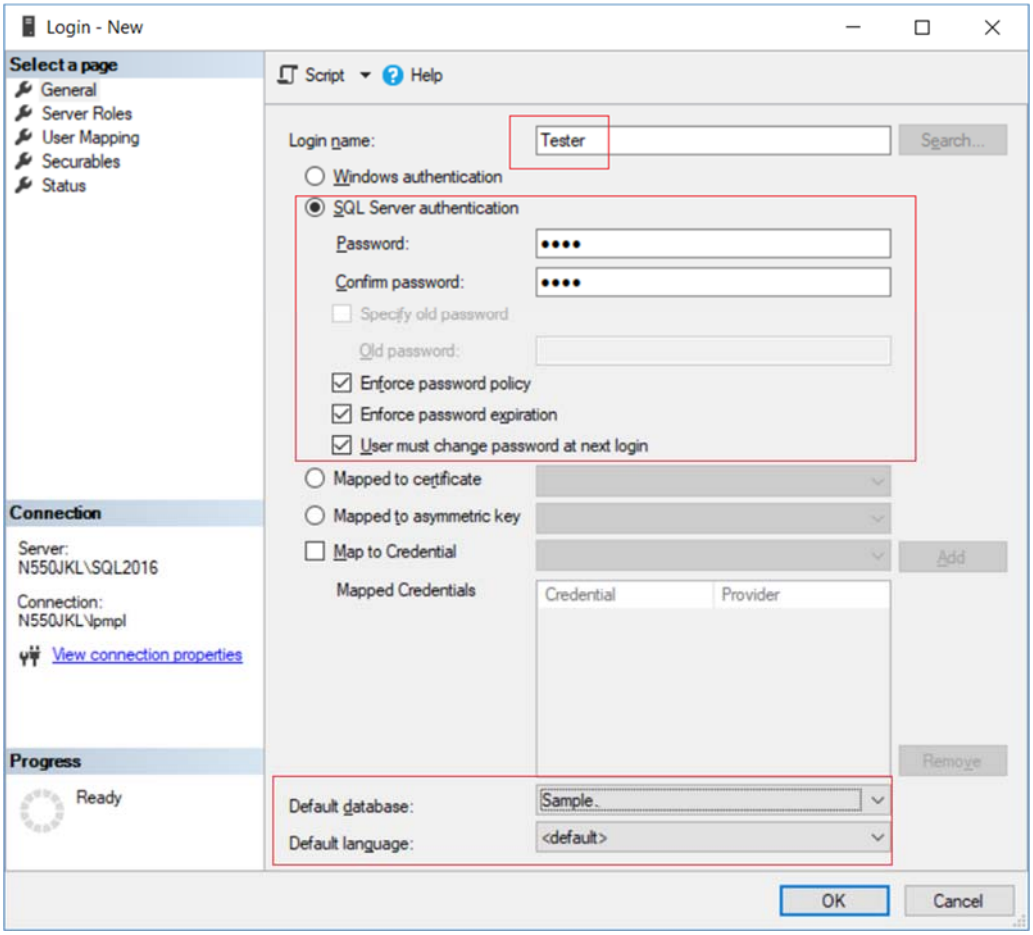
Login Name :

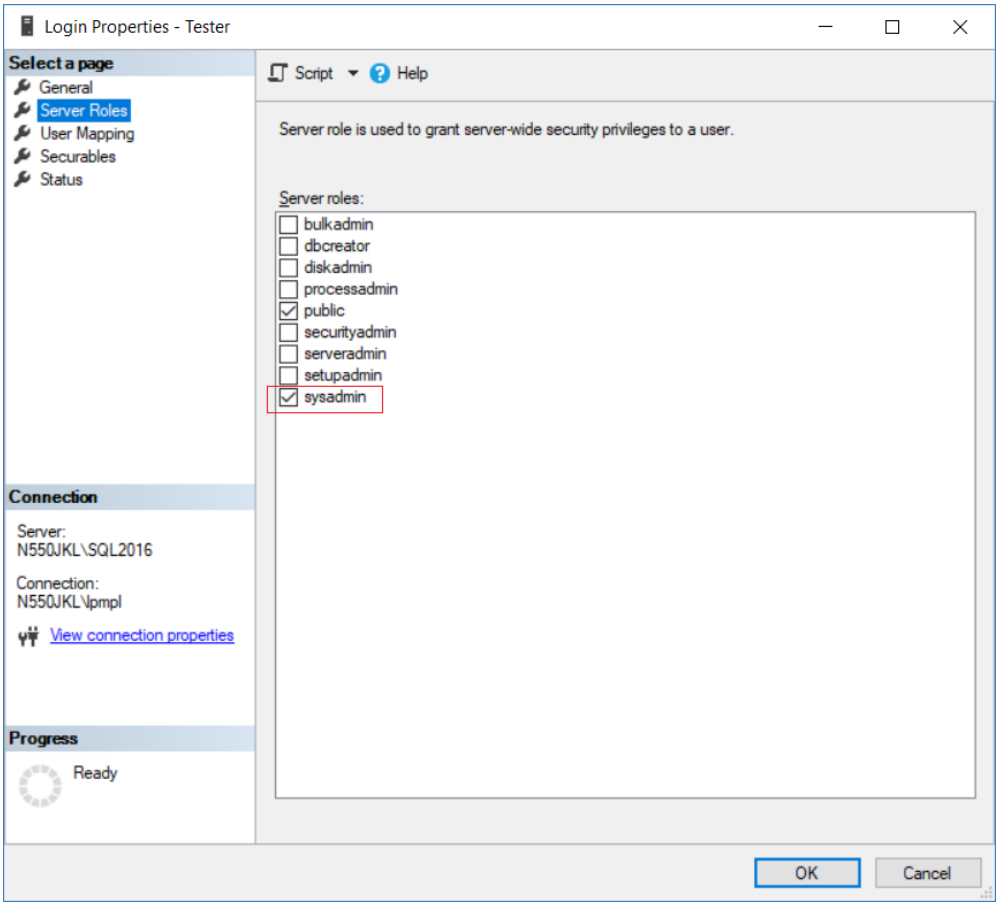
Tester

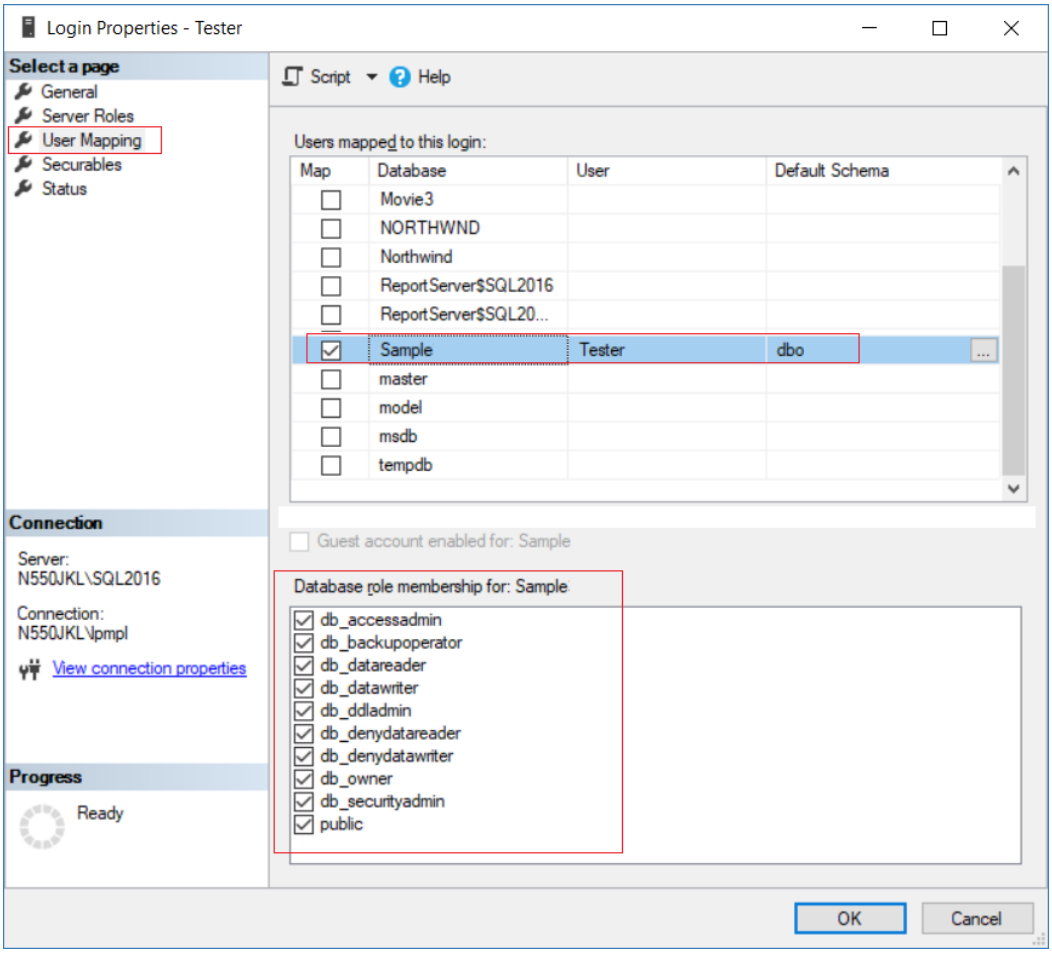
Password:

1234
Default Database:
OnlineGame
-->
Server Roles Tab
Select
sysadmin
-->
User Mapping Tab
Select **OnlineGame**
Select every single role.



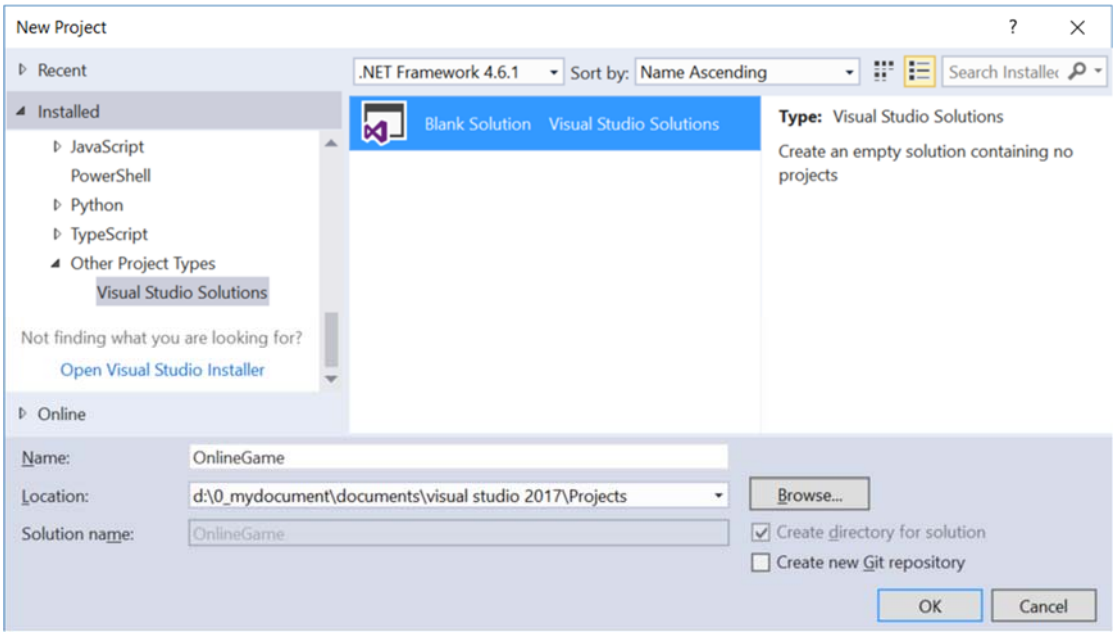



Login Properties - Tester
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Server role is used to grant server-wide security privileges to a user.
Server roles:
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connection
Server: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


Login Properties - Tester
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
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
View connection properties
Progress
Ready
OK
Cancel

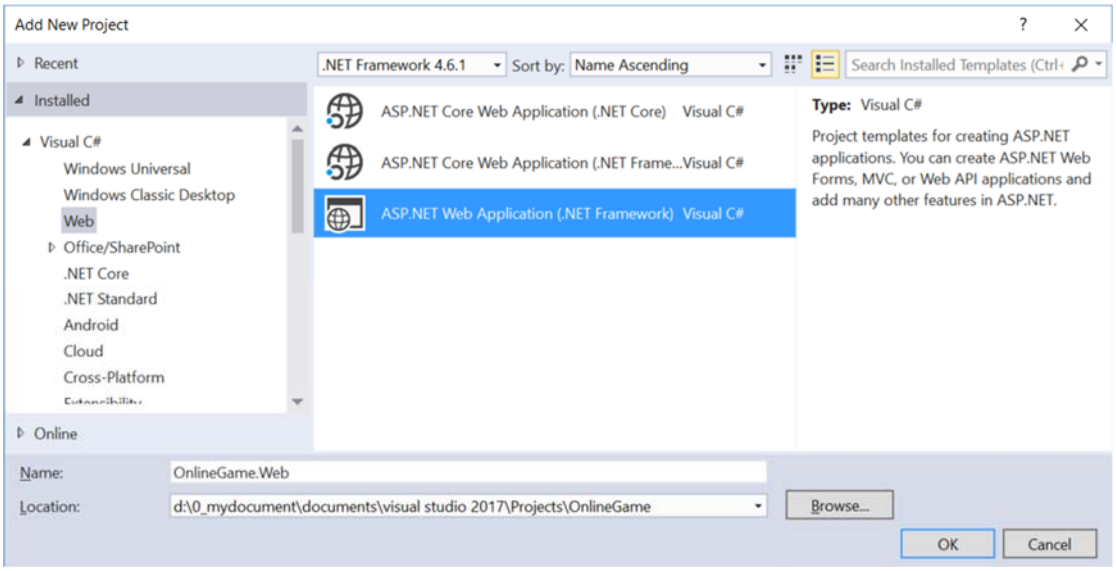
2. New Project - OnlineGame

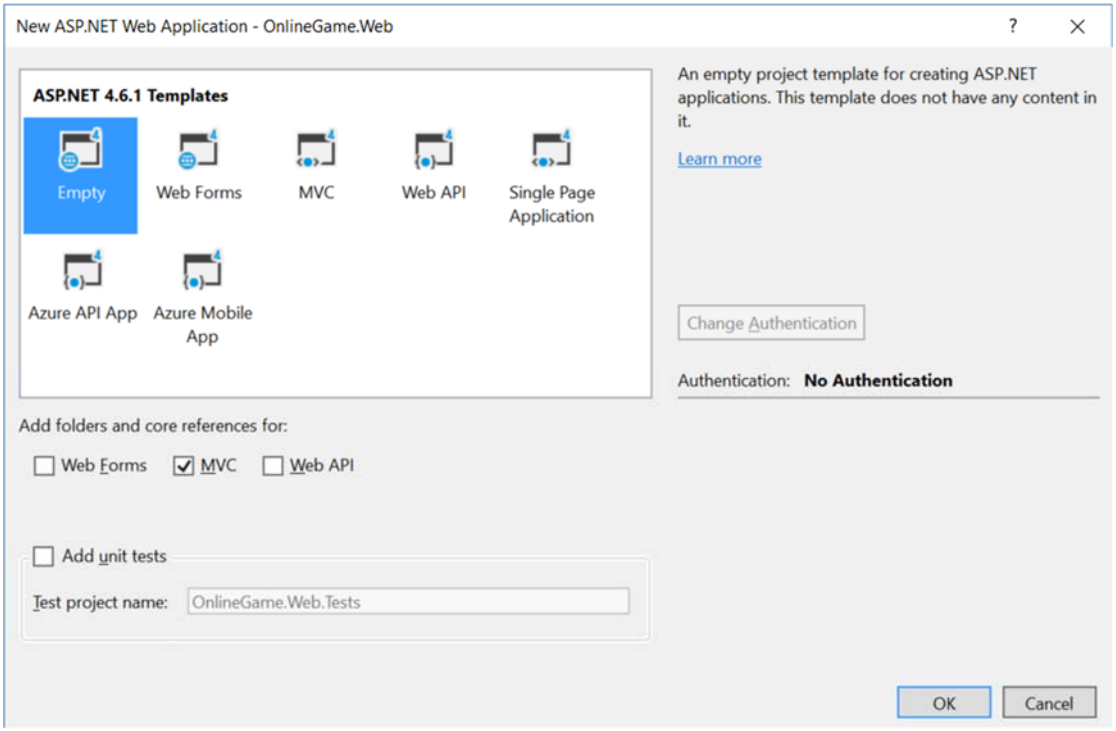
File --> New --> Project... -->
Other Project Types --> Visual Studio Solutions --> Blank Solution
-->
Name: **OnlineGame**



2.1. New Project - OnlineGame.Web

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





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);
        }
    }
}
```

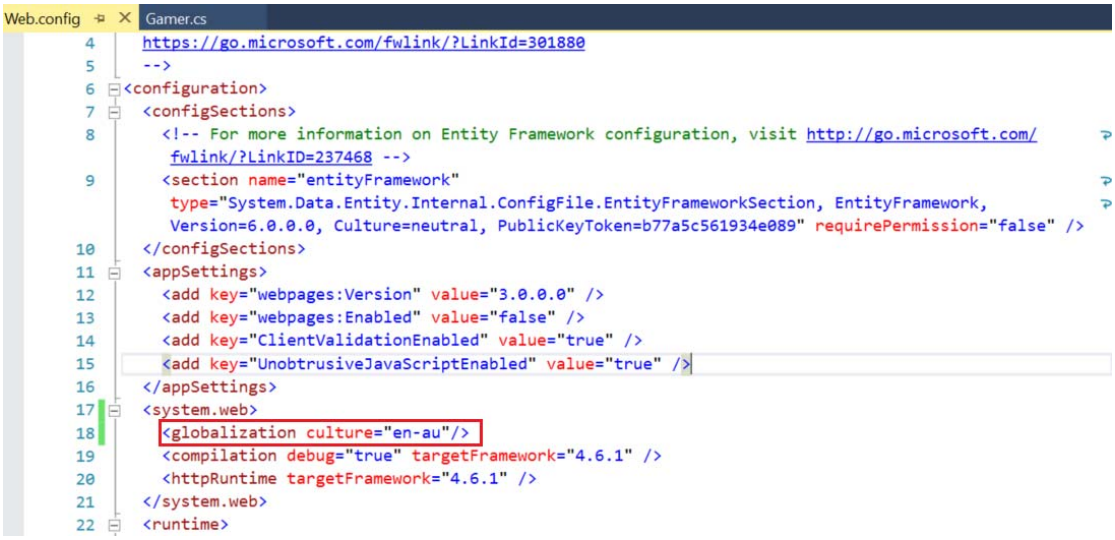
2.1.2. App_Start/RouteConfig.cs

```
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 = "Gamer", 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.
*/
```

2.1.3. Web.config




```
  <system.web>
    <globalization culture="en-au"/>
```

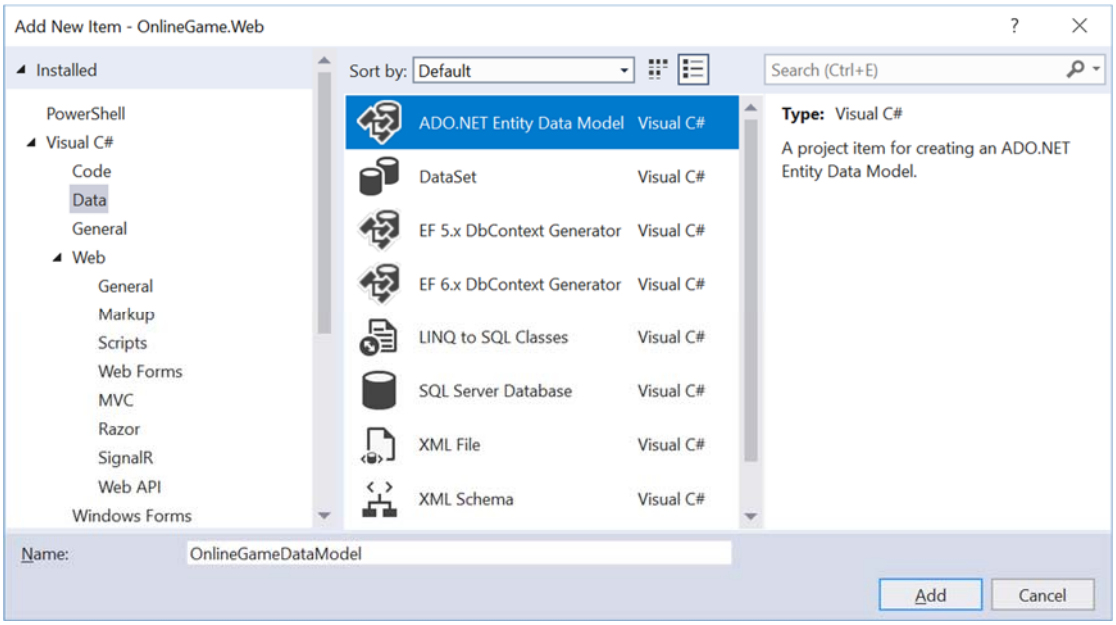
3. OnlineGame.Web

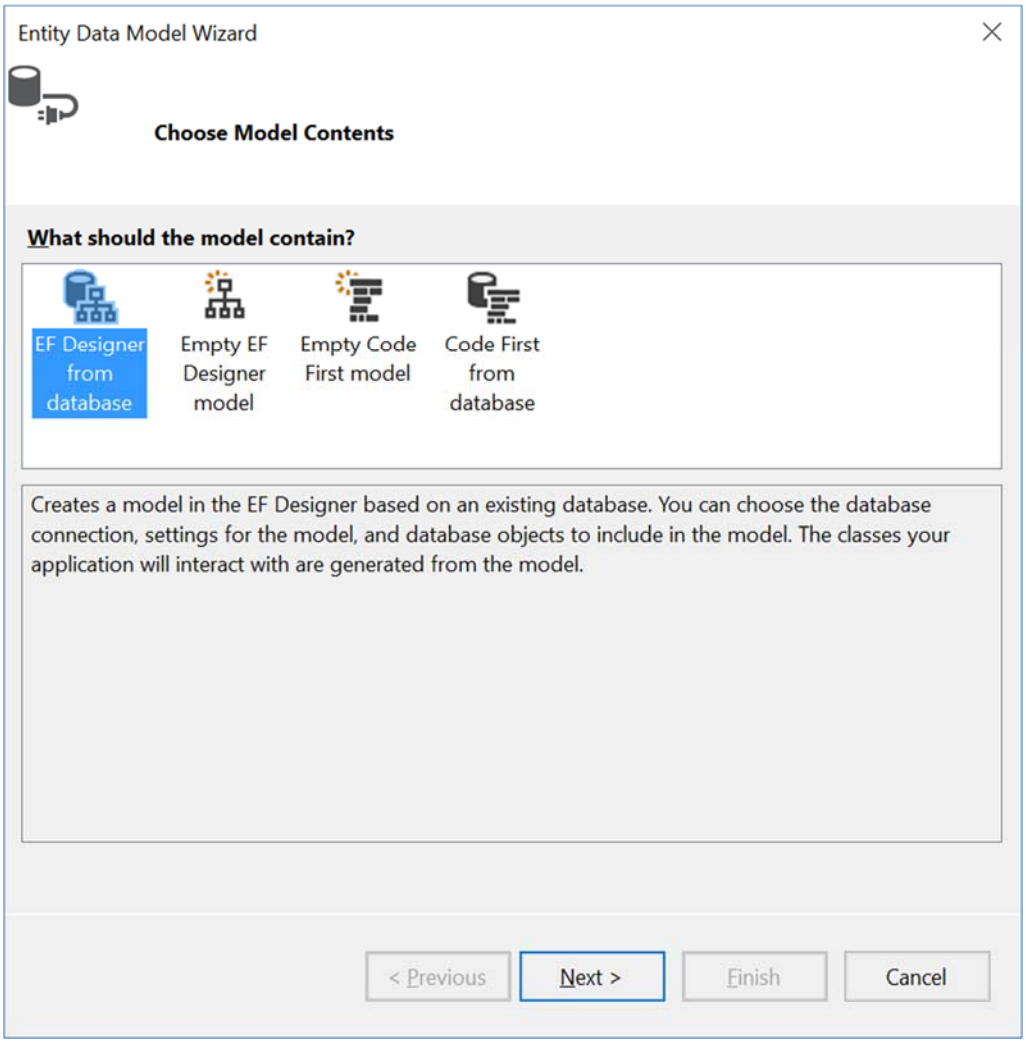
3.1. ADO.Net Entity Data Model - Entity Framework

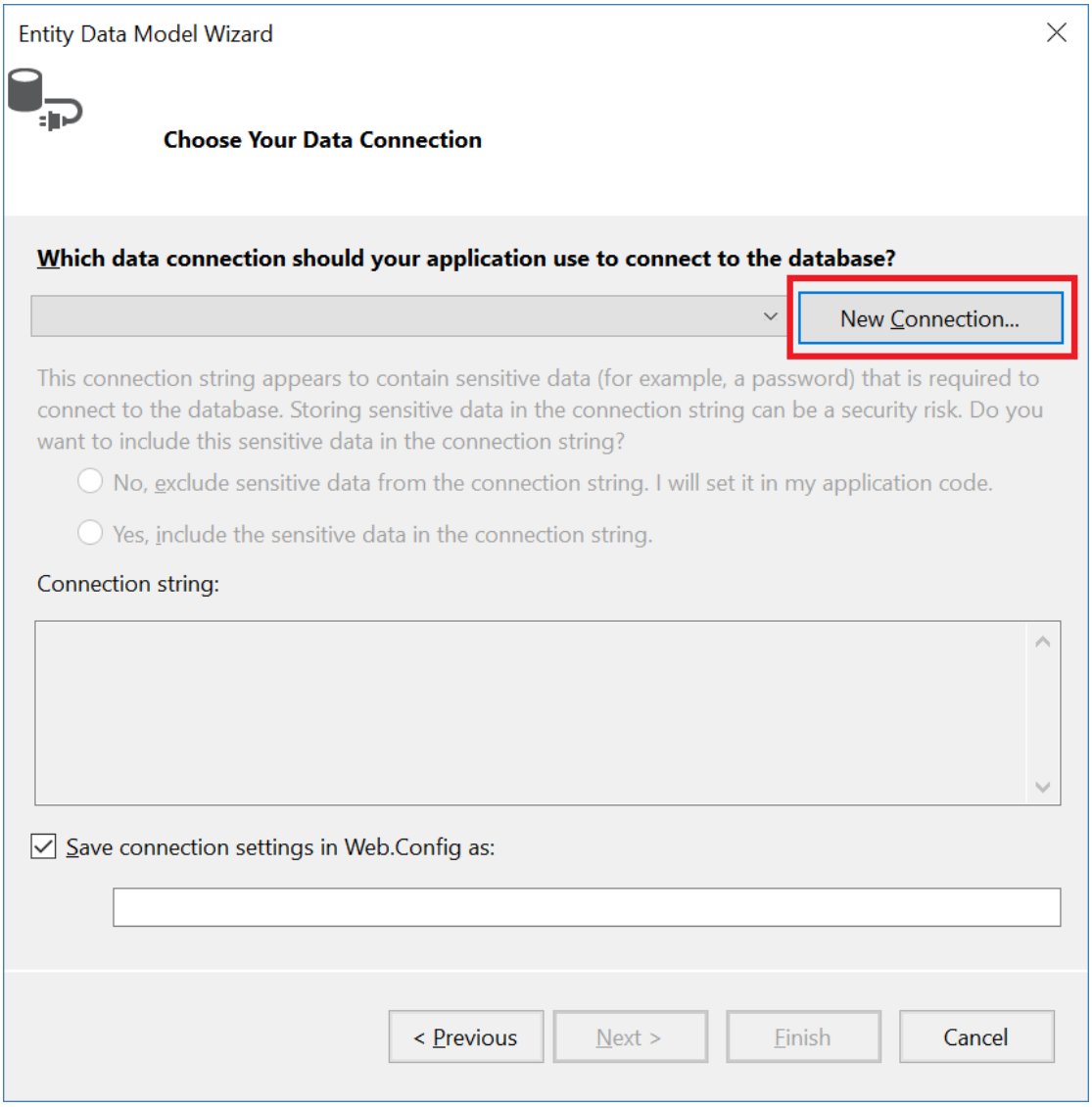
In Visual Studio 2017
Models folder --> Right Click --> Add --> New Item
--> Visual C# --> Data --> ADO.Net Entity Data Model
Name:
OnlineGameDataModel

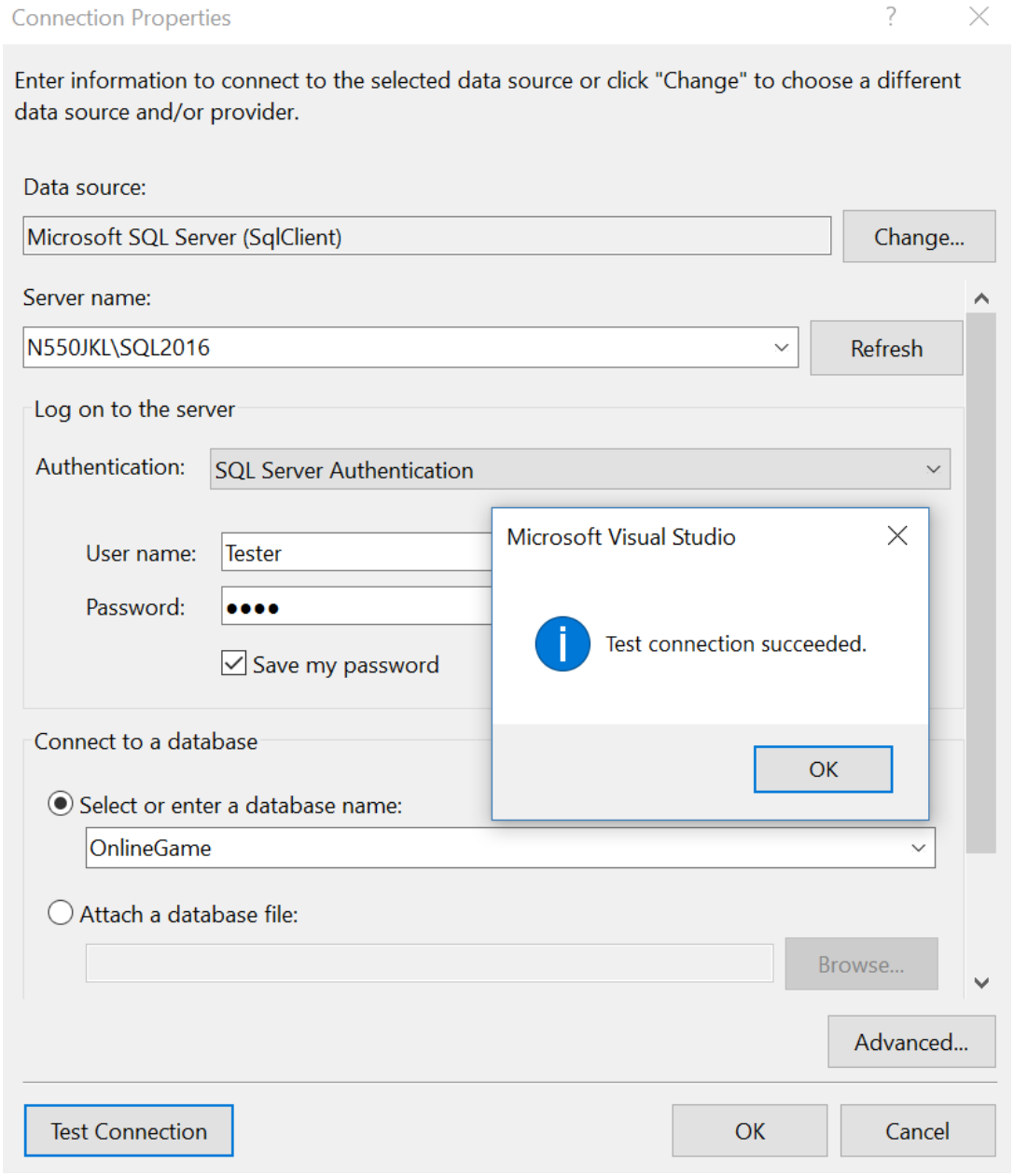
-->
EF Designer from database
....

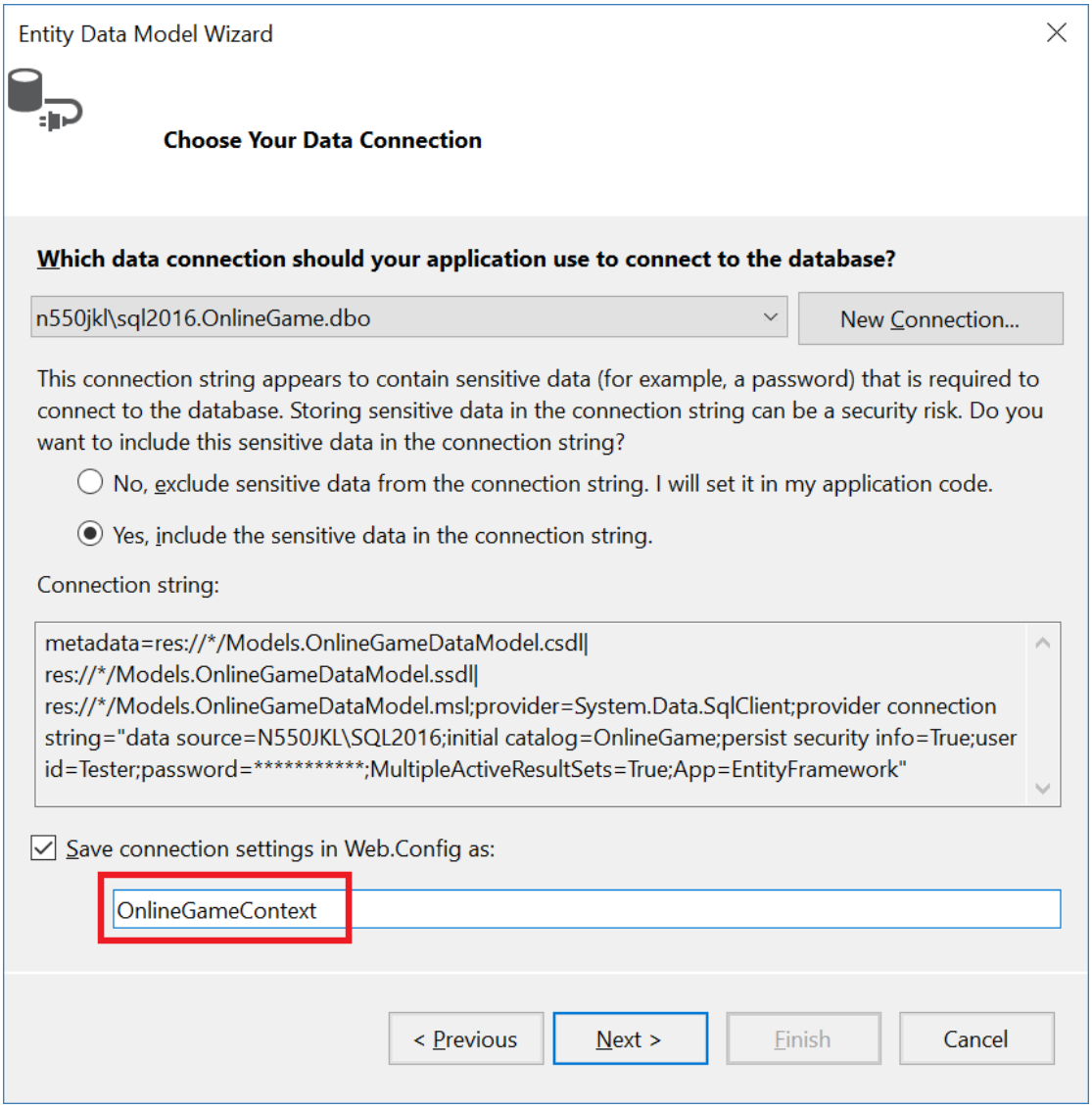
-->
Save Connection settings in Web.Config as:
OnlineGameContext

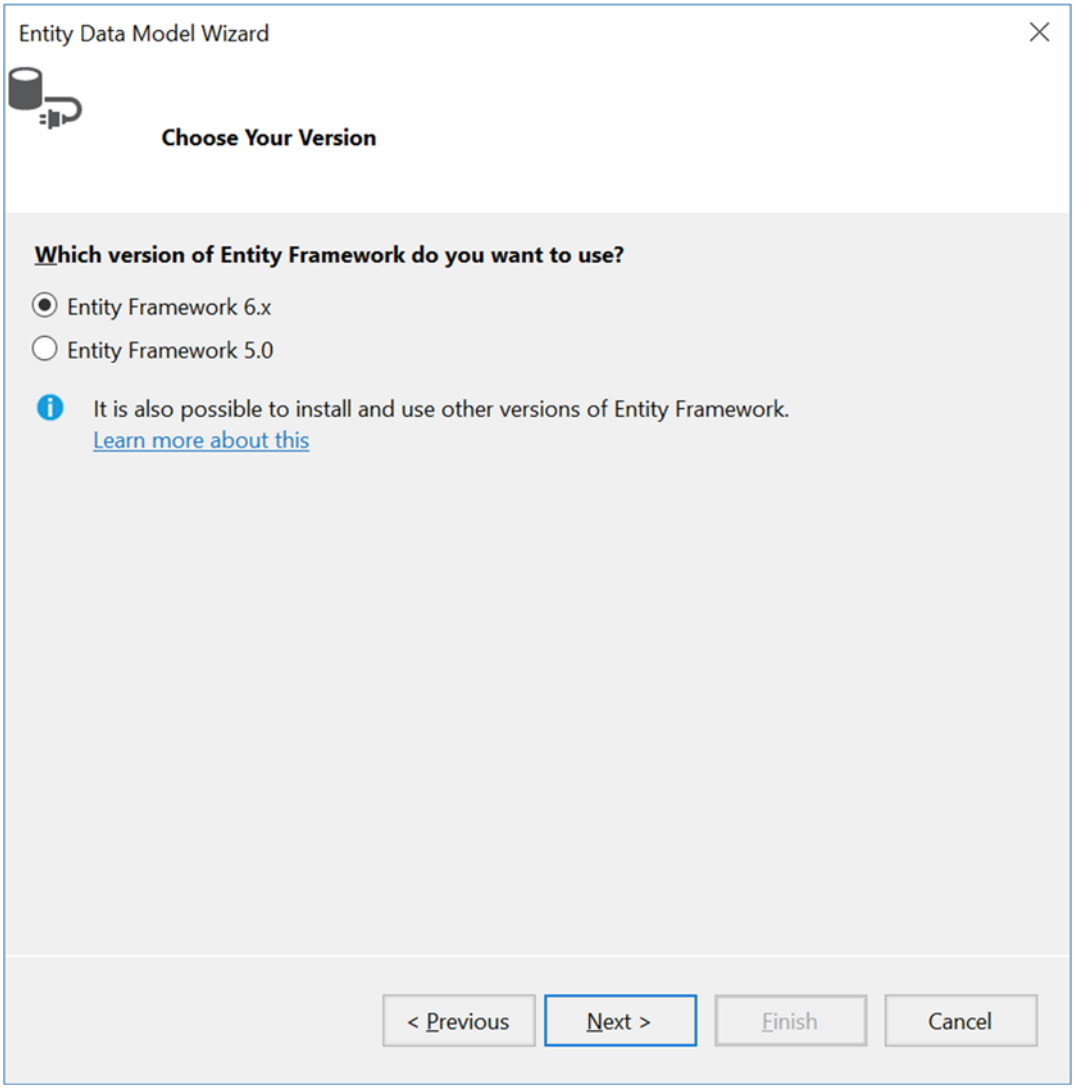

Add New Item - OnlineGame.Web
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Web API
Windows Forms
Sort by: Default
Search (Ctrl+E)
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator 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

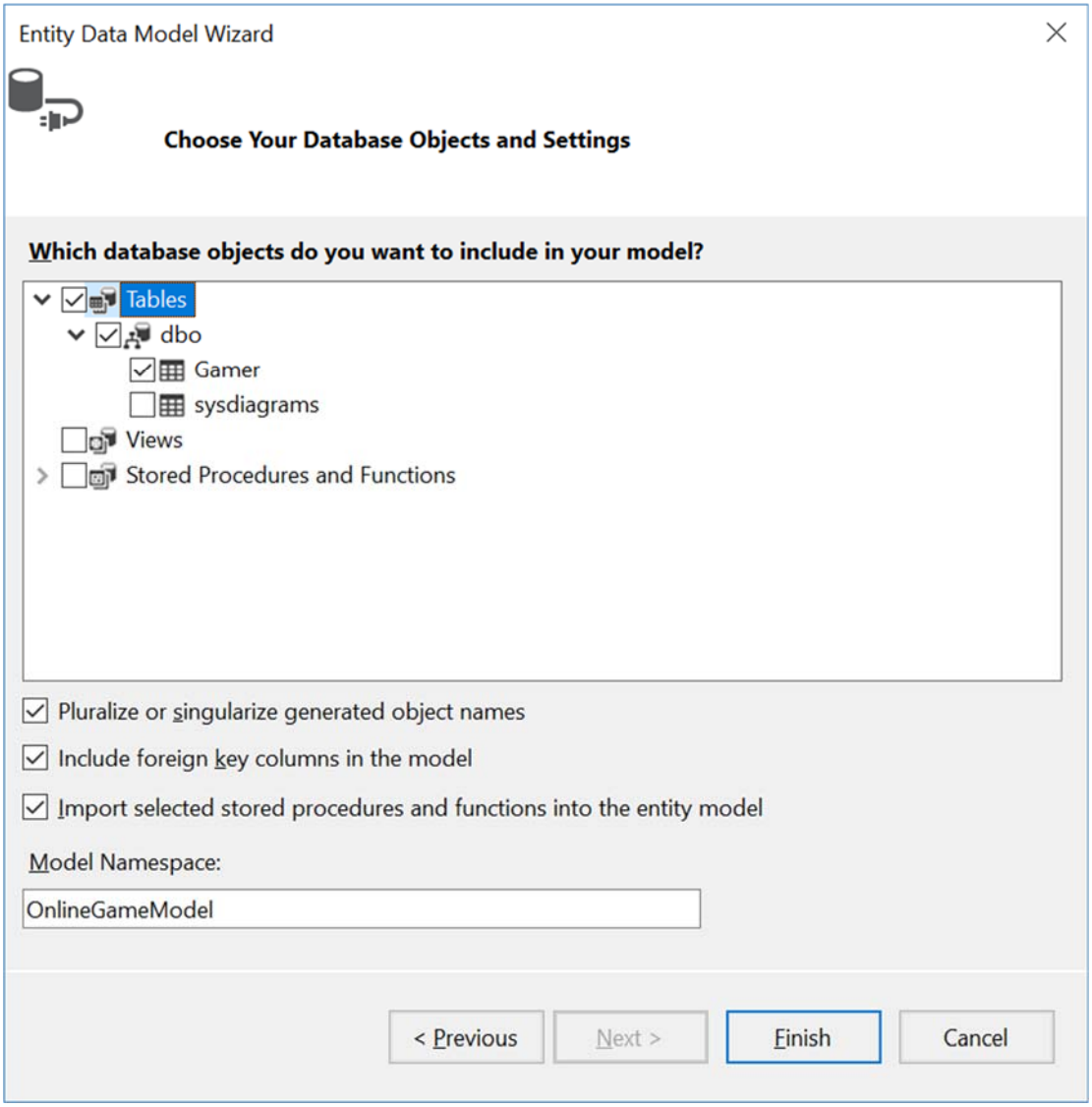

Entity Data Model Wizard
Choose Model Contents
What should the model contain?
EF Designer from database
Empty EF Designer model
Empty Code First model
Code First from database
Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model,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

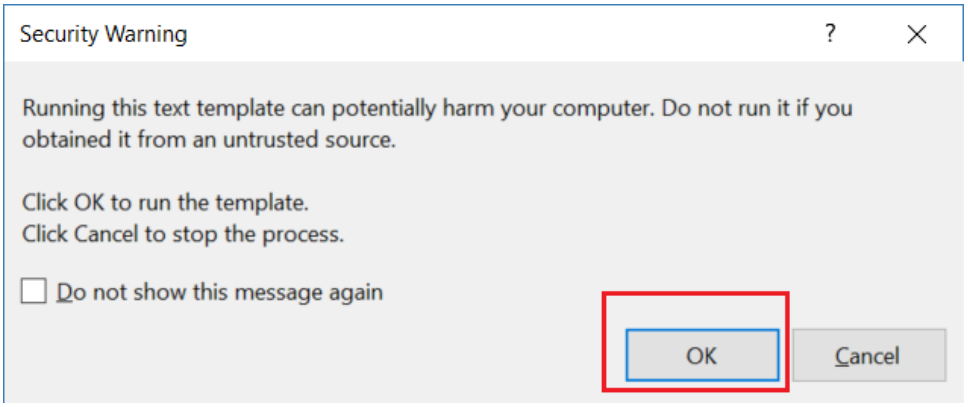

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in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

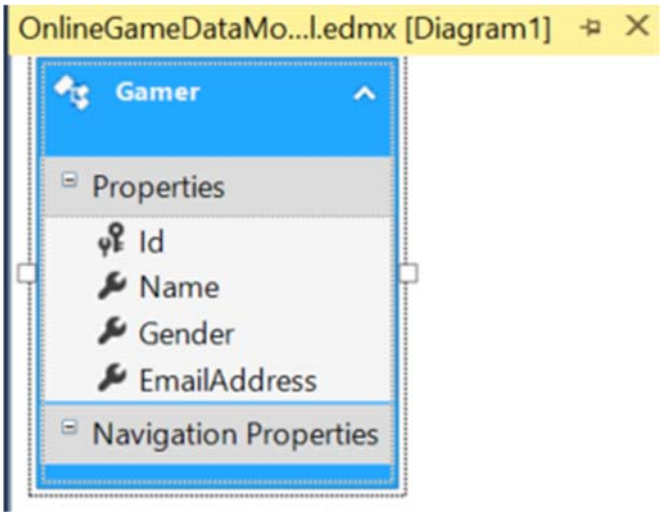

Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
N550JKL\SQL2016
Refresh
Log on to the server
Authentication:
SQL Server Authentication
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
sysdiagrams
Views
Stored Procedures and Functions
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

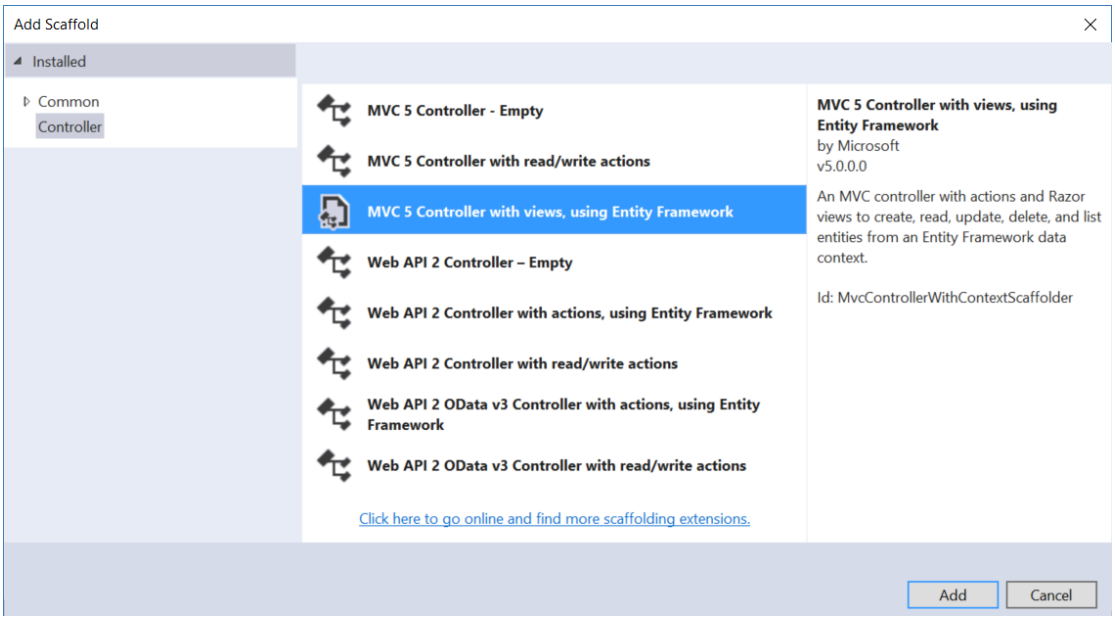


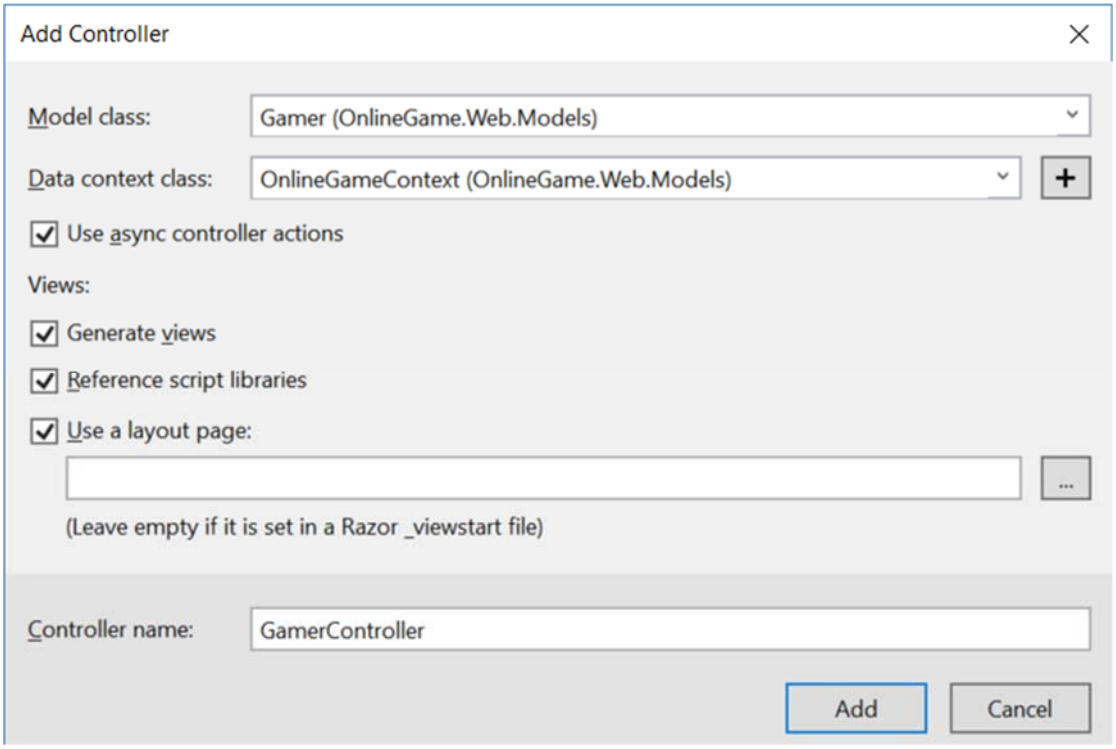
3.2. Controllers/GamersController.cs

Controllers --> Right click --> Add --> Controller

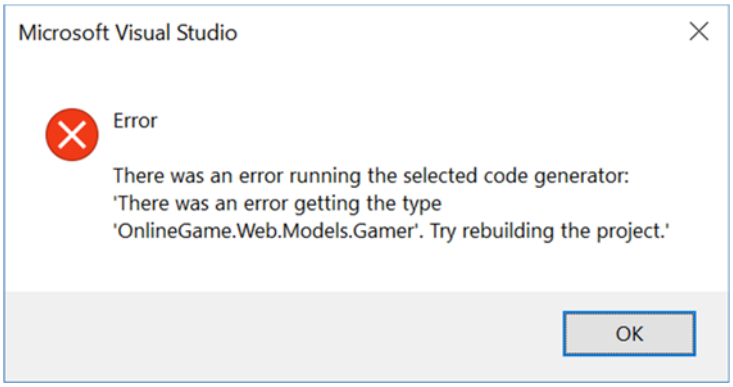
-->

MVC 5 Controller with views, using Entity Framework

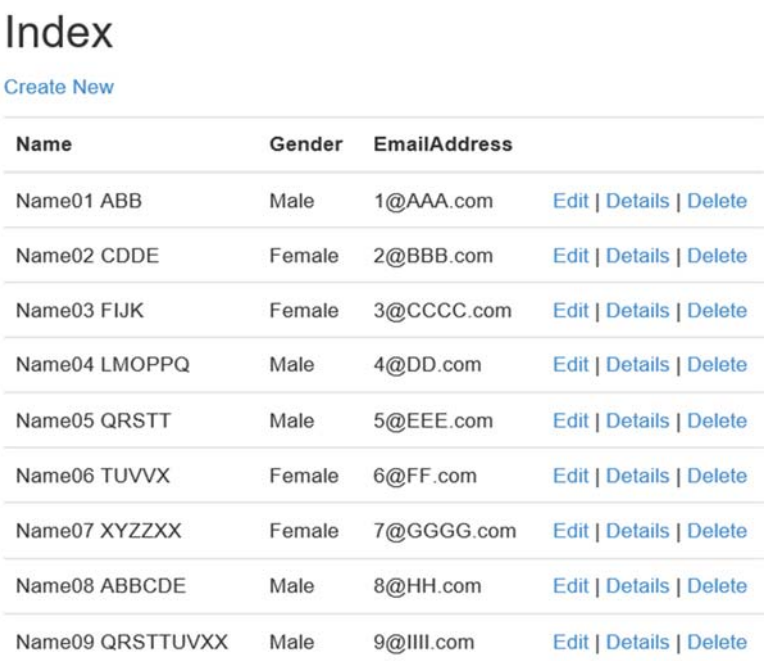




If you see the following error message, then you have to re-build solution before you create the controller.



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



4. OnlineGame.Web - Search Bar

4.1. Views/Gamer/Index.cshtml

```
@model IEnumerable<OnlineGame.Web.Models.Gamer>
@{
    ViewBag.Title = "Gamer Index";
}
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<p>
    @using (Html.BeginForm("Index", "Gamer", FormMethod.Get))
    {
        <b>Search By:</b><br />
        @Html.RadioButton("searchBy", "Name", true) <text>Name</text>
        @Html.RadioButton("searchBy", "Gender") <text>Gender</text><br />
        @Html.TextBox("searchText") <br />
        <input type="submit" value="Go" />
    }
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Gender)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.EmailAddress)
        </th>
```

```
        <th>
            Action
        </th>
    </tr>
    @if (!Model.Any())
    {
        <tr>
            <td colspan="4">
                No matched records.
            </td>
        </tr>
    }
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Gender)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmailAddress)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Details", "Details", new { id = item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Id })
            </td>
        </tr>
    }
</table>
```

4.2. Controllers/GamerController.cs

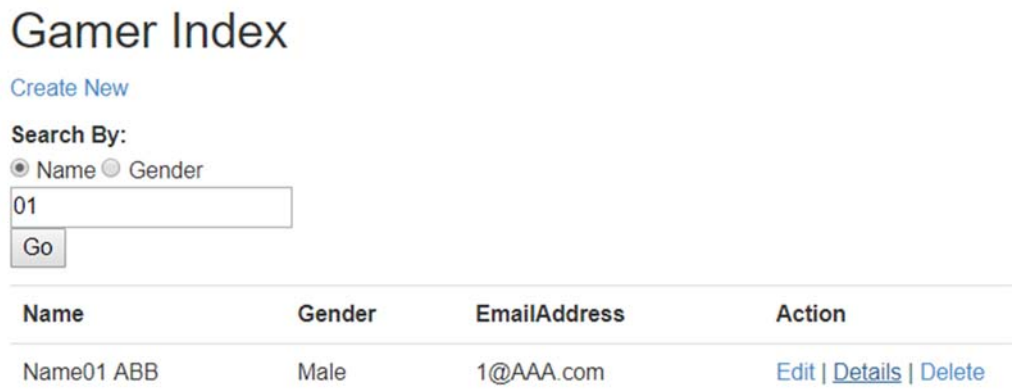
Modify the index action

```
// GET: Gamer
[HttpGet]
public async Task<ActionResult> Index(string searchBy, string searchText)
{
    List<Gamer> gamers = await db.Gamers.ToListAsync();
    if (searchBy == "Gender")
    {
        gamers = await db.Gamers
            .Where(x => x.Gender == searchText || searchText == null)
            .ToListAsync();
    }
    if (searchBy == "Name")
    {
        gamers = await db.Gamers
            .Where(x => x.Name.Contains(searchText) || searchText == null)
```

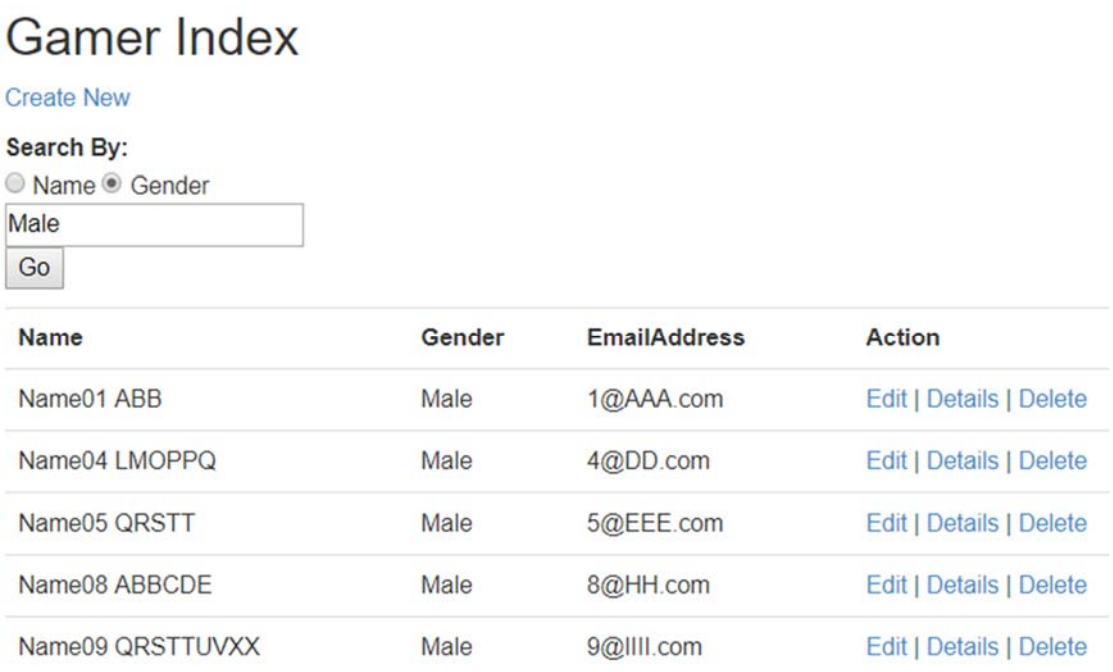
```
                .ToListAsync();
        }
        return View(gamers);
    }
```

4.3. Run Search Bar

http://localhost:52319/?searchBy=Name&searchText=01



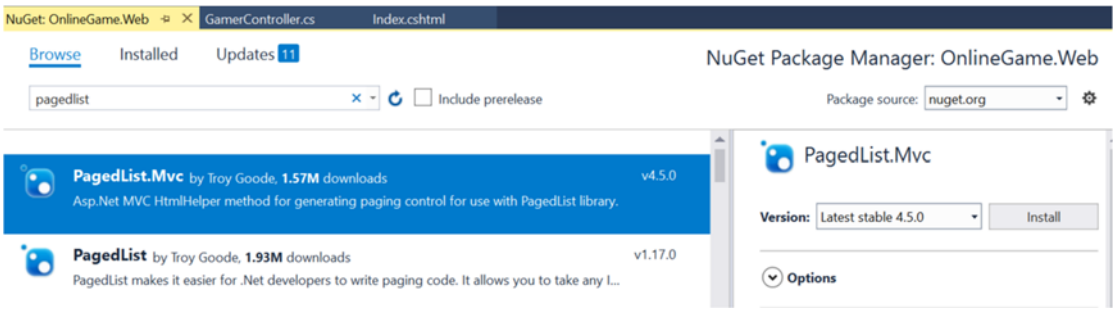
http://localhost:52319/?searchBy=Gender&searchText=Male



5. OnlineGame.Web - PagedList, PagedList.Mvc

5.1. Install NuGet Package

When we install "**PagedList.Mvc**", it will automatically install "**PagedList**".



5.2. Controllers/GamerController.cs

```
// GET: Gamer
[HttpGet]
public async Task<ActionResult> Index(string searchBy, string searchText, int? pageNumber)
{
    List<Gamer> gamers = await db.Gamers.ToListAsync();
    if (searchBy == "Gender")
    {
        gamers = await db.Gamers
            .Where(x => x.Gender == searchText || searchText == null)
            .ToListAsync();
    }
    if (searchBy == "Name")
    {
        gamers = await db.Gamers
            .Where(x => x.Name.Contains(searchText) || searchText == null)
            .ToListAsync();
    }
    //return View(gamers);
    //1.
    //The first parameter is pagenumber
    //pageNumber ?? 1 means if the pageNumber==null, then pageNumber==1
    //2.
    //The 2nd parameter is page size.
    //We set page size is 5.
    IPagedList<Gamer> gamerPagedList = gamers.ToPagedList(pageNumber ?? 1, 5);
    return View(gamerPagedList);
}
```

5.3. Views/Gamer/Index.cshtml

```
@using OnlineGame.Web.Models
```

```
@using PagedList
@using PagedList.Mvc
@*@model IEnumerable<Gamer>*@
@model IPagedList<Gamer>
@{
    ViewBag.Title = "Gamer Index";
}
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<p>
    @using (Html.BeginForm("Index", "Gamer", FormMethod.Get))
    {
        <b>Search By:</b><br />
        @Html.RadioButton("searchBy", "Name", true) <text>Name</text>
        @Html.RadioButton("searchBy", "Gender") <text>Gender</text><br />
        @Html.TextBox("searchText") <br />
        <input type="submit" value="Go" />
    }
</p>
<table class="table">
    <tr>
        <th>
            @*@Html.DisplayNameFor(model => model.Name)*@
            @Html.DisplayNameFor(model => model.First().Name)
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.Gender)*@
            @Html.DisplayNameFor(model => model.First().Gender)
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.EmailAddress)*@
            @Html.DisplayNameFor(model => model.First().EmailAddress)
        </th>
        <th>
             Action
        </th>
    </tr>
    @if (!Model.Any())
    {
        <tr>
            <td colspan="4">
                 No matched records.
            </td>
        </tr>
    }
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Gender)
            </td>
```

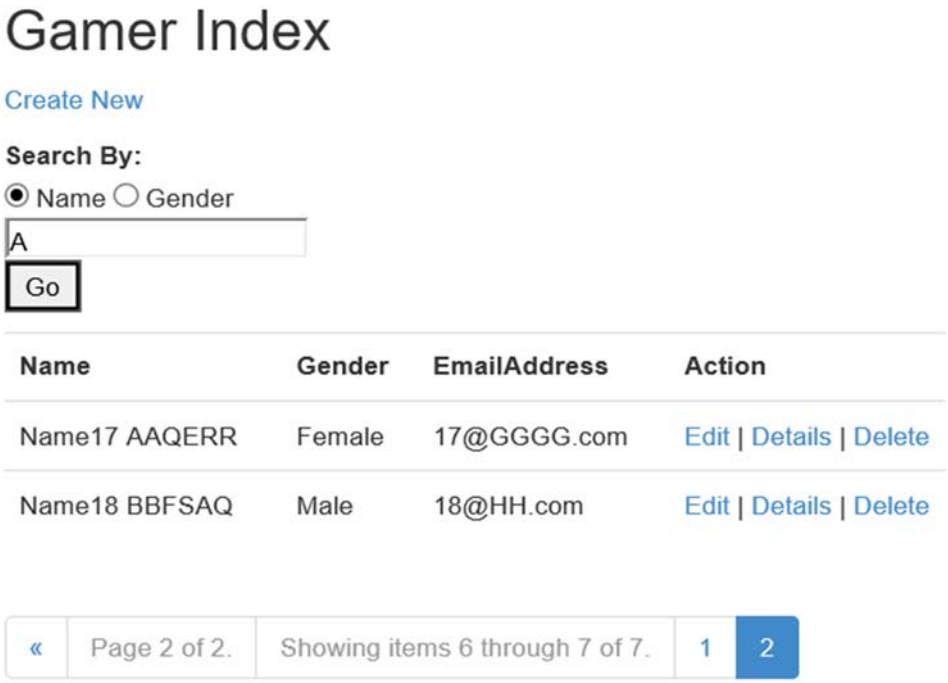
```
            <td>
                @Html.DisplayFor(modelItem => item.EmailAddress)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Details", "Details", new { id = item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Id })
            </td>
        </tr>
    }
</table>
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }))*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true
    })*@
@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
//pageNumber = pageNumber,
pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true,
        DisplayItemSliceAndTotal = true
    })
@*
1.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"] }))
1.1.
```

```
The PagedListPager() 1st parameter is IPagedList
which is the collection of gamers of current page.
1.2.
The PagedListPager() 2nd parameter is Func<int,string> generatePageURL
which we use Url.Action() to generate the Func<int,string>.
In this case, Func<int,string> means that
the input parameter is int, and the output is a string.
1.2.1.
The input parameter is the pageNumber which comes from the query string.
1.2.2.
The output is a string which is the generatePageURL.
In this case, it is Url.Action().
1.2.2.1.
The Url.Action() 1st parameter is action name which is "Index" action.
1.2.2.2.
The Url.Action() 2nd parameter is the route value.
1.2.2.2.1.
searchBy parameter should come from the query string, Request.QueryString["searchBy"].
1.2.2.2.2.
searchText parameter should come from the query string, Request.QueryString["searchText"].
1.3.
It will display the page number even there is only one page.
---------------------------------------
2.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })
2.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
---------------------------------------
3.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true
//    })
3.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
3.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
---------------------------------------
4.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
```

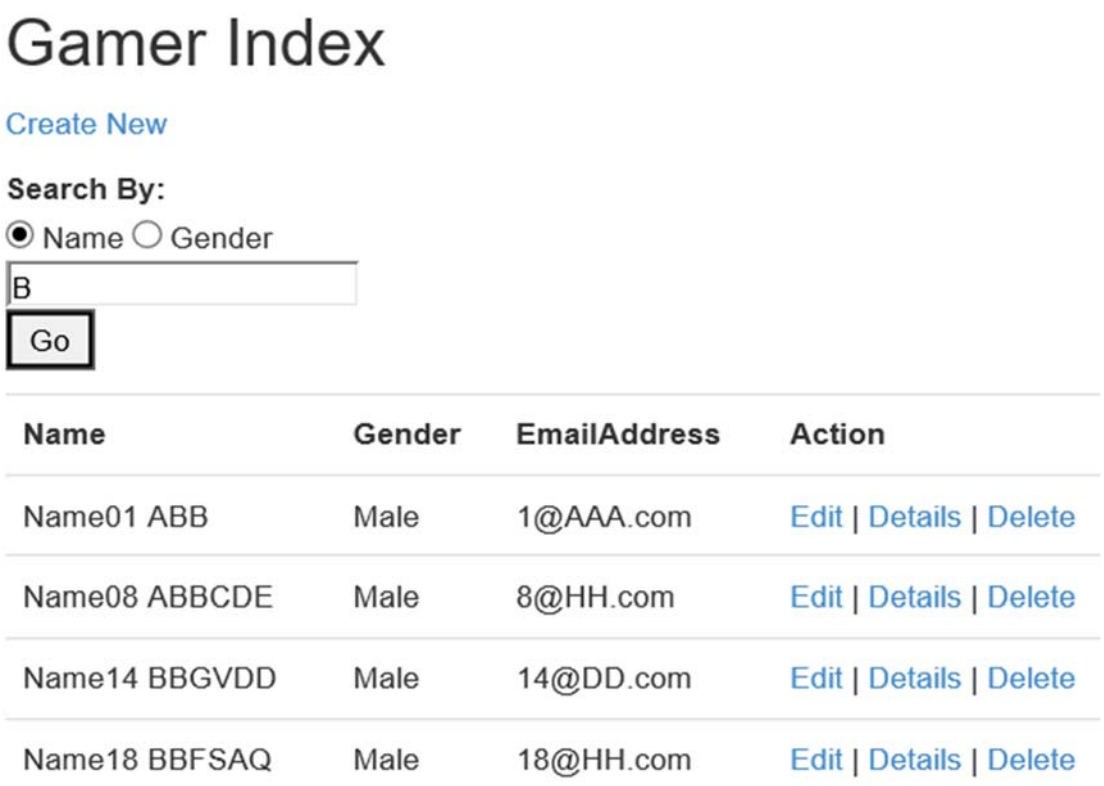
```
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true,
//        DisplayItemSliceAndTotal = true
//    })
4.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
4.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
4.3.
//DisplayItemSliceAndTotal = true
It will display "Showing items 6 through 7 of 7"
*@
```

5.4. Run Search Bar with paging

http://localhost:52319/?pageNumber=2&searchBy=Name&searchText=A



http://localhost:52319/?searchBy=Name&searchText=B



6. OnlineGame.Web - Sorting

6.1. Controllers/GamerController.cs

```
// GET: Gamer
[HttpGet]
public async Task<ActionResult> Index(string searchBy, string searchText, int? pageNumber, string sortBy)
{
    ViewBag.NameSort = String.IsNullOrEmpty(sortBy) ? "Name desc" : "";
    ViewBag.GenderSort = sortBy == "Gender" ? "Gender desc" : "Gender";

    List<Gamer> gamers = await db.Gamers.ToListAsync();
    if (searchBy == "Gender")
    {
        gamers = await db.Gamers
            .Where(x => x.Gender == searchText || searchText == null)
            .ToListAsync();
    }
    if (searchBy == "Name")
    {
        gamers = await db.Gamers
            .Where(x => x.Name.Contains(searchText) || searchText == null)
            .ToListAsync();
    }
    IOrderedEnumerable<Gamer> gamersOrderedEnumerable;
    switch (sortBy)
    {
        case "Name desc":
```

```
            gamersOrderedEnumerable = gamers.OrderByDescending(x => x.Name);
            break;
        case "Gender desc":
            gamersOrderedEnumerable = gamers.OrderByDescending(x => x.Gender);
            break;
        case "Gender":
            gamersOrderedEnumerable = gamers.OrderBy(x => x.Gender);
            break;
        default:
            gamersOrderedEnumerable = gamers.OrderBy(x => x.Name);
            break;
    }
    //1.
    //The first parameter is pagenumber
    //pageNumber ?? 1 means if the pageNumber==null, then pageNumber==1
    //2.
    //The 2nd parameter is page size.
    //We set page size is 5.
    //IPagedList<Gamer> gamerPagedList = gamers.ToPagedList(pageNumber ?? 1, 5);
    IPagedList<Gamer> gamerPagedList = gamersOrderedEnumerable.ToPagedList(pageNumber ?? 1, 5);
    return View(gamerPagedList);
}
```

6.2. Views/Gamer/Index.cshtml

```
@using OnlineGame.Web.Models
@using PagedList
@using PagedList.Mvc
@*@model IEnumerable<Gamer>*@
@model IPagedList<Gamer>
@{
    ViewBag.Title = "Gamer Index";
}
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<p>
    @using (Html.BeginForm("Index", "Gamer", FormMethod.Get))
    {
        <b>Search By:</b><br />
        @Html.RadioButton("searchBy", "Name", true) <text>Name</text>
        @Html.RadioButton("searchBy", "Gender") <text>Gender</text><br />
        @Html.TextBox("searchText") <br />
        <input type="submit" value="Go" />
    }
</p>
<table class="table">
    <tr>
        <th>
            @*@Html.DisplayNameFor(model => model.Name)*@
            @*@Html.DisplayNameFor(model => model.First().Name)*@
            @Html.ActionLink("Name", "Index", new
            {
```

```
                sortBy = ViewBag.NameSort,
                searchBy = Request.QueryString["searchBy"],
                searchText = Request.QueryString["searchText"],
            })
            @*<a href="/?sortBy=Name%20desc">Name</a>*@
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.Gender)*@
            @*@Html.DisplayNameFor(model => model.First().Gender)*@
            @Html.ActionLink("Gender", "Index", new
            {
                sortBy = ViewBag.GenderSort,
                searchBy = Request.QueryString["searchBy"],
                searchText = Request.QueryString["searchText"],
            })
            @*<a href="/?sortBy=Gender">Gender</a>*@
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.EmailAddress)*@
            @Html.DisplayNameFor(model => model.First().EmailAddress)
        </th>
        <th>
            Action
        </th>
    </tr>
    @if (!Model.Any())
    {
        <tr>
            <td colspan="4">
                No matched records.
            </td>
        </tr>
    }
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Gender)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmailAddress)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Details", "Details", new { id = item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Id })
            </td>
        </tr>
    }
</table>
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
```

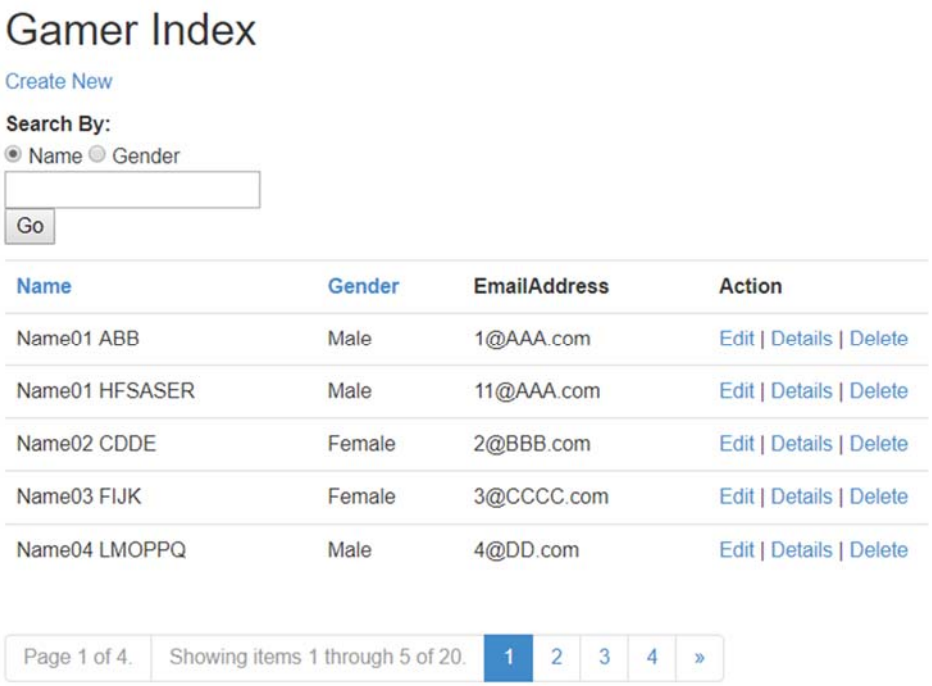
```
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }))*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true
    })*@
@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"],
        sortBy = Request["sortBy"]  //Request from ViewBag
    }),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true,
        DisplayItemSliceAndTotal = true
    })
@*
1.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"] }))
1.1.
The PagedListPager() 1st parameter is IPagedList
which is the collection of gamers of current page.
1.2.
The PagedListPager() 2nd parameter is Func<int,string> generatePageURL
which we use Url.Action() to generate the Func<int,string>.
In this case, Func<int,string> means that
the input parameter is int, and the output is a string.
1.2.1.
The input parameter is the pageNumber which comes from the query string.
1.2.2.
The output is a string which is the generatePageURL.
In this case, it is Url.Action().
1.2.2.1.
The Url.Action() 1st parameter is action name which is "Index" action.
1.2.2.2.
The Url.Action() 2nd parameter is the route value.
```

```
1.2.2.2.1.
searchBy parameter should come from the query string, Request.QueryString["searchBy"].
1.2.2.2.2.
searchText parameter should come from the query string, Request.QueryString["searchText"].
1.3.
It will display the page number even there is only one page.
---------------------------------------
2.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })
2.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
---------------------------------------
3.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true
//    })
3.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
3.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
---------------------------------------
4.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true,
//        DisplayItemSliceAndTotal = true
//    })
4.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
4.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
4.3.
```

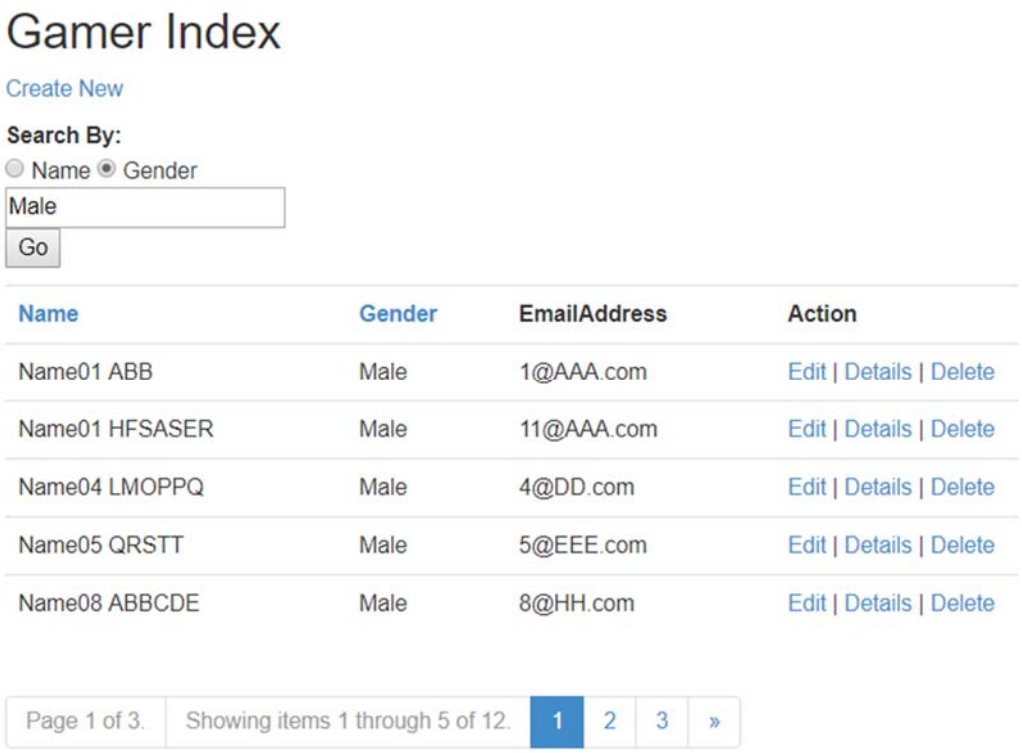
```
//DisplayItemSliceAndTotal = true
It will display "Showing items 6 through 7 of 7"
----------------------------------------
5.
//@Html.ActionLink("Name", "Index", new
//    {
//        sortBy = ViewBag.NameSort,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"],
//    })
...
//@Html.ActionLink("Gender", "Index", new
//    {
//        sortBy = ViewBag.GenderSort,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"],
//    })
5.1.
When http://localhost:52319/Gamer/Index
//<a href="/?sortBy=Name%20desc">Name</a>
//<a href="/?sortBy=Gender">Gender</a>
5.2.
http://localhost:52319/?searchBy=Gender&searchText=Male
//<a href="/?sortBy=Name%20desc">Name</a>
//<a href="/?sortBy=Gender">Gender</a>
5.3.
http://localhost:52319/?sortBy=Name%20desc&searchBy=Gender&searchText=Male
//<a href="/?searchBy=Gender&amp;searchText=Male">Name</a>
//<a href="/?sortBy=Gender&amp;searchBy=Gender&amp;searchText=Male">Gender</a>
*@
```

6.3. Run Search Bar, paging, and Sorting

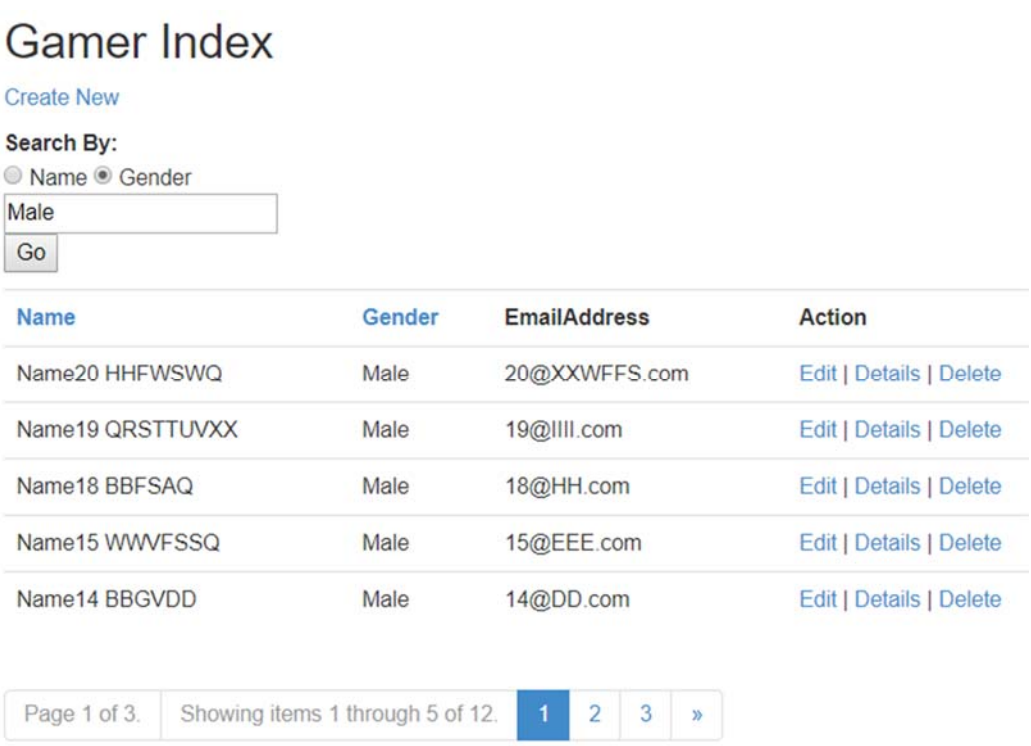
When
http://localhost:52319/Gamer/Index
//Name
//Gender



When
http://localhost:52319/?searchBy=Gender&searchText=Male
//Name
//Gender



When
http://localhost:52319/?sortBy=Name%20desc&searchBy=Gender&searchText=Male
//Name
//Gender



7. OnlineGame.Web - Check box delete All

7.1. Controllers/GamerController.cs

```
[HttpPost]
public async Task<ActionResult> DeleteMultiple(IEnumerable<int>
GamerIdsToDelete, string searchBy, string searchText, int? pageNumber, string sortBy)
{
    //Delete a list of gamers
     List<Gamer> gamers = await db.Gamers.Where(g=> GamerIdsToDelete.Contains(g.Id)).ToListAsync();
     gamers.ForEach(g => db.Gamers.Remove(g));
    await db.SaveChangesAsync();
    return RedirectToAction("Index", new{ searchBy, searchText, pageNumber, sortBy });
}
```

7.2. Views/Gamer/Index.cshtml

```
@using OnlineGame.Web.Models
@using PagedList
@using PagedList.Mvc
@*@model IEnumerable<Gamer>*@
@model IPagedList<Gamer>
@{
    ViewBag.Title = "Gamer Index";
}
```

```
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<p>
    @using (Html.BeginForm("Index", "Gamer", FormMethod.Get))
    {
        <b>Search By:</b><br />
        @Html.RadioButton("searchBy", "Name", true) <text>Name</text>
        @Html.RadioButton("searchBy", "Gender") <text>Gender</text><br />
        @Html.TextBox("searchText") <br />
        <input type="submit" value="Go" />
    }
</p>
@using (Html.BeginForm("DeleteMultiple", "Gamer",new
{
    searchBy = Request.QueryString["searchBy"],
    searchText = Request.QueryString["searchText"],
    pageNumber = Request.QueryString["pageNumber"],
    sortBy = Request["sortBy"]  //Request from ViewBag
}, FormMethod.Post))
{
<table class="table">
    <tr>
        <th></th>
        <th>
            @*@Html.DisplayNameFor(model => model.Name)*@
            @*@Html.DisplayNameFor(model => model.First().Name)*@
            @Html.ActionLink("Name", "Index", new
            {
                sortBy = ViewBag.NameSort,
                searchBy = Request.QueryString["searchBy"],
                searchText = Request.QueryString["searchText"],
            })
            @*<a href="/?sortBy=Name%20desc">Name</a>*@
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.Gender)*@
            @*@Html.DisplayNameFor(model => model.First().Gender)*@
            @Html.ActionLink("Gender", "Index", new
            {
                sortBy = ViewBag.GenderSort,
                searchBy = Request.QueryString["searchBy"],
                searchText = Request.QueryString["searchText"],
            })
            @*<a href="/?sortBy=Gender">Gender</a>*@
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.EmailAddress)*@
            @Html.DisplayNameFor(model => model.First().EmailAddress)
        </th>
        <th>
            Action
        </th>
    </tr>
    @if (!Model.Any())
    {
        <tr>
```

```
            <td colspan="5">
                No matched records.
            </td>
        </tr>
    }
    @foreach (Gamer item in Model)
    {
        <tr>
            <td>
                <input type="checkbox" name="GamerIdsToDelete" id="GamerIdsToDelete" value="@item.Id" />
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Gender)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmailAddress)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Details", "Details", new { id = item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Id })
            </td>
        </tr>
    }
    <tr>
        <td colspan="5">
            <input type="submit" value="Delete Selected" />
        </td>
    </tr>
</table>
}
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }))*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions
```

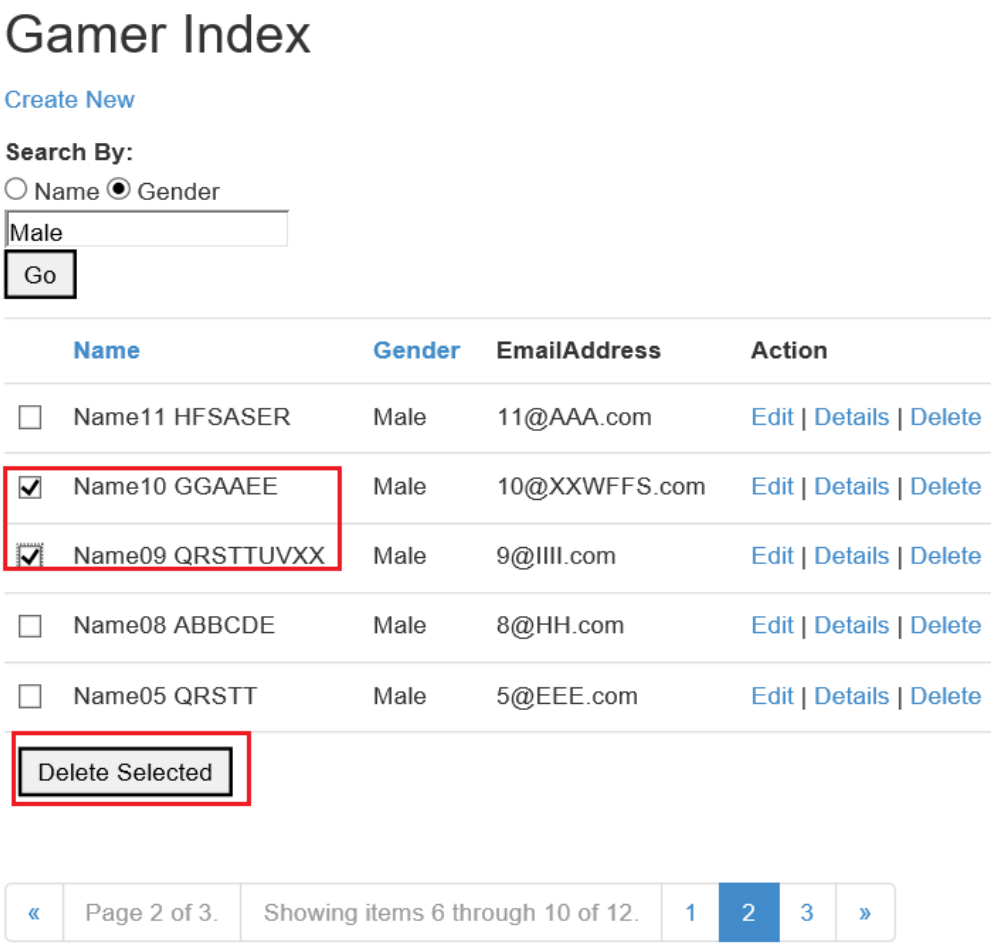
```
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true
    })*@
@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
//pageNumber = pageNumber,
pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"],
        sortBy = Request["sortBy"]  //Request from ViewBag
}),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true,
        DisplayItemSliceAndTotal = true
    })
@*
1.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"] }))
1.1.
The PagedListPager() 1st parameter is IPagedList
which is the collection of gamers of current page.
1.2.
The PagedListPager() 2nd parameter is Func<int,string> generatePageURL
which we use Url.Action() to generate the Func<int,string>.
In this case, Func<int,string> means that
the input parameter is int, and the output is a string.
1.2.1.
The input parameter is the pageNumber which comes from the query string.
1.2.2.
The output is a string which is the generatePageURL.
In this case, it is Url.Action().
1.2.2.1.
The Url.Action() 1st parameter is action name which is "Index" action.
1.2.2.2.
The Url.Action() 2nd parameter is the route value.
1.2.2.2.1.
searchBy parameter should come from the query string, Request.QueryString["searchBy"].
1.2.2.2.2.
searchText parameter should come from the query string, Request.QueryString["searchText"].
1.3.
It will display the page number even there is only one page.
---------------------------------------
2.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })
2.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
---------------------------------------
```

```
3.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true
//    })
3.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
3.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
---------------------------------------
4.
//@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
//    new
//    {
//        //pageNumber = pageNumber,
//        pageNumber,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"]
//    }),
//    new PagedListRenderOptions
//    {
//        Display = PagedListDisplayMode.IfNeeded,
//        DisplayPageCountAndCurrentLocation = true,
//        DisplayItemSliceAndTotal = true
//    })
4.1.
//Display = PagedListDisplayMode.IfNeeded,
It will hide the page number when there is only one page,
because only one page means no need paging.
4.2.
//DisplayPageCountAndCurrentLocation = true
It will display "Page 1 of 3"
4.3.
//DisplayItemSliceAndTotal = true
It will display "Showing items 6 through 7 of 7"
---------------------------------------
5.
//@Html.ActionLink("Name", "Index", new
//    {
//        sortBy = ViewBag.NameSort,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"],
//    })
...
//@Html.ActionLink("Gender", "Index", new
//    {
//        sortBy = ViewBag.GenderSort,
//        searchBy = Request.QueryString["searchBy"],
//        searchText = Request.QueryString["searchText"],
//    })
5.1.
When http://localhost:52319/Gamer/Index
//<a href="/?sortBy=Name%20desc">Name</a>
//<a href="/?sortBy=Gender">Gender</a>
5.2.
```

http://localhost:52319/?searchBy=Gender&searchText=Male
//Name
//Gender
5.3.
http://localhost:52319/?sortBy=Name%20desc&searchBy=Gender&searchText=Male
//Name
//Gender
*@

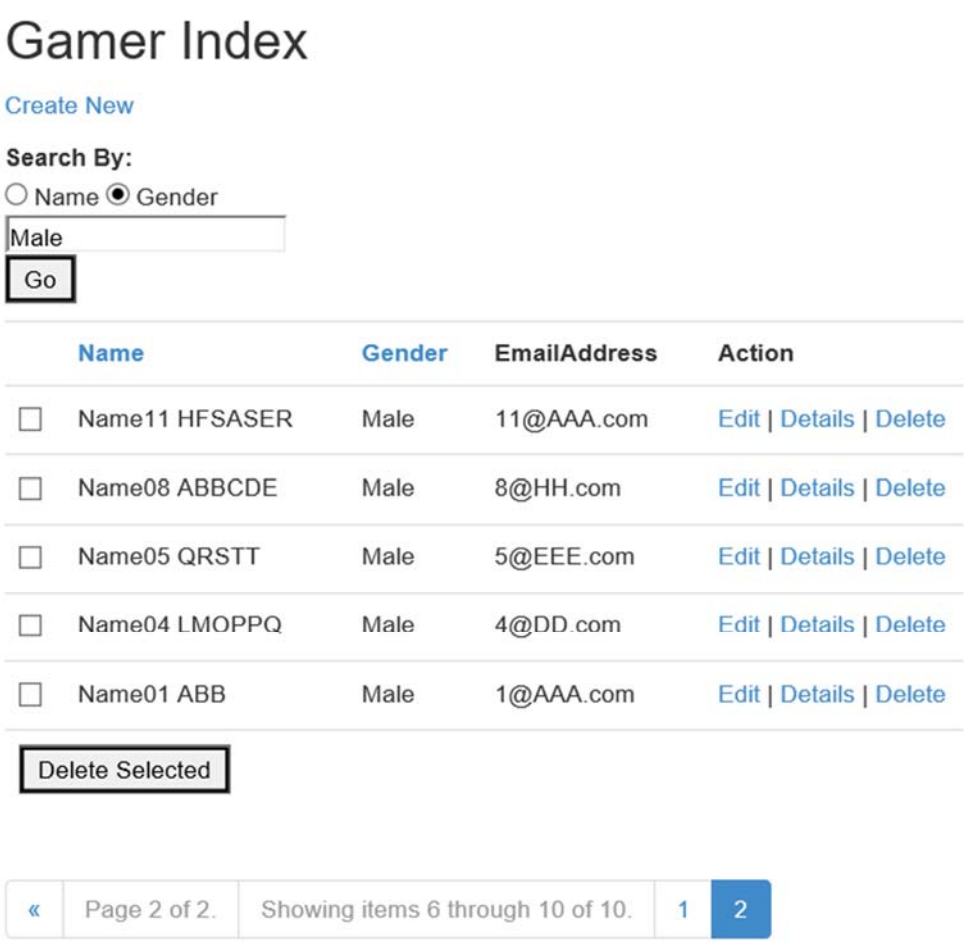
7.3. Run Search Bar, paging, and Sorting

http://localhost:52319/?pageNumber=2&searchBy=Gender&searchText=Male&sortBy=Name%20desc



---

http://localhost:52319/?searchBy=Gender&searchText=Male&pageNumber=2&sortBy=Name%20desc



8. OnlineGame.Web - Check box delete All

8.1. Views/Gamer/Index.cshtml

```
@using OnlineGame.Web.Models
@using PagedList
@using PagedList.Mvc
@*@model IEnumerable<Gamer>*@
@model IPagedList<Gamer>
@{
    ViewBag.Title = "Gamer Index";
}
<script src="~/Scripts/jquery-1.10.2.min.js" type="text/javascript"></script>
<script type="text/javascript" language="javascript">
    $(function () {
        $('#SelectAll').click(function () {
            $("input[name='GamerIdsToDelete']").prop("checked", this.checked);
            //1.
            //this.checked means $("#SelectAll").checked
            //if $("#SelectAll").checked==true, then
            //$("input[name='GamerIdsToDelete']").prop("checked", true);
            //if $("#SelectAll").checked==false, then
            //$("input[name='GamerIdsToDelete']").prop("checked", false);
```

```
            //2.
            //$("input[name='GamerIdsToDelete']") will select all the elements which
name==GamerIdsToDelete.
            //$("input['#GamerIdsToDelete']") will select all only one element which Id==GamerIdsToDelete.
             $("input[name='GamerIdsToDelete']").click(function () {
                 if ($("input[name='GamerIdsToDelete']").length ===
$("input[name='GamerIdsToDelete']:checked").length)
                 {
                     $("#SelectAll").prop("checked", "checked");
                 }
                 else {
                     $("#SelectAll").removeProp("checked");
                 }
             });
            //1.
            //When any of name==GamerIdsToDelete elements has been ckicked,
            //if all input[name='GamerIdsToDelete'] have been checked,
            //then the $("#SelectAll") must be checked.
            //Otherwise, the $("#SelectAll") is un-checked.
        });
        $("#btnDeleteSelected").click(function () {
            var count = $("input[name='GamerIdsToDelete']:checked").length;
            if (count === 0) {
                alert("Please select items to delete.");
                return false;
            }
            else
            {
                return confirm(count + " row(s) will be deleted. Are you sure to continue.");
            }
        });
    });
</script>
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<p>
    @using (Html.BeginForm("Index", "Gamer", FormMethod.Get))
    {
        <b>Search By:</b><br />
        @Html.RadioButton("searchBy", "Name", true) <text>Name</text>
        @Html.RadioButton("searchBy", "Gender") <text>Gender</text><br />
        @Html.TextBox("searchText") <br />
        <input type="submit" value="Go" />
    }
</p>
@using (Html.BeginForm("DeleteMultiple", "Gamer", new
{
    searchBy = Request.QueryString["searchBy"],
    searchText = Request.QueryString["searchText"],
    pageNumber = Request.QueryString["pageNumber"],
    sortBy = Request["sortBy"]  //Request from ViewBag
}, FormMethod.Post))
{
    <table class="table">
        <tr>
            <th>
                <input type="checkbox" name="SelectAll" id="SelectAll" />
```

```
            </th>
            <th>
                @*@Html.DisplayNameFor(model => model.Name)*@
                @*@Html.DisplayNameFor(model => model.First().Name)*@
                @Html.ActionLink("Name", "Index", new
                {
                    sortBy = ViewBag.NameSort,
                    searchBy = Request.QueryString["searchBy"],
                    searchText = Request.QueryString["searchText"],
                })
                @*<a href="/?sortBy=Name%20desc">Name</a>*@
            </th>
            <th>
                @*@Html.DisplayNameFor(model => model.Gender)*@
                @*@Html.DisplayNameFor(model => model.First().Gender)*@
                @Html.ActionLink("Gender", "Index", new
                {
                    sortBy = ViewBag.GenderSort,
                    searchBy = Request.QueryString["searchBy"],
                    searchText = Request.QueryString["searchText"],
                })
                @*<a href="/?sortBy=Gender">Gender</a>*@
            </th>
            <th>
                @*@Html.DisplayNameFor(model => model.EmailAddress)*@
                @Html.DisplayNameFor(model => model.First().EmailAddress)
            </th>
            <th>
                Action
            </th>
        </tr>
        @if (!Model.Any())
        {
            <tr>
                <td colspan="5">
                    No matched records.
                </td>
            </tr>
        }
        @foreach (Gamer item in Model)
        {
            <tr>
                <td>
                    <input type="checkbox" name="GamerIdsToDelete" id="GamerIdsToDelete" value="@item.Id" />
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Name)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Gender)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.EmailAddress)
                </td>
                <td>
                    @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |
```

```
                    @Html.ActionLink("Details", "Details", new { id = item.Id }) |
                    @Html.ActionLink("Delete", "Delete", new { id = item.Id })
                </td>
            </tr>
        }
        <tr>
            <td colspan="5">
                <input type="submit" value="Delete
Selected" id="btnDeleteSelected" name="btnDeleteSelected"/>
            </td>
        </tr>
    </table>
}
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }))*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions{ Display = PagedListDisplayMode.IfNeeded })*@
@*@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
        //pageNumber = pageNumber,
        pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"]
    }),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true
    })*@
@Html.PagedListPager(Model, pageNumber => Url.Action("Index",
    new
    {
//pageNumber = pageNumber,
pageNumber,
        searchBy = Request.QueryString["searchBy"],
        searchText = Request.QueryString["searchText"],
        sortBy = Request["sortBy"]  //Request from ViewBag
}),
    new PagedListRenderOptions
    {
        Display = PagedListDisplayMode.IfNeeded,
        DisplayPageCountAndCurrentLocation = true,
        DisplayItemSliceAndTotal = true
    })
```

Gamer Index

Create New

Search By:

◉ Name ○ Gender

Go

☑	Name	Gender	EmailAddress	Action
☑	Name01 ABB	Male	1@AAA.com	Edit | Details | Delete
☑	Name02 CDDE	Female	2@BBB.com	Edit | Details | Delete
☑	Name03 FIJK	Female	3@CCCC.com	Edit | Details | Delete
☑	Name04 LMOPPQ	Male	4@DD.com	Edit | Details | Delete
☑	Name06 TUVVX	Female	6@FF.com	Edit | Details | Delete

Delete Selected

Page 1 of 4. | Showing items 1 through 5 of 16. | 1 | 2 | 3 | 4 | »

Gamer Index

Create New

Search By:

◉ Name ○ Gender

Go

☐	Name	Gender	EmailAddress	Action
☑	Name01 ABB	Male	1@AAA.com	Edit | Details | Delete
☐	Name02 CDDE	Female	2@BBB.com	Edit | Details | Delete
☑	Name03 FIJK	Female	3@CCCC.com	Edit | Details | Delete
☑	Name04 LMOPPQ	Male	4@DD.com	Edit | Details | Delete
☑	Name06 TUVVX	Female	6@FF.com	Edit | Details | Delete

Delete Selected

Gamer Index

Create New

Search By:

◉ Name ○ Gender

Go

☑	Name	Gender	EmailAddress	Action
☑	Name01 ABB	Male	1@AAA.com	Edit | Details | Delete
☑	Name02 CDDE	Female	2@BBB.com	Edit | Details | Delete
☑	Name03 FIJK	Female	3@CCCC.com	Edit | Details | Delete
☑	Name04 LMOPPQ	Male	4@DD.com	Edit | Details | Delete
☑	Name06 TUVVX	Female	6@FF.com	Edit | Details | Delete

Delete Selected

Page 1 of 4. | Showing items 1 through 5 of 16. | 1 | 2 | 3 | 4 | »

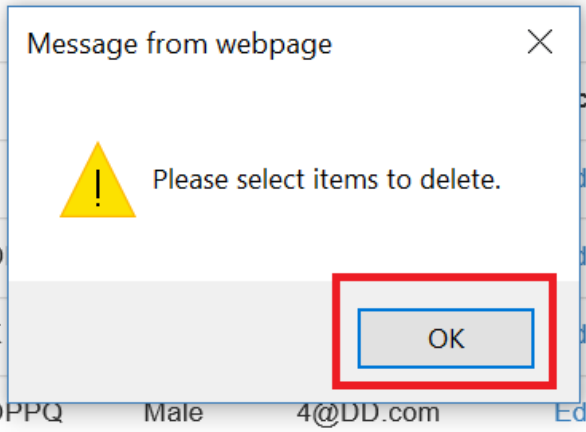
Gamer Index

Create New

Search By:

◉ Name ○ Gender

Go



	Name			ction
☐	Name01 ABB			dit | Details | Delete
☐	Name02 CDD			dit | Details | Delete
☐	Name03 FIJK			dit | Details | Delete
☐	Name04 LMOPPQ	Male	4@DD.com	Edit | Details | Delete
☐	Name06 TUVVX	Female	6@FF.com	Edit | Details | Delete

Delete Selected

Page 1 of 4. | Showing items 1 through 5 of 16. | 1 | 2 | 3 | 4 | »

Gamer Index

Create New

Search By:

◉ Name ○ Gender

Go

	Name			
☐	Name01 A			te
☑	Name02 C			te
☑	Name03 F			te
☐	Name04 L			te
☐	Name06 TUVVX	Female	6@FF.com	Edit | Details | Delete

Delete Selected

Message from webpage

2 row(s) will be deleted. Are you sure to continue.

OK Cancel