(T11)討論 TextArea 的 XSS(CrossSiteScripting)Attack(跨站腳本攻擊)
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924
==========
(T11)討論 TextArea 的 XSS(CrossSiteScripting)Attack(跨站腳本攻擊)
==========

==========

0. Summary

==========
In this tutorial, we will discuss
* AdoDotNetEntityDataModel
* Cross Site Scripting Attack (XSS Attack)
* 2 ways to create TextArea
  * [DataType(DataType.MultilineText)]
  * In the Model, use "**[DataType(DataType.MultilineText)]**"
attribute to decorate the property. It will create TextArea for the property.
  * In the View, use "**@Html.TextAreaFor(model => model.CommentText, new { htmlAttributes = new { @class = "form-control" } })**"
It will create the text area for this property.
* Html Encode

* "**@Html.DisplayFor(modelItem => item.CommentText)**" will return the HTML encoded text.
* "**@Html.Raw(item.CommentText))**" will return mark up that is not HTML encoded.

第 8 章: 駭客任務之使用 XSS 入侵 Web。關於 TextArea 以及 Cross Site Attack。

關於 Cross Site Attack (XSS) 是如何入侵網站的!?以及如何預防大部分的 XSS。

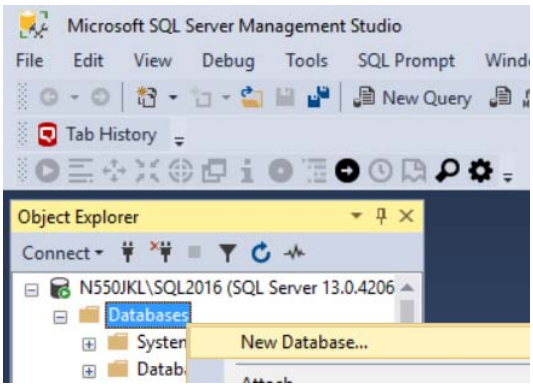
==========

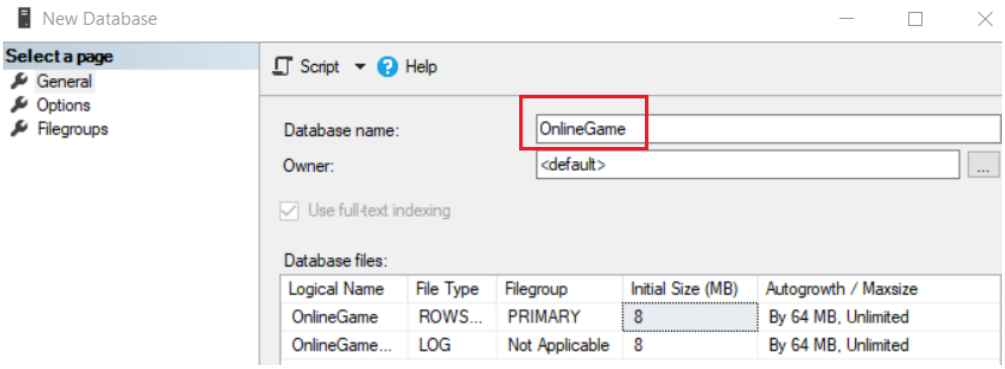
1. OnlineGame DB

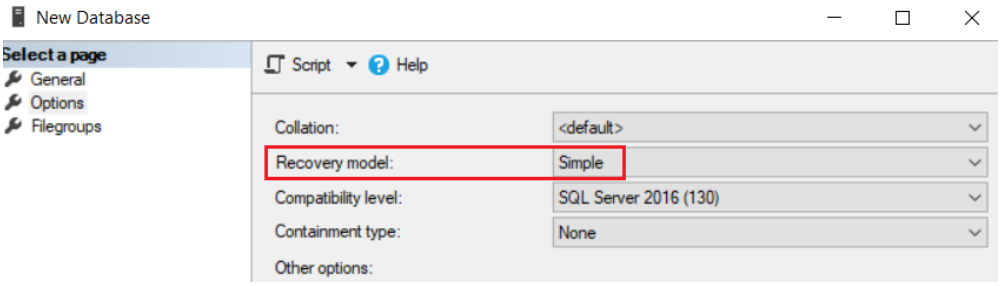
1.1. TSQL

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 = 'ContactComment' ) )
    BEGIN
        TRUNCATE TABLE ContactComment;
        DROP TABLE ContactComment;
    END;
GO -- Run the previous command and begins new batch
--2. Create Table
CREATE TABLE ContactComment
(
   Id INT PRIMARY KEY
              IDENTITY(1, 1)
              NOT NULL ,
   [Name] NVARCHAR(100) NULL ,
   CommentText NVARCHAR(500) NULL
)
--3. Insert Data
INSERT  ContactComment
VALUES  ( N'Name1', N'The comment text from Name1' );
INSERT  ContactComment
VALUES  ( N'Name2', N'The comment text from Name2' );
INSERT  ContactComment
VALUES  ( N'Name3', N'The comment text from Name3' );
--EXEC spGetGamers
--GO -- Run the previous command and begins new batch
```

1.2. Security login

In SQL server

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

-->

General Tab

Login Name :

Tester

Password:

1234

Default Database:

OnlineGame

-->

Server Roles Tab

Select

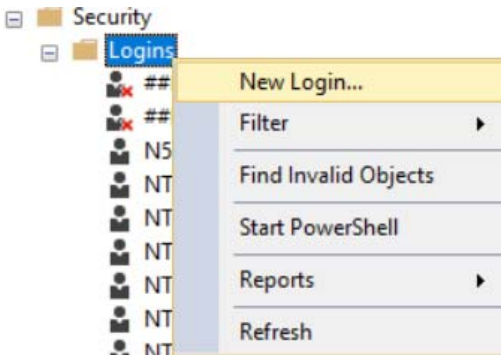
sysadmin

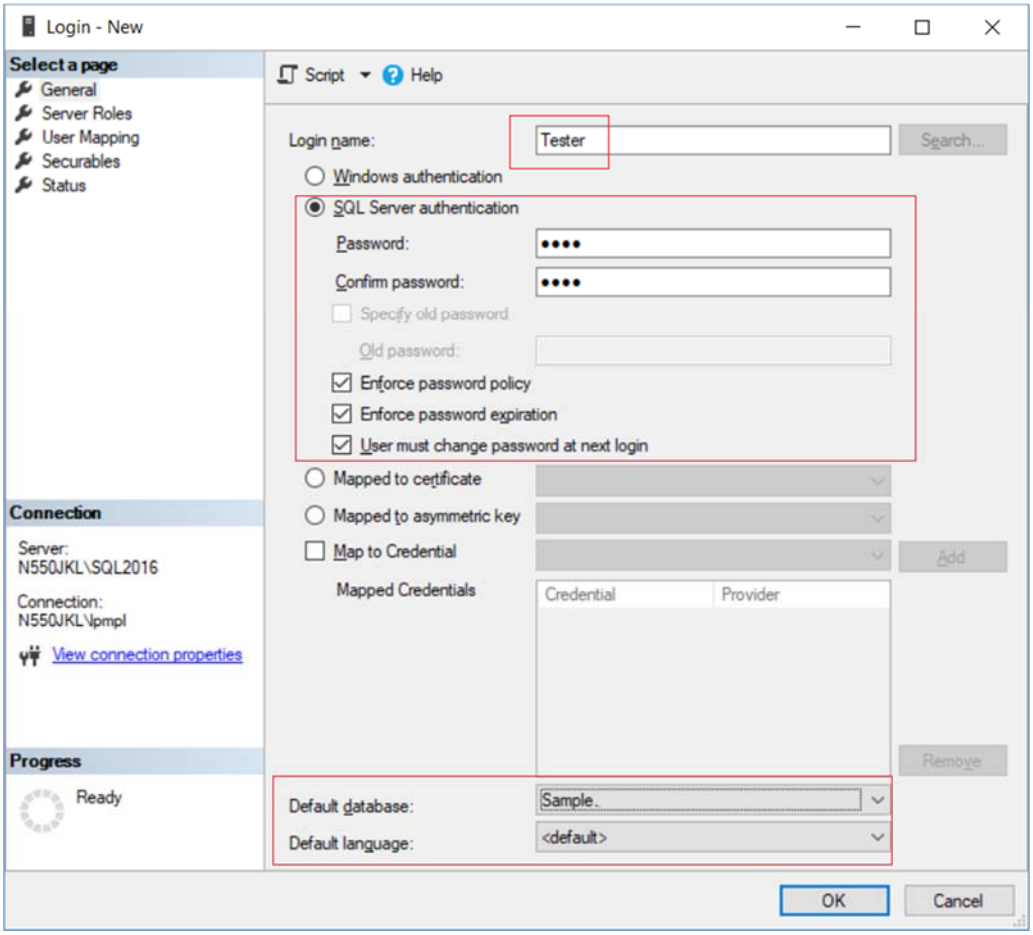
-->

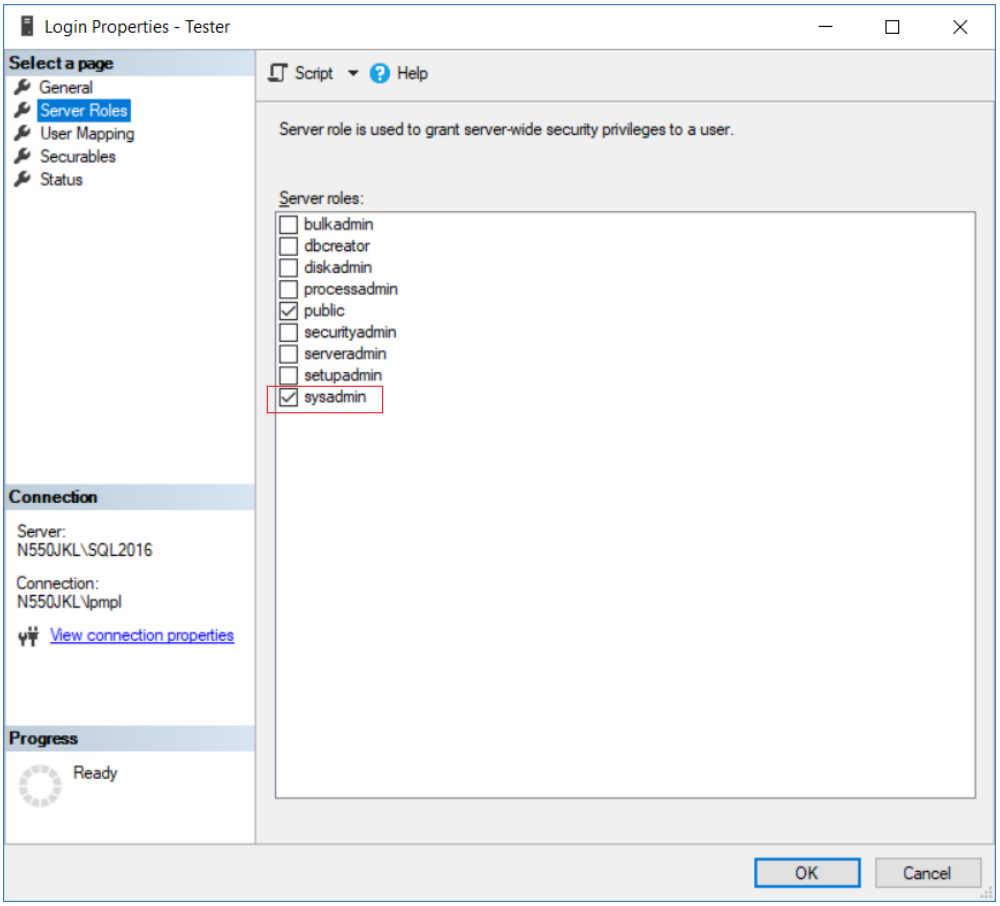
User Mapping Tab

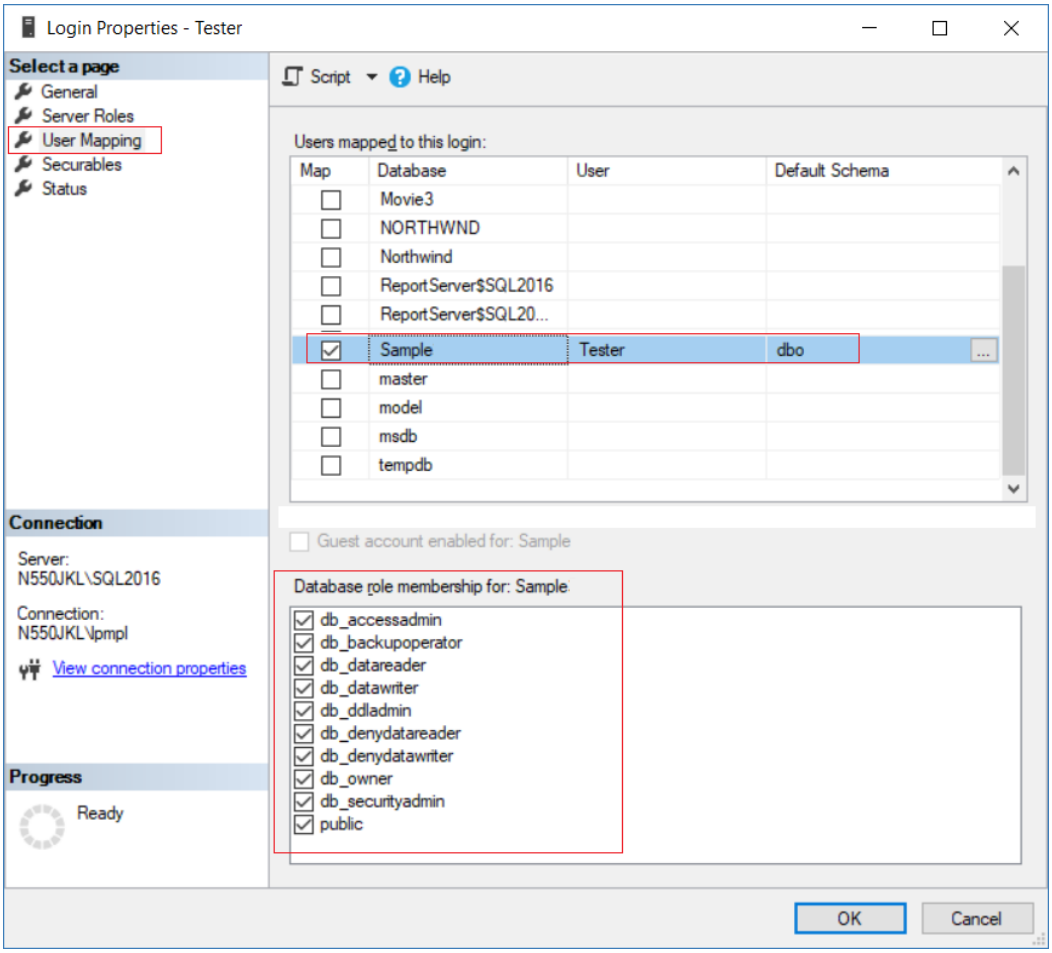
Select **OnlineGame**

Select every single role.


Security
Logins
New Login...
Filter
Find Invalid Objects
Start PowerShell
Reports
Refresh

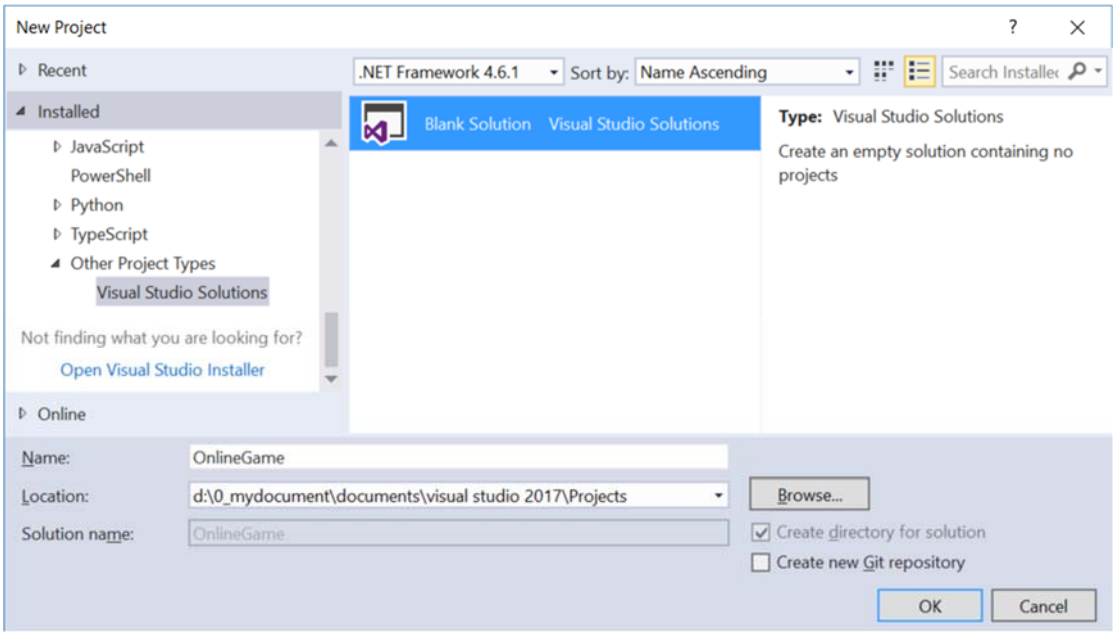

Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Tester
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password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


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

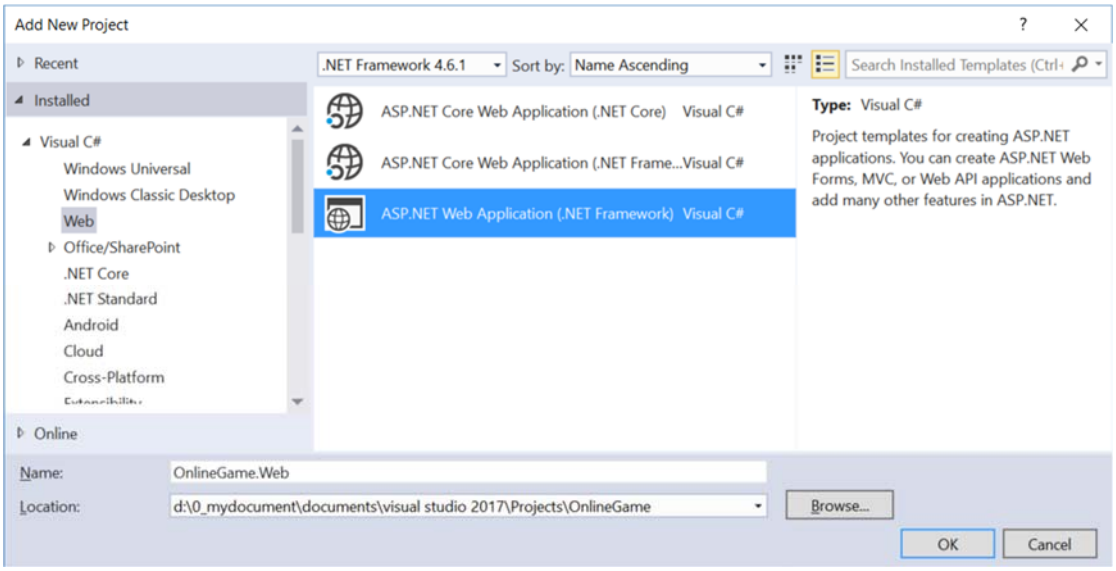
2. New Project - OnlineGame

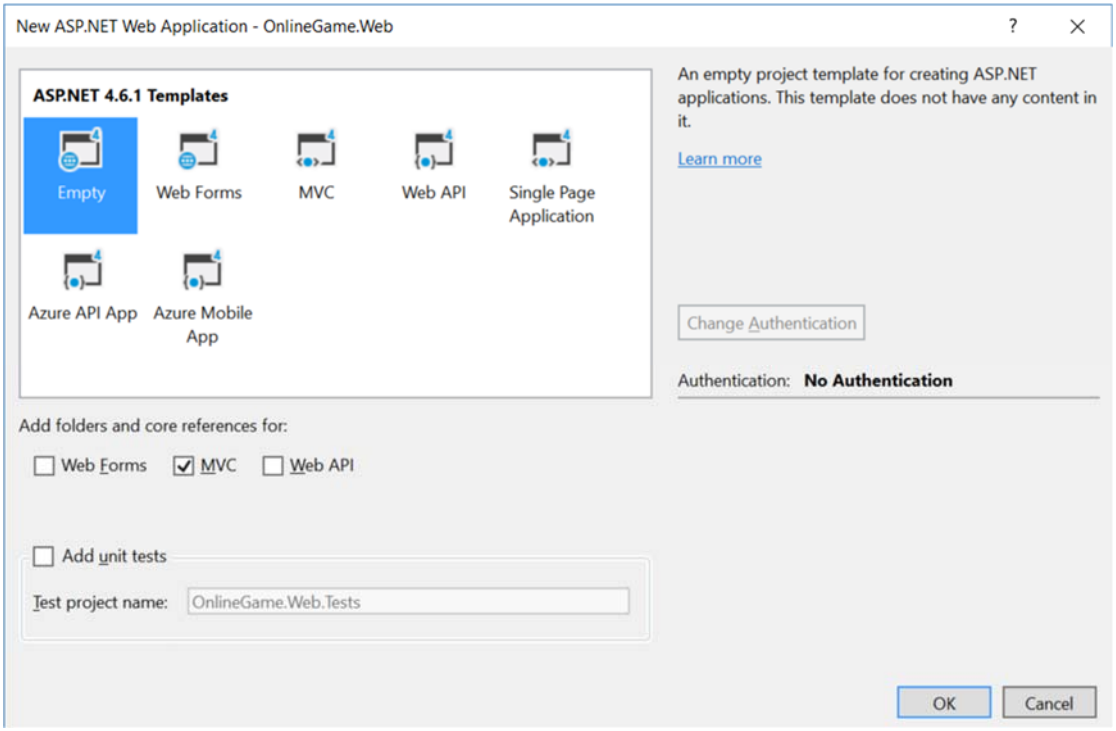
File --> New --> Project... -->
Other Project Types --> Visual Studio Solutions --> Blank Solution
-->
Name: **OnlineGame**



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
```

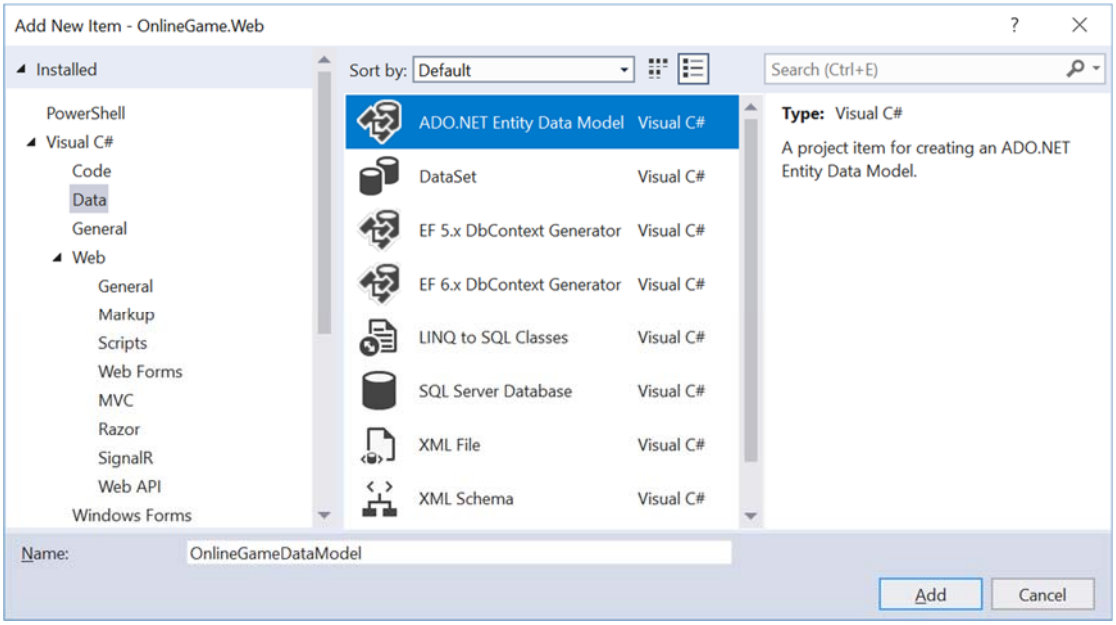
```
Add <trace enabled="true" pageOutput="false"/> under <system.web>
Then run the project, type the following URL
http://localhost/OnlineGame.Web/trace.axd
This will return ASP.NET trace, trace.axd.
If you do not have
// routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
then you can not enable the trace.axd.
*/
```

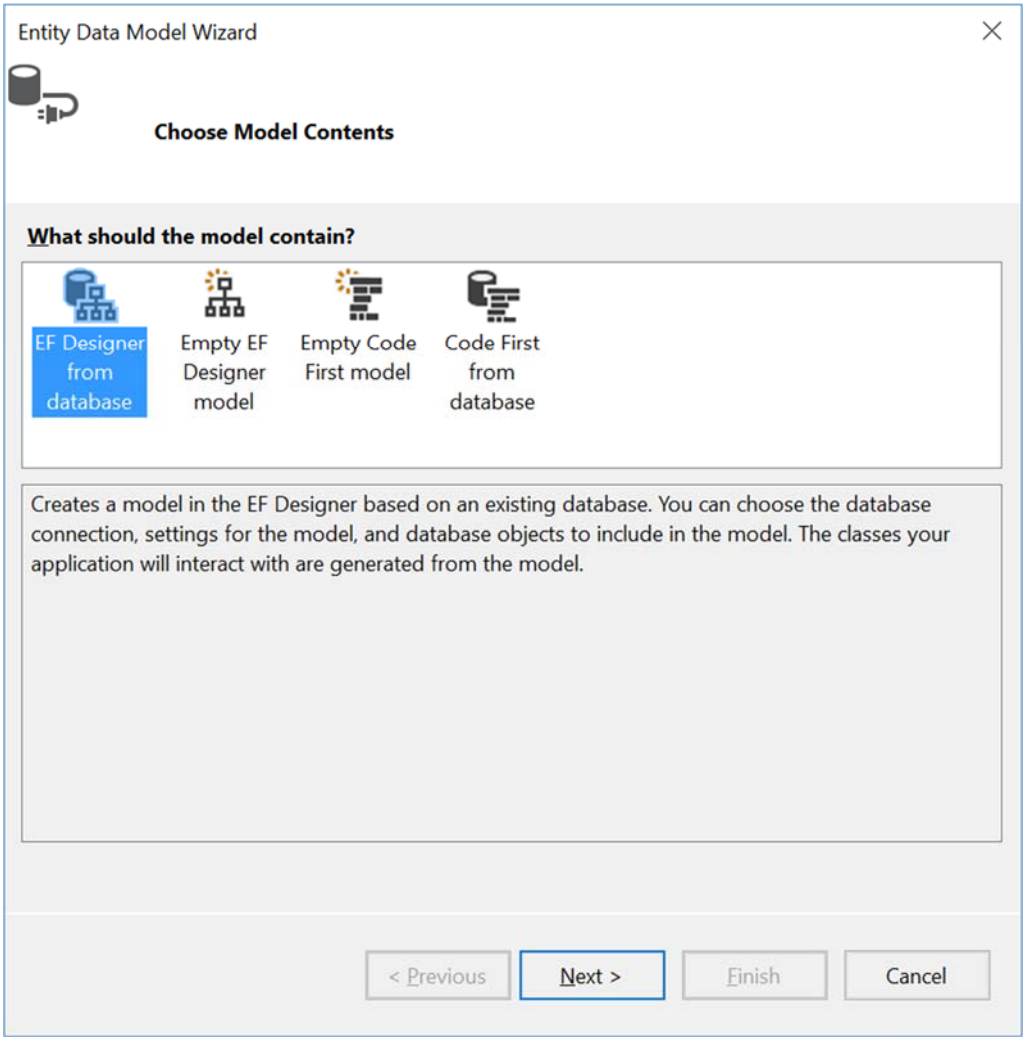
2.2. ADO.Net Entity Data Model - Entity Framework

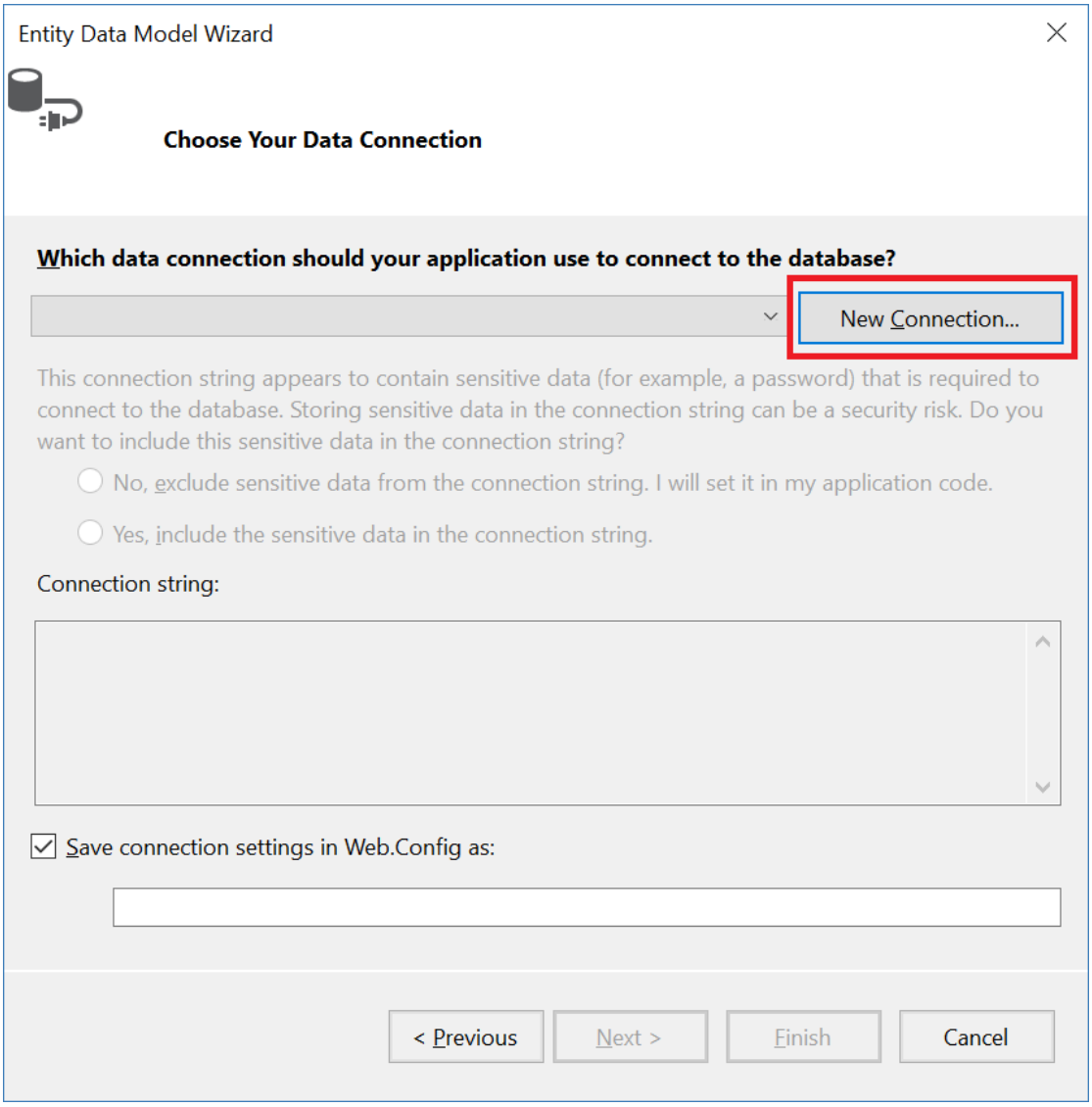
In Visual Studio 2017
Models folder --> Right Click --> Add --> New Item
--> Visual C# --> Data --> ADO.Net Entity Data Model
Name:
OnlineGameDataModel

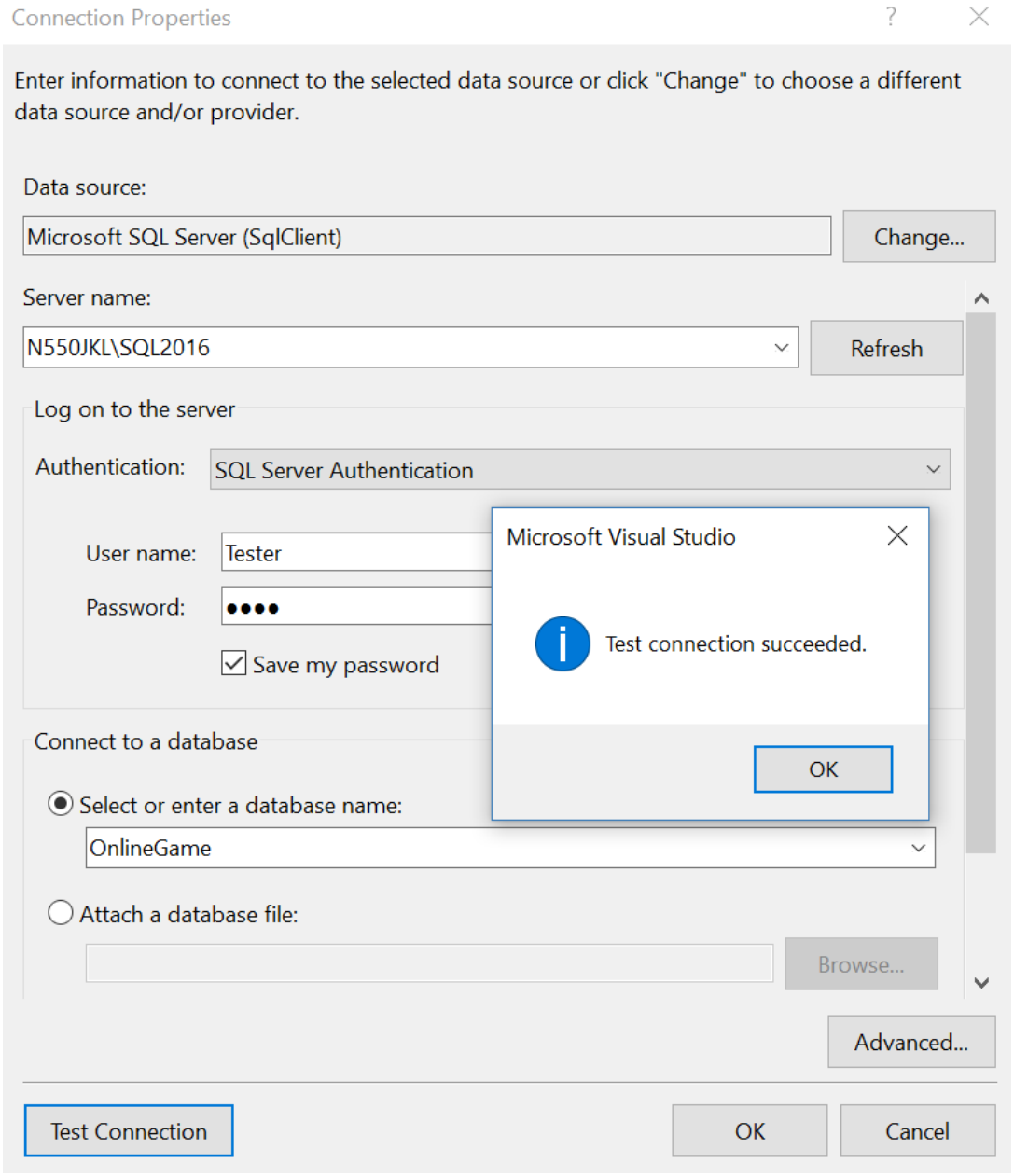
-->
EF Designer from database
....

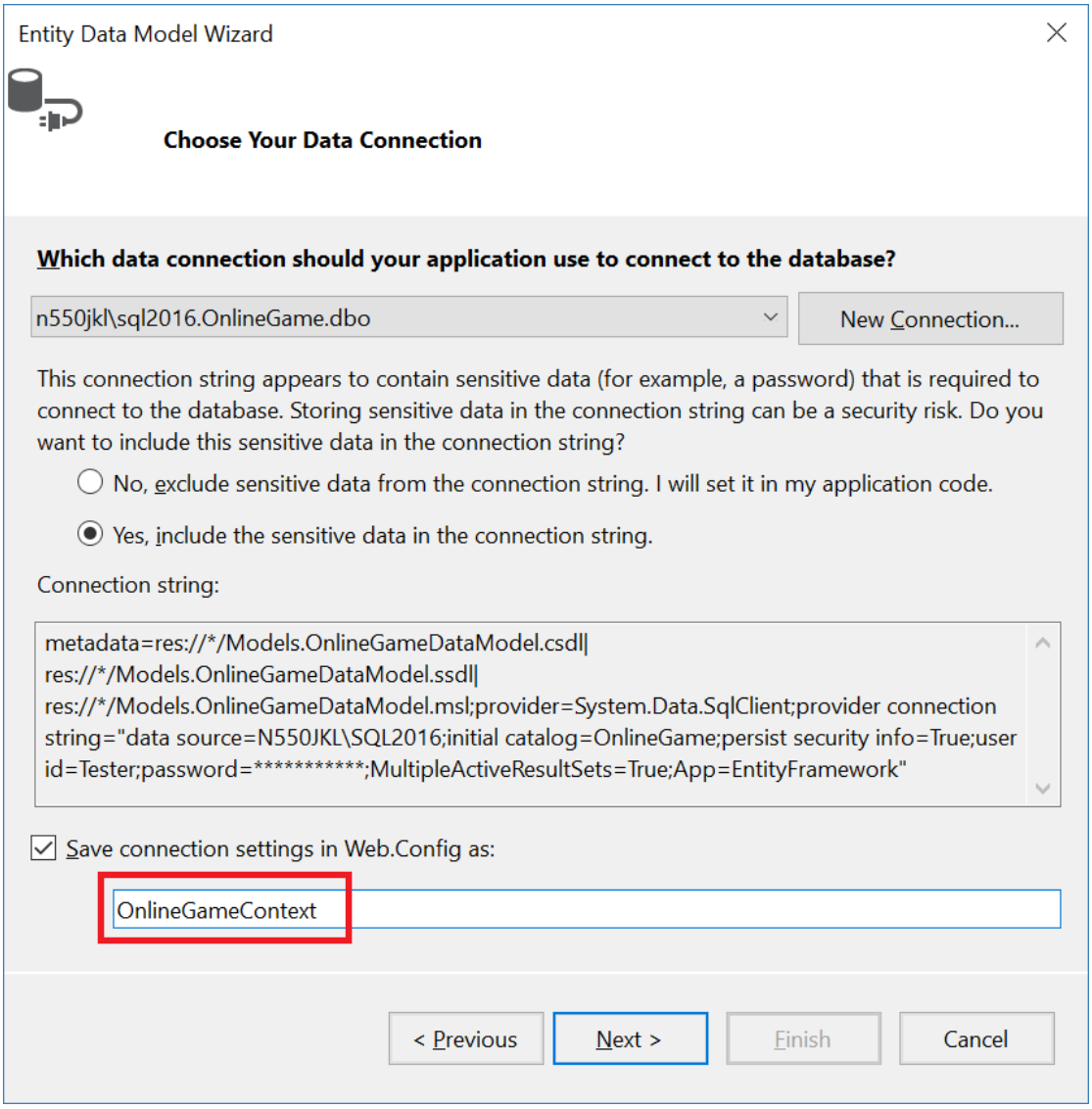
-->
Save Connection settings in Web.Config as:
OnlineGameContext

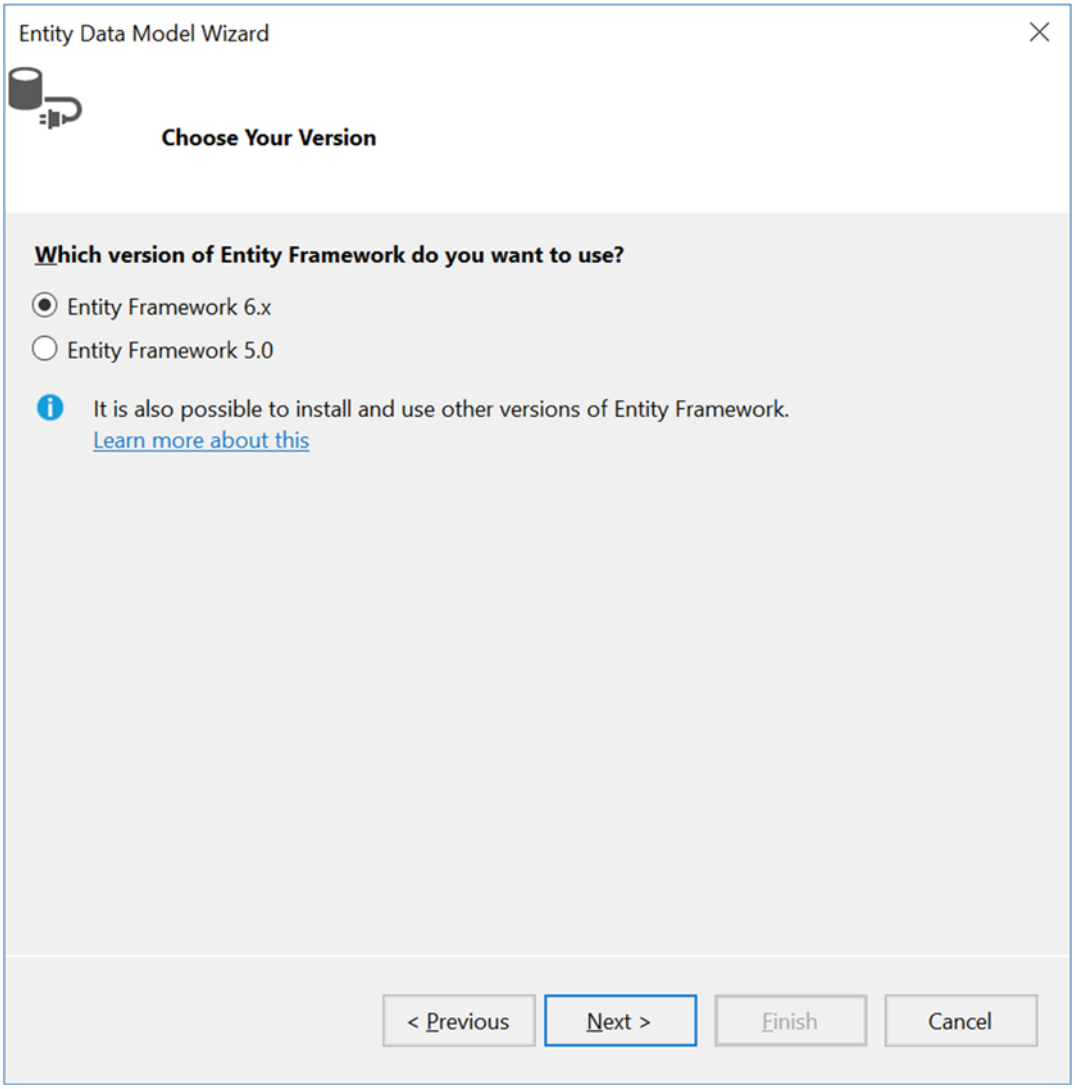


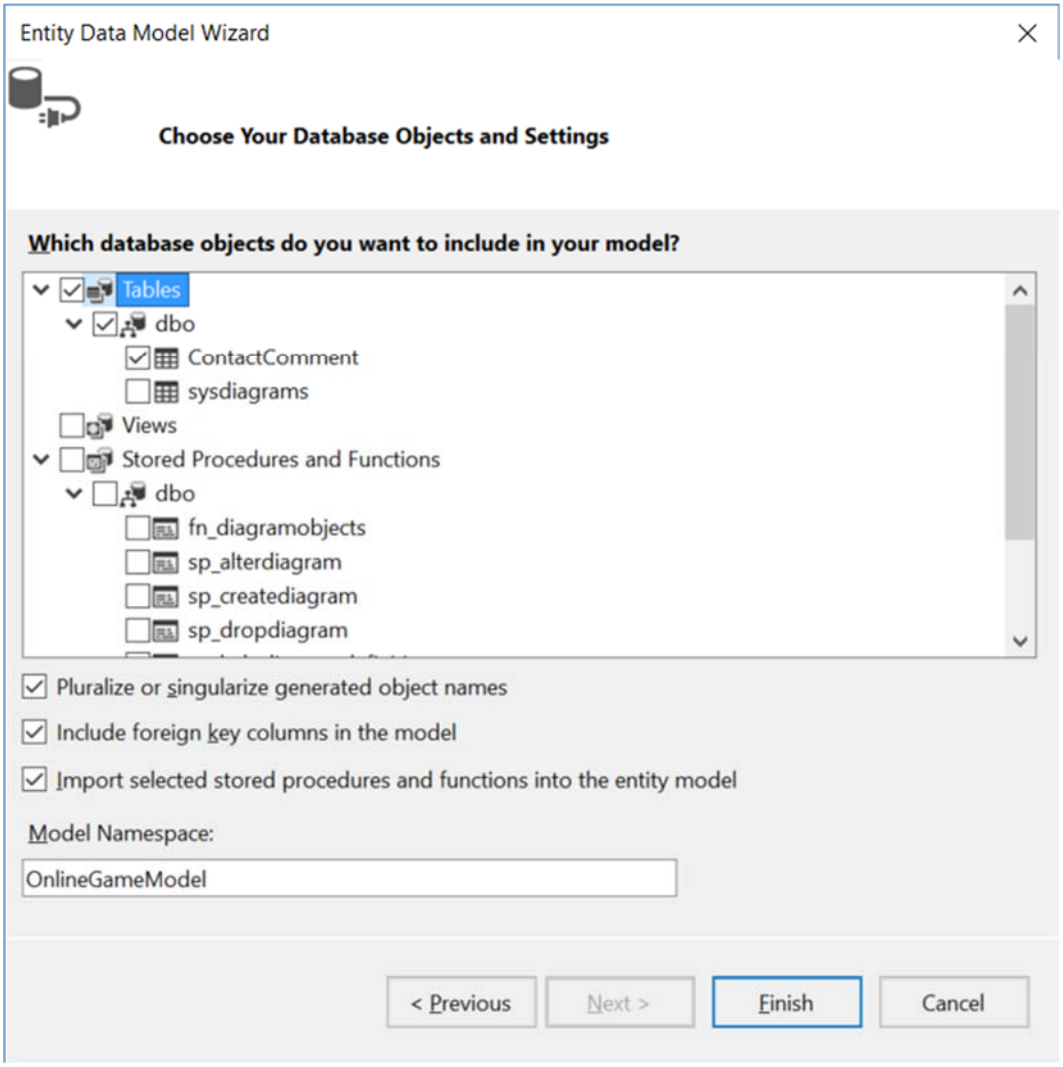

Entity Data Model Wizard
Choose Model Contents
What should the model contain?
EF Designer from database
Empty EF Designer model
Empty Code First model
Code First from database
Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model,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

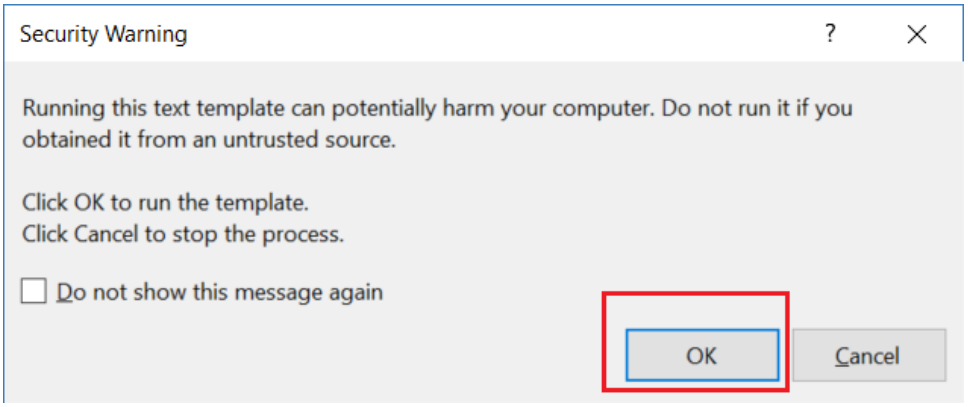

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in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

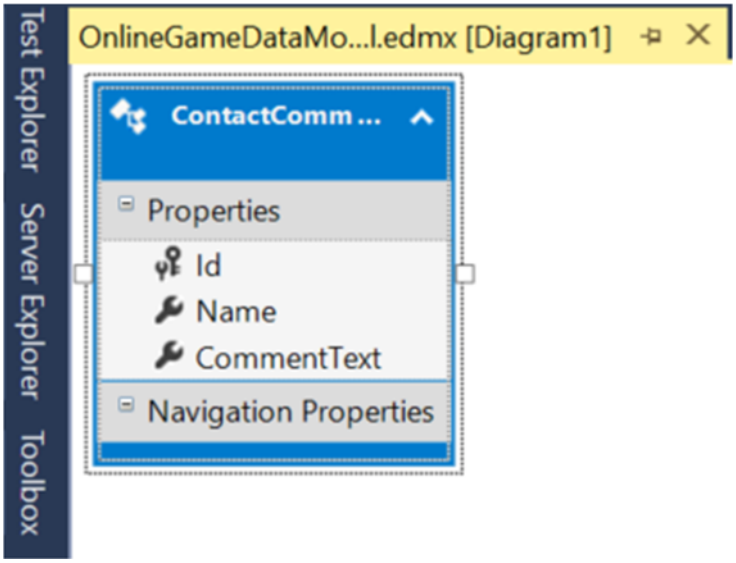

Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
N550JKL\SQL2016
Refresh
Log on to the server
Authentication:
SQL Server Authentication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

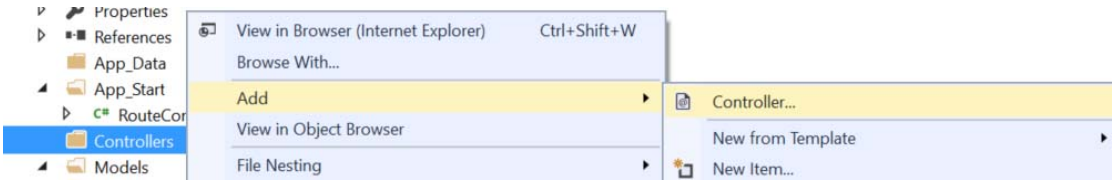

Security Warning
Running this text template can potentially harm your computer. Do not run it if you obtained it from an untrusted source.
Click OK to run the template.
Click Cancel to stop the process.
Do not show this message again
OK
Cancel

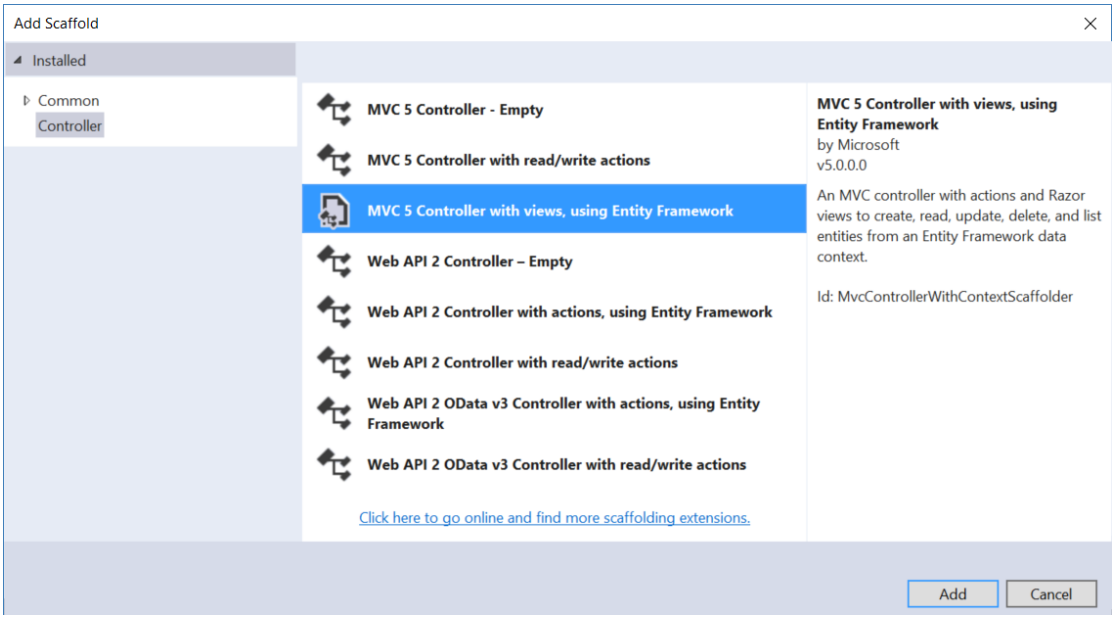


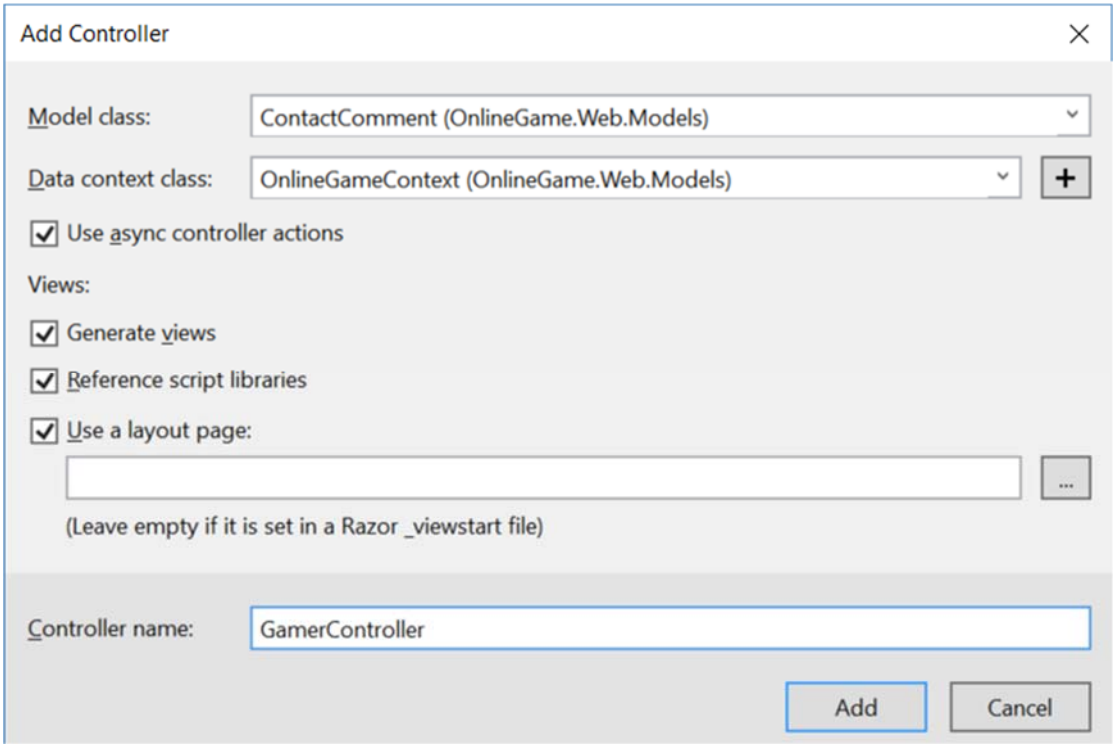
3. OnlineGame.Web

3.1. Controllers/GamersController.cs

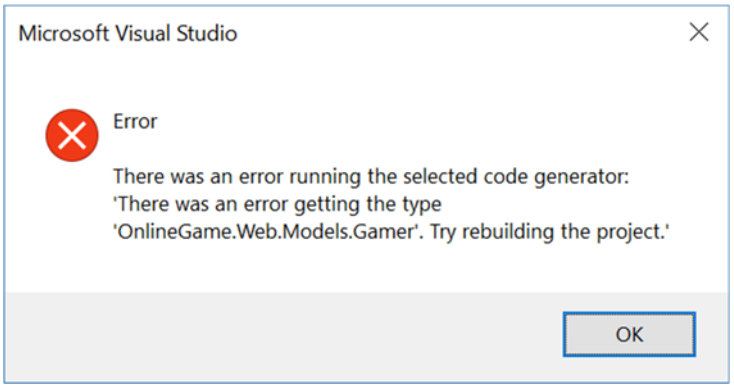
Controllers --> Right click --> Add --> Controller
-->
MVC 5 Controller with views, using Entity Framework



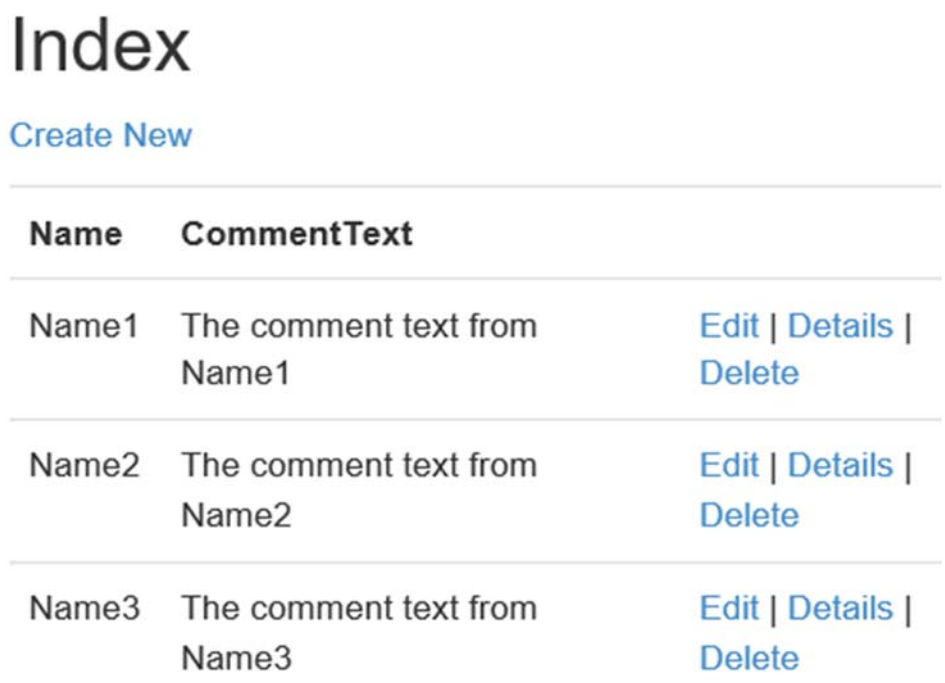




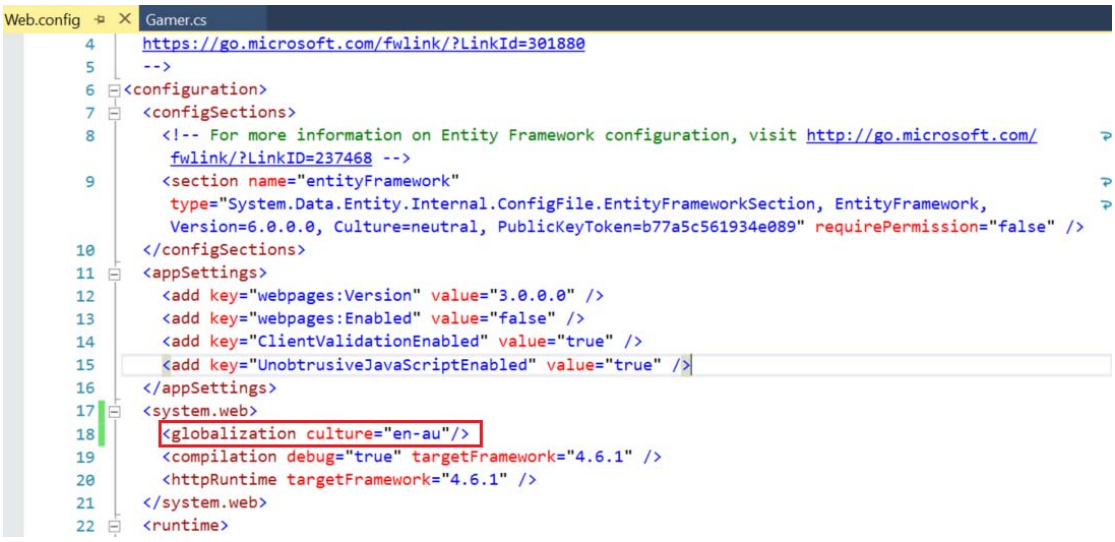
If you see the following error message, then you have to re-build solution before you create the controller.



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



3.2. web.config




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

4. OnlineGame.Web

4.1. Models/Gamer/ContactComment.cs

```
using System.ComponentModel.DataAnnotations;
namespace OnlineGame.Web.Models
{
    [MetadataType(typeof(ContactCommentMetaData))]
    public partial class ContactComment
    {
    }
}
```

4.2. Models/Gamer/ContactCommentMetaData.cs

```
using System.ComponentModel.DataAnnotations;
namespace OnlineGame.Web.Models
{
    public class ContactCommentMetaData
    {
        public int Id { get; set; }
        public string Name { get; set; }
        ////Create TextArea for this property.
        //[DataType(DataType.MultilineText)]
        public string CommentText { get; set; }
    }
}
```

4.3. Views/Gamer/Create.cshtml

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

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

```
            <div class="col-md-10">
                @Html.TextAreaFor(model => model.CommentText, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.CommentText, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

4.4. Views/Gamer/Edit.cshtml

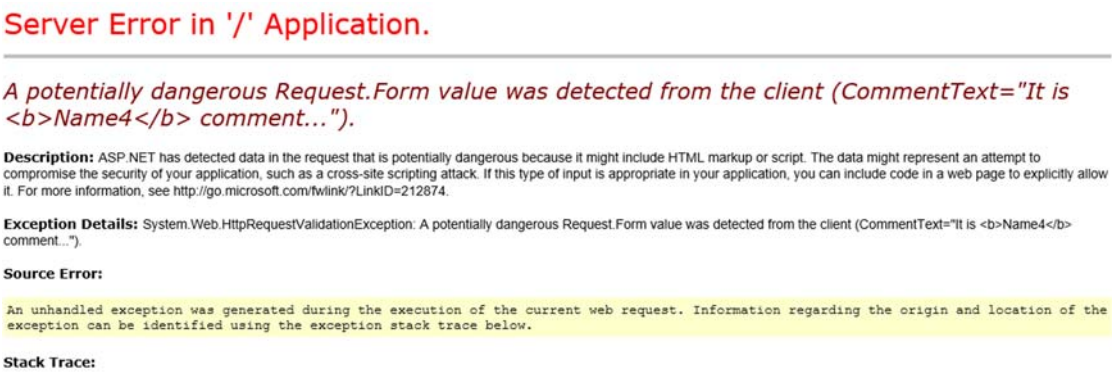
```
@model OnlineGame.Web.Models.ContactComment
@{
    ViewBag.Title = "Edit";
}
<h2>Edit</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

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

```
                @Html.ValidationMessageFor(model => model.CommentText, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

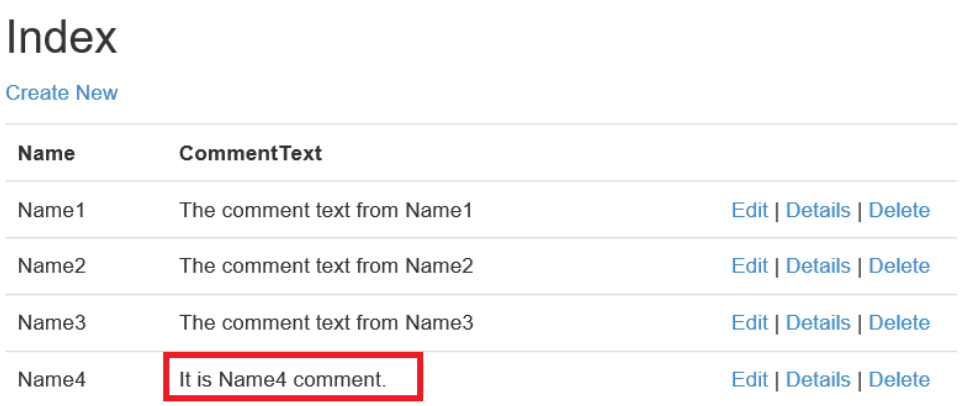
4.5. Create a Data

http://localhost:56064/Gamer/Create
Name4
It is Name4 comment.



Let's do again
http://localhost:56064/Gamer/Create
Name4

It is Name4 comment.



Therefore, we understand by default we can not use any HTML tag in the comment text.
By default, **[ValidateInput(true)]** attribute decorate the Create and Edit action.
If you want to allow HTML tag in the comment text, then you need to use **[ValidateInput(false)]**

4.6. Controllers/GamerController.cs

```
using System.Data.Entity;
using System.Threading.Tasks;
using  System.Net;
using System.Web.Mvc;
using OnlineGame.Web.Models;
namespace OnlineGame.Web.Controllers
{
    public class GamerController : Controller
    {
        private OnlineGameContext db = new OnlineGameContext();
        // GET: Gamer
        public async Task<ActionResult> Index()
        {
            return View(await db.ContactComments.ToListAsync());
        }
        // GET: Gamer/Details/5
        public async Task<ActionResult> Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
            return View(contactComment);
        }
        // GET: Gamer/Create
        public ActionResult Create()
        {
```

```
            return View();
        }
        // POST: Gamer/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        [ValidateInput(false)]
        public async Task<ActionResult> Create([Bind(Include = "Id,Name,CommentText")] ContactComment
contactComment)
        {
            if (ModelState.IsValid)
            {
                db.ContactComments.Add(contactComment);
                await db.SaveChangesAsync();
                return RedirectToAction("Index");
            }
            return View(contactComment);
        }
        // GET: Gamer/Edit/5
        public async Task<ActionResult> Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
            return View(contactComment);
        }
        // POST: Gamer/Edit/5
        // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        [ValidateInput(false)]
        public async Task<ActionResult> Edit([Bind(Include = "Id,Name,CommentText")] ContactComment
contactComment)
        {
            if (ModelState.IsValid)
            {
                db.Entry(contactComment).State = EntityState.Modified;
                await db.SaveChangesAsync();
                return RedirectToAction("Index");
            }
            return View(contactComment);
        }
        // GET: Gamer/Delete/5
        public async Task<ActionResult> Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
```

```
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
            return View(contactComment);
        }
        // POST: Gamer/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> DeleteConfirmed(int id)
        {
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            db.ContactComments.Remove(contactComment);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

4.7. Create a Data

http://localhost:56064/Gamer/Create
Name5
It is Name5 comment.

http://localhost:56064/gamer/index
Go back to Index page
You will see it return the HTML encode text, not mark up.
Thus, we have to use @Html.Raw(...) to return the mark up that is not HTML encoded.

4.8. Use @Html.Raw(...) - Views/Gamer/Index.cshtml

@Html.Raw(...) returns the mark up that is not HTML encoded.

```
@model IEnumerable<OnlineGame.Web.Models.ContactComment>
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.CommentText)
        </th>
        <th></th>
    </tr>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.Raw(item.CommentText))
            @*@Html.DisplayFor(modelItem => item.CommentText)*@
        </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>
```

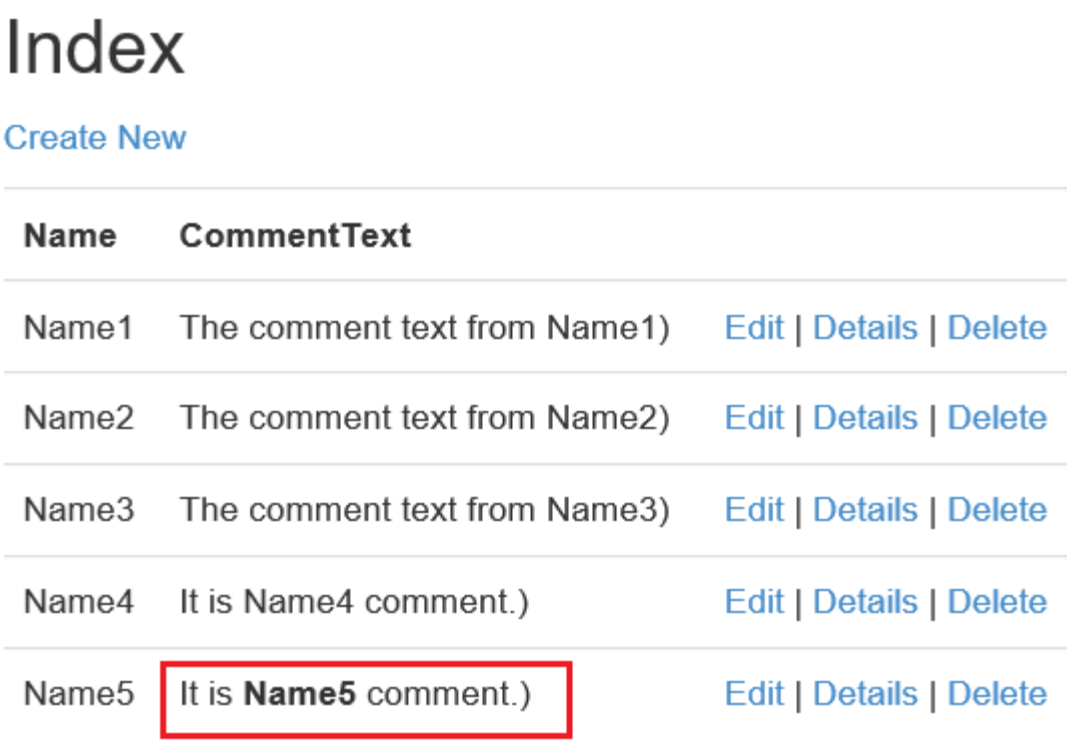
4.9. Use @Html.Raw(...) - Views/Gamer/Details.cshtml

@Html.Raw(...) returns the mark up that is not HTML encoded.

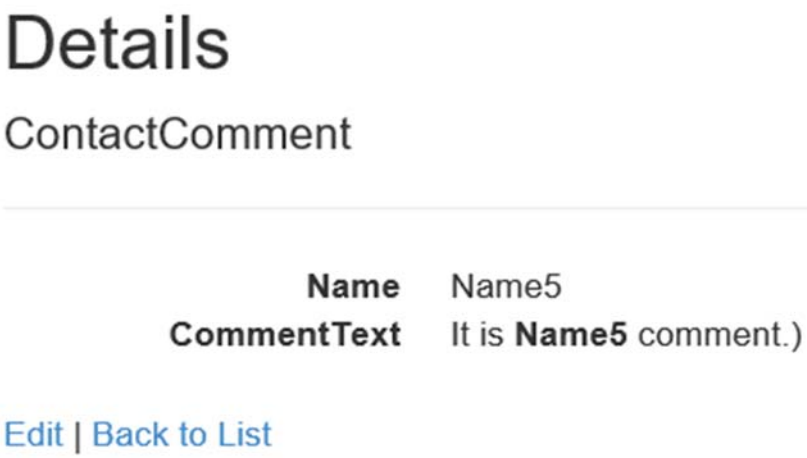
```
@model OnlineGame.Web.Models.ContactComment
@{
    ViewBag.Title = "Details";
}
<h2>Details</h2>
<div>
    <h4>ContactComment</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Name)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.Name)
        </dd>
        <dt>
            @Html.DisplayNameFor(model => model.CommentText)
        </dt>
        <dd>
            @Html.Raw(Model.CommentText))
            @*@Html.DisplayFor(model => model.CommentText)*@
        </dd>
    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

4.10. See Data

http://localhost:56064/Gamer/Index



http://localhost:56064/Gamer/Details/5

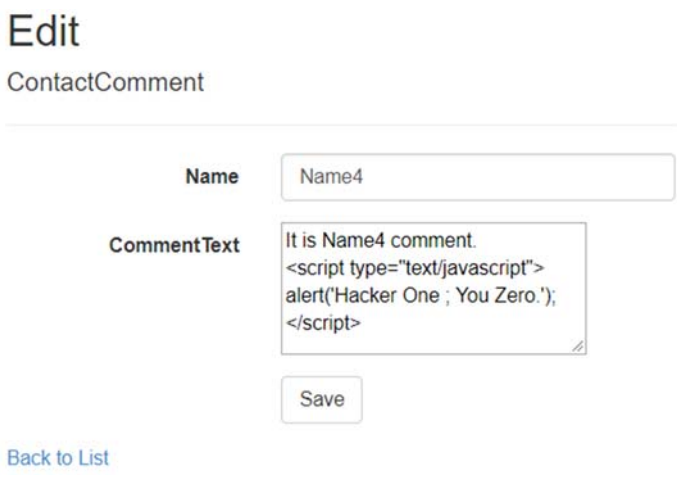


4.11. Edit Data - Cross Site Scripting Attack (XSS Attack)

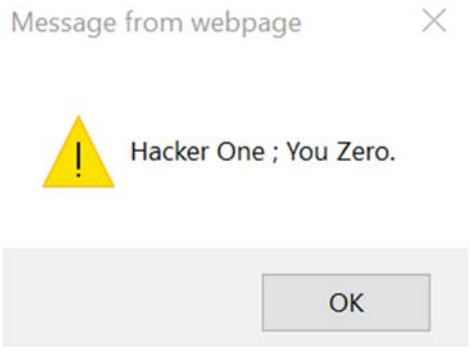
http://localhost:56064/Gamer/Edit/4

Name4

It is Name4 comment.
<script type="text/javascript">
alert('Hacker One ; You Zero.');
</script>



Go back to Index page.
http://localhost:56064/Gamer/Index



Please delete Name4 data

4.12. Controllers/GamerController.cs - Fix Cross Site Scripting Attack (XSS Attack)

```
using System.Data.Entity;
using System.Threading.Tasks;
usingSystem.Net;
using System.Text;
using System.Web;
using System.Web.Mvc;
using OnlineGame.Web.Models;
namespace OnlineGame.Web.Controllers
{
    public class GamerController : Controller
    {
        private OnlineGameContext db = new OnlineGameContext();
        // GET: Gamer
        public async Task<ActionResult> Index()
        {
            return View(await db.ContactComments.ToListAsync());
        }
        // GET: Gamer/Details/5
```

```
        public async Task<ActionResult> Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
            return View(contactComment);
        }
        // GET: Gamer/Create
        public ActionResult Create()
        {
            return View();
        }
        // POST: Gamer/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        [ValidateInput(false)]
        public async Task<ActionResult> Create([Bind(Include = "Id,Name,CommentText")] ContactComment
contactComment)
        {
            if (!ModelState.IsValid)
                return View(contactComment);
            StringBuilder sbCommentText = new StringBuilder();
            // HTML Encode the CommentText
            sbCommentText.Append(HttpUtility.HtmlEncode(contactComment.CommentText));
            // Decode <b> and <u>
            sbCommentText.Replace("&lt;b&gt;", "<b>");
            sbCommentText.Replace("&lt;/b&gt;", "</b>");
            sbCommentText.Replace("&lt;u&gt;", "<u>");
            sbCommentText.Replace("&lt;/u&gt;", "</u>");
contactComment.CommentText = sbCommentText.ToString();
            // HTML Encode the Name
            string strEncodedName = HttpUtility.HtmlEncode(contactComment.Name);
            contactComment.Name = strEncodedName;
            db.ContactComments.Add(contactComment);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
        // GET: Gamer/Edit/5
        public async Task<ActionResult> Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
```

```
            return View(contactComment);
        }
        // POST: Gamer/Edit/5
        // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        [ValidateInput(false)]
        public async Task<ActionResult> Edit([Bind(Include = "Id,Name,CommentText")] ContactComment
contactComment)
        {
            if (!ModelState.IsValid)
                return View(contactComment);
            StringBuilder sbCommentText = new StringBuilder();
            // HTML Encode the CommentText
            sbCommentText.Append(HttpUtility.HtmlEncode(contactComment.CommentText));
            // Decode <b> and <u>
            sbCommentText.Replace("&lt;b&gt;", "<b>");
            sbCommentText.Replace("&lt;/b&gt;", "</b>");
            sbCommentText.Replace("&lt;u&gt;", "<u>");
            sbCommentText.Replace("&lt;/u&gt;", "</u>");
contactComment.CommentText = sbCommentText.ToString();
            // HTML Encode the Name
            string strEncodedName = HttpUtility.HtmlEncode(contactComment.Name);
            contactComment.Name = strEncodedName;
            db.Entry(contactComment).State = EntityState.Modified;
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
        // GET: Gamer/Delete/5
        public async Task<ActionResult> Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            if (contactComment == null)
            {
                return HttpNotFound();
            }
            return View(contactComment);
        }
        // POST: Gamer/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> DeleteConfirmed(int id)
        {
            ContactComment contactComment = await db.ContactComments.FindAsync(id);
            db.ContactComments.Remove(contactComment);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
```

```
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

4.13. Create a data

http://localhost:56064/Gamer/Create

Name6

It is Name6 comment.
<script type="text/javascript">
alert('Hacker One ; You Zero.');
</script>