(T20)討論Validation(驗證)、ResourceFile、WebConfigString、EmailFormatValidation(Email格式驗證)、EmailNotRegisteredValidation(驗證Email還沒被註冊)、JqueryUiDatePicker  
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924  
=======================================================================  
(T20)討論Validation(驗證)、ResourceFile、WebConfigString、EmailFormatValidation(Email格式驗證)、EmailNotRegisteredValidation(驗證Email還沒被註冊)、JqueryUiDatePicker

(T20-1)討論Validation(驗證)、ResourceFile (1. to 2.8.)

(T20-2)討論Validation(驗證) (2.9.)

(T20-3)討論StringLength、RegularExpression、Required、EnableValidation、EnableValidationSummar (2.10. to 2.11.)

(T20-4)討論DisplayFormat、DateTimeformat (2.12.)

(T20-5)討論Range、DateRange、BetweenMinimumDateAndToday、BeforeToday、DisplayFormat、DateTimeformat (3.)

(T20-6)討論RegularExpression、EmailFormatValidation(Email格式驗證) (4.1. to 4.2.)

(T20-7)討論Compare、Remote、RemoteClientServer、EmailNotRegisteredValidation(驗證Email還沒被註冊) (4.3. to 4.5.)  
=======================================================================  
0. Summary

-----------

1. OnlineGame DB

1.1. TSQL

1.2. Security login

-----------

2. OnlineGame Solution

2.1. OnlineGame Solution

2.2. OnlineGame.Web

2.3. Regular expression

2.4. Web.config

2.4.1. Web.config

2.4.2. The way to use key of Web.config in a class

2.4.3. The way to use key of in view

2.5. WebShare/WebShareConst.cs

2.6. Global.asax.cs

2.7. App\_Start/RouteConfig.cs

2.8. Error Handling

2.8.1. Web.config

2.8.2. App\_Start/FilterConfig.cs

2.8.3. Global.asax.cs

2.8.4. Controllers/ErrorController.cs

2.8.5. Error View

2.8.5.1. Views/Shared/Resource/ErrorResource.resx (NameSpace: OnlineGame.Web.Error)

2.8.5.2. Views/Shared/Error.cshtml

2.8.5.3. Views/Shared/UnauthorizedError.cshtml

2.8.5.4. Views/Shared/NotFound.cshtml

2.7.5.5. Views/Shared/InternalServerError.cshtml

2.9. WebShare/Attribute

2.9.1. CustomizeCacheAttribute

2.9.1.1. WebShare/Attribute/CustomizeCacheAttribute.cs

2.9.1.2. The way to use WebShare/Attribute/CustomizeCacheAttribute.cs

2.9.2. DateRangeAttribute, BetweenMinimumDateAndTodayAttribute, BeforeTodayAttribute

2.9.2.1. WebShare/Attribute/DateRangeAttribute.cs

2.9.2.2. WebShare/Attribute/BetweenMinimumDateAndTodayAttribute.cs

2.9.2.3. WebShare/Attribute/BeforeTodayAttribute.cs

2.9.2.4. The way to use DateRangeAttribute, BetweenMinimumDateAndTodayAttribute, BeforeTodayAttribute

2.9.3. RemoteClientServerAttribute.cs

2.9.3.1. WebShare/Attribute/RemoteClientServerAttribute.cs

2.9.3.2. The way to use RemoteClientServerAttribute

2.10. Enable client side validation

2.10.1. Enable Unobtrusive Validation

2.10.1.1. Enable Unobtrusive Validation Method1 - Web.config

2.10.1.2. Enable client side validation Method2 - Global.asax.cs

2.10.1.3. Enable client side validation Method3 - In the View

2.10.2. In the View

2.10.3. How it works

2.10.3.1. How it works - Models/Gamer/Gamer.cs

2.10.3.2. How it works - Models/Gamer/GamerMetaData.cs

2.10.3.3. How it works - Edit.cshtml

2.11. Enable client side validation Summary

2.11.1. \_Layout.cshtml

2.11.2. Content/Site.css

2.11.3. In the View - validation Summary

2.11.4. In the View - Field validation

2.12. Fix the Date format dd/mm/yyy issue - Jquery Validation in Chrome and Safari won't validate with dd/mm/yyyy

2.12.1. Induction

2.12.1.1. Web.config

2.12.1.2. WebShare/WebShareConst.cs

2.12.1.3. Models/Gamer/Gamer.cs

2.12.1.4. Models/Gamer/GamerMetaData.cs

2.12.1.5. Views/Gamer/Edit.cshtml Or Views/Gamer/Create.cshtml

2.12.2. Fix the Date format dd/mm/yyy issue

2.12.2.1. Scripts/operation.js

2.12.2.2. Views/Shared/\_Layout.cshtml

2.12.3. DatePicker

2.12.3.1. Install Jquery UI

2.12.3.2. Views/Shared/\_Layout.cshtml

2.12.3.3. WebShare/WebShareConst.cs

2.12.3.4. Views/Gamer/Edit.cshtml Or Views/Gamer/Create.cshtml

2.12.3.5. Client Side Date Range

-----------

3. OnlineGame.Web

3.1. ADO.Net Entity Data Model - Entity Framework

3.2. Controllers/GamerController.cs

-----------

4. OnlineGame.Web

4.1. Models/Gamer/Gamer.cs

4.2. Models/Gamer/GamerMetaData.cs

4.3. Controllers/GamerController.cs

4.4. Views/Gamer/Edit.cshtml

4.5. Views/Gamer/Create.cshtml  
=======================================================================

0. Summary

=========================================

In this tutorial, we will discuss

\* Resource file

\* Unobtrusive JavaScript is a js file which separated from the main html markup.

\* 2 Ways to store date format string

    \* In the public const variable

        \* private const string DateFormat = "dd/MM/yyyy";

        \* You may apply this const variable in validation attributes in Model class.

    \* In Web.config

        \* private static string DateFormat =  System.Configuration.ConfigurationManager.AppSettings["DateDormat"];

        \* You may NOT apply this web config key variable in validation attributes in Model class.

\* StringLength

    \* [StringLength(20, MinimumLength = 2)]

\* RegularExpression

Reference:

    \* <https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>

    \* <https://regexr.com/>

    \* [RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$")] //First Name Last  Name

    \* [RegularExpression(@"^[\w-\.\_\+%]+@(?:[\w-]+\.)+[\w]{2,6}$", ErrorMessage = "Email is  not valid")]  //EmailAddress Regular Expression

\* Range

    \* [Range(1, 1000000)]

    \* [Range(typeof(DateTime), "1/1/1970", "1/1/2001", ErrorMessage = "Date is out of  Range")]    //Error - Client Side validation will never pass

\* CustomizeValidationAttribute

    \* [DateRange("01/01/1970", "01/01/2001")]

    \* [BetweenMinimumDateAndToday("01/01/1970")]

    \* [BeforeToday]

\* Validate Date

    \* In the Web Config

        \* <globalization culture="en-au" uiCulture="en-AU" />

    \* In the WebShare/WebShareConst.cs

        \* public const string DateFormat = "dd/MM/yyyy";

        \* public const string DateStringFormat = "{0:dd/MM/yyyy}";

    \* In the Model Class

        \* [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

        \* [DisplayFormat(DataFormatString = WebShareConst.DateStringFormat,  ApplyFormatInEditMode = true)]

\* jquery ui datepicker

-------------------------------------------------

手殘驅魔人之Validation驗證。String解放地之ResourceFile和WebConfigString。無招勝有招的RegularExpression，又見Jquery Ui DatePicker

\* Bad Data(不良數據)常常是將資料庫毀壞的兇手。

\* Bad Data(不良數據)通常來自使用者的手殘。

\* 所以我們需要「手殘驅魔人 」Validation驗證，讓使用者可以防呆，避免Bad Data(不良數據)進入資料庫。

\* 詳細解說String該放在ResourceFile還是WebConfigString。

\* 詳細解說「無招勝有招」的RegularExpression。

\* 又見Jquery UI DatePicker。

\* 關於NameProperty，StringLength，RegularExpression，Required，EnableValidation，EnableValidationSummary。

\* 關於DisplayFormat，DateTimeformat。

\* 關於Range，DateRange，BetweenMinimumDateAndToday，BeforeToday，DisplayFormat，DateTimeformat。

\* 驗證Email，關於RegularExpression。

\* 驗證Email，確認Email，關於Compare，Remote，RemoteClientServer。

\* 修改Site.css

-------------------------------------------------

Reference:

\* ASP.NET MVC 3: Integrating with the jQuery UI date picker and adding a jQuery validate date range validator

    \* <https://blogs.msdn.microsoft.com/stuartleeks/2011/01/25/asp-net-mvc-3-integrating-with-the-jquery-ui-date-picker-and-adding-a-jquery-validate-date-range-validator/>

\* One place to save date format dd/mm/yyyy

    \* <https://stackoverflow.com/questions/30255539/using-constant-for-defining-property-format-in-mvc>

    \* <https://stackoverflow.com/questions/15880443/creating-a-string-in-a-web-config-and-use-it-in-a-web-api>

\* Jquery Validation fix for date format dd/mm/yyyy in Chrome and Safari

    \* <https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

    \* <http://devdens.blogspot.com.au/2011/11/jquery-validation-fix-for-date-format_29.html>

    \* <https://github.com/globalizejs/globalize#readme>

    \* <https://www.nuget.org/packages/jquery-globalize/1.3.0>

    \* <https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

    \* <https://stackoverflow.com/questions/14020117/dd-mm-yyyy-date-validation-error-with-google-chrome>

\* Remote Validation in mvc affected Edit

    \* [https://forums.asp.net/t/2131813.aspx?Remote+Validation+in+mvc+affected+Edit](https://forums.asp.net/t/2131813.aspx?Remote%2BValidation%2Bin%2Bmvc%2Baffected%2BEdit)

\* Regular expression

    \* <https://regexr.com/>

    \* <https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>

\* Range with Date issue

    \* <https://blogs.msdn.microsoft.com/stuartleeks/2011/01/25/asp-net-mvc-3-integrating-with-the-jquery-ui-date-picker-and-adding-a-jquery-validate-date-range-validator/>

    \* [https://forums.asp.net/t/1831436.aspx?Validating+a+datetime+range+in+EF+annotation](https://forums.asp.net/t/1831436.aspx?Validating%2Ba%2Bdatetime%2Brange%2Bin%2BEF%2Bannotation)

    \* <https://stackoverflow.com/questions/17321948/is-there-a-rangeattribute-for-datetime>

    \* <https://archive.codeplex.com/?p=foolproof>

    \* <https://stackoverflow.com/questions/13183647/date-range-validation-with-entity-framework-4-data-annotations>

=========================================

更正1:

Tutorial 20-03的影片

時間在  12:03

在介紹RegularExpression時提到

ErrorMessage不要放在ResoureFile裡,

而是要放在Global Variable。

這邊我講的不夠精確

其實你可以這樣寫的

[RegularExpression(@"^[0-9][0-9]?$|^100$", ErrorMessageResourceName = "IntPositive", ErrorMessageResourceType = typeof(Resource))]

或是你也可以參考這邊這樣寫

<https://stackoverflow.com/questions/2523579/how-to-use-regularexpression-dataannotation-with-resource-file>

另外

黑暗執行緒有一篇文章是教人如何在excel裡維護多國語系，再生成資源檔，還不錯。也可以參考

1. OnlineGame DB

1.1. TSQL

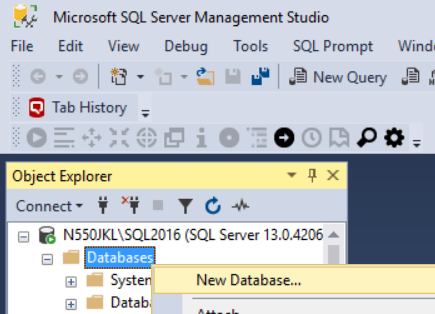
In SQL server Management Studio (SSMS)

Database --> Right Click --> New Database -->

In General Tab -->

Name: OnlineGame

In options Tab --> Recovery model : Simple



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

--1. Drop if it exists

--Drop Table if it exists.

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'Gamer' ) )

    BEGIN

        TRUNCATE TABLE Gamer;

        DROP TABLE Gamer;

    END;

GO -- Run the previous command and begins new batch

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'Team' ) )

    BEGIN

        TRUNCATE TABLE Team;

        DROP TABLE Team;

    END;

GO -- Run the previous command and begins new batch

--2. Create Table

CREATE TABLE Team

    (

      Id INT PRIMARY KEY

             IDENTITY(1, 1)

             NOT NULL ,

      [Name] NVARCHAR(100) NOT NULL

    );

GO -- Run the previous command and begins new batch

CREATE TABLE Gamer

    (

      Id INT PRIMARY KEY

             IDENTITY(1, 1)

             NOT NULL ,

      [Name] NVARCHAR(100) NOT NULL ,

      Gender NVARCHAR(10) NOT NULL ,

      City NVARCHAR(50) NOT NULL ,

      DateOfBirth DATETIME NOT NULL ,

      EmailAddress NVARCHAR(100) NOT NULL,

      Score INT NOT NULL,

      ProfileUrl NVARCHAR(100) ,

      GameMoney INT NOT NULL,

      RolePhoto NVARCHAR(100) ,

      RolePhotoAltText NVARCHAR(100) ,

      TeamId INT FOREIGN KEY REFERENCES Team ( Id ) NOT NULL

    );

GO -- Run the previous command and begins new batch

--3. Insert Data

INSERT  Team

VALUES  ( N'TeamOne' );

INSERT  Team

VALUES  ( N'TeamTwo' );

INSERT  Team

VALUES  ( N'TeamThree' );

GO -- Run the previous command and begins new batch

INSERT  Gamer

VALUES  ( N'NameOne ABB', N'Male', N'City01', '1979/4/28', '[1@AAA.com](mailto:1@AAA.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 1000,

          '~/Photos/Name01.png', 'Name01RolePhoto', 1 );

INSERT  Gamer

VALUES  ( N'NameTwo CDDE', N'Female', N'City03', '1981/7/24', '[2@BBB.com](mailto:2@BBB.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 1500,

          '~/Photos/Name02.png', 'Name02RolePhoto', 2 );

INSERT  Gamer

VALUES  ( N'NameThree FIJK', N'Female', N'City01', '1984/12/5', '[3@CCCC.com](mailto:3@CCCC.com)',

          3500, '<https://ithandyguytutorial.blogspot.com.au/>', 4000,

          '~/Photos/Name03.png', 'Name03RolePhoto', 3 );

INSERT  Gamer

VALUES  ( N'NameFour LMOPPQ', N'Male', N'City02', '1983/5/29', '[4@DD.com](mailto:4@DD.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 2500,

          '~/Photos/Name04.png', 'Name04RolePhoto', 1 );

INSERT  Gamer

VALUES  ( N'NameFive QRSTT', N'Male', N'City01', '1979/6/20', '[5@EEE.com](mailto:5@EEE.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 3500,

          '~/Photos/Name05.png', 'Name05RolePhoto', 3 );

INSERT  Gamer

VALUES  ( N'NameSix TUVVX', N'Female', N'City03', '1984/5/15', '[6@FF.com](mailto:6@FF.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 2500,

          '~/Photos/Name06.png', 'Name06RolePhoto', 3 );

INSERT  Gamer

VALUES  ( N'NameSeven XYZZXX', N'Female', N'City01', '1986/4/29', '[7@GGGG.com](mailto:7@GGGG.com)',

          3500, '<https://ithandyguytutorial.blogspot.com.au/>', 4550,

          '~/Photos/Name07.png', 'Name07RolePhoto', 2 );

INSERT  Gamer

VALUES  ( N'NameEight ABBCDE', N'Male', N'City02', '1985/7/28', '[8@HH.com](mailto:8@HH.com)', 3500,

          '<https://ithandyguytutorial.blogspot.com.au/>', 3550,

          '~/Photos/Name08.png', 'Name08RolePhoto', 1 );

INSERT  Gamer

VALUES  ( N'NameNine QRSTTUVXX', N'Male', N'City02', '1983/4/16', '[9@IIII.com](mailto:9@IIII.com)',

          3500, '<https://ithandyguytutorial.blogspot.com.au/>', 2510,

          '~/Photos/Name09.png', 'Name09RolePhoto', 1 );

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.





Graphical user interface, text, application

Description automatically generated



2. OnlineGame Solution

2.1. OnlineGame Solution

File --> New --> Project... -->

Other Project Types --> Visual Studio Solutions -->  Blank Solution

-->

Name: **OnlineGame**

Graphical user interface, application

Description automatically generated

2.2. OnlineGame.Web

Solutions Name --> Add --> New Project -->

Visual C# --> Web --> [ASP.NET](http://asp.net/)Web Application (.Net Framework)

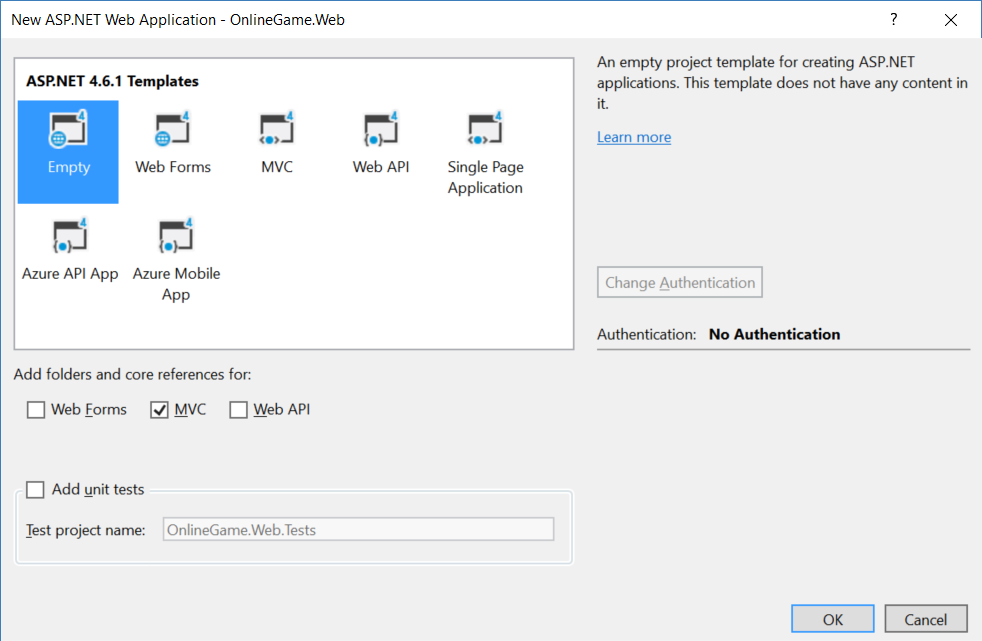
-->

Name: **OnlineGame.Web**

Empty --> Select "MVC" --> OK

Graphical user interface, text, application

Description automatically generated



2.3. Regular expression

Reference:

<https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>

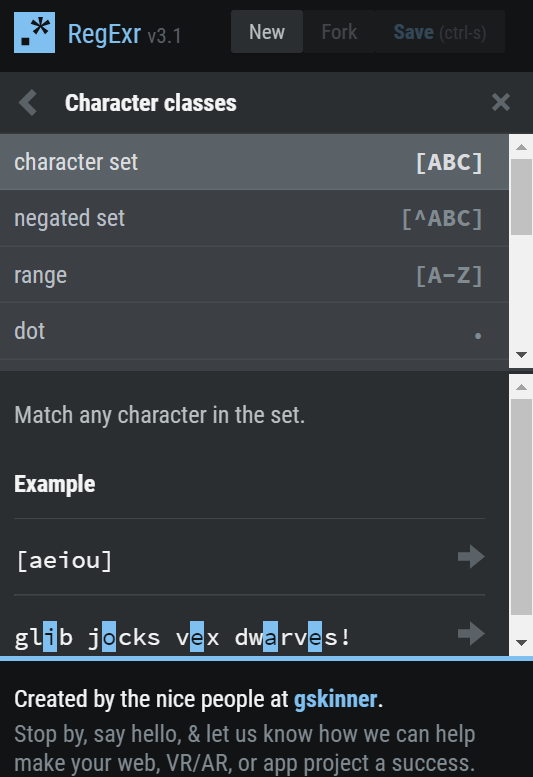
<https://regexr.com/>

Go to

<https://regexr.com/>

Click "Reference" --> Click "Character classes"

Click all of examples and have a look.



Go to

<https://regexr.com/>

Click "**Reference**" --> Click "**Anchors**"

Click all of examples.

Graphical user interface, application, Teams

Description automatically generated

If you don't understand any of the regular expression,

then at least you have to learn how to use it.

Go to

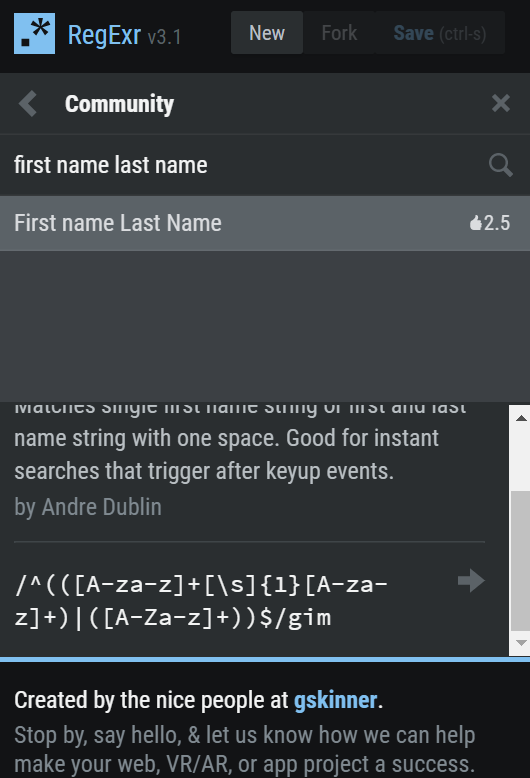
<https://regexr.com/>

Click "**Community**" --> Search "**First Name, Last Name**"

**^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$**

Matches single first name string or first and last name string with one space.

Good for instant searches that trigger after keyup events.



Go to

<https://regexr.com/>

Click "**Community**" --> Search "**Email**"

**^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$**

Acceptable

[abc1.1@domain.com](mailto:abc1.1@domain.com)

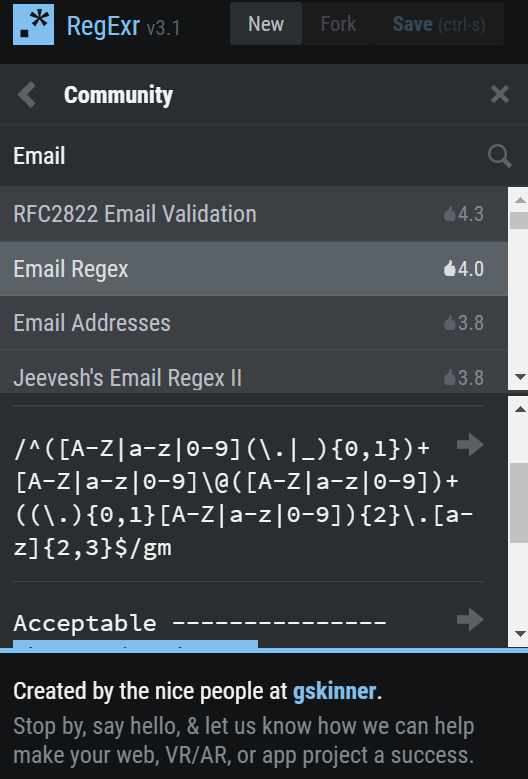
[abc.abc@domain.com](mailto:abc.abc@domain.com)

[abc\_41.15768@domain.com](mailto:abc_41.15768@domain.com)

[12345@domain.com](mailto:12345@domain.com)

[12.12.23@domain.com](mailto:12.12.23@domain.com)

[ABC.ABC.ABC.ABC0046@domain.com](mailto:ABC.ABC.ABC.ABC0046@domain.com)



2.4. Web.config

2.4.1. Web.config



<appSettings>

  <add key="webpages:Version" value="3.0.0.0" />

  <add key="webpages:Enabled" value="false" />

  <add key="ClientValidationEnabled" value="true" />

  <add key="UnobtrusiveJavaScriptEnabled" value="true" />

  <add key="ApplicationName" value="Online Game Web" />

</appSettings>

<system.web>

  <caching>

    <outputCacheSettings>

      <outputCacheProfiles>

        <clear />

        <add name="outputCacheProfile1" duration="60" varyByParam="none" />

      </outputCacheProfiles>

    </outputCacheSettings>

  </caching>

  <customErrors mode="On">

    <error statusCode="401" redirect="Error/UnauthorizedError" />

    <error statusCode="404" redirect="Error/NotFound" />

    <error statusCode="500" redirect="Error/InternalServerError" />

  </customErrors>

  <globalization culture="en-au" uiCulture="en-AU" />

  <compilation debug="true" targetFramework="4.6.1" />

  <httpRuntime targetFramework="4.6.1" />

</system.web>

2.4.2. The way to use key of Web.config in a class

private static string \_applicationName = System.Configuration.ConfigurationManager.AppSettings["ApplicationName"];

2.4.3. The way to use key of in view

@Html.ActionLink(System.Configuration.ConfigurationManager.AppSettings["ApplicationName"], "Index", "Gamer", new { area = "" }, new { @class = "navbar-brand" })

2.5. WebShare/WebShareConst.cs

namespace OnlineGame.Web.WebShare

{

    public class WebShareConst

    {

        public const string DateFormat = "dd/MM/yyyy";

        public const string DateFormatJavascriptString = "dd/mm/yy";

        public const string DateStringFormat = "{0:dd/MM/yyyy}";

        //RegularExpression

        //<https://regexr.com/>

        public const string FirstNameLastNameRegularExpression = @"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$";

        public const string EmailRegularExpression = @"^[\w-\.\_\+%]+@(?:[\w-]+\.)+[\w]{2,6}$";

        //Validation String

        public const string EmailNotValid = "Email is not valid";

        public const string FirstNameLastNameNotValid = "Please enter first name or first name and last name.";

        public const string EmailHasBeenTaken = "The email has already been taken.";

        public const string ValidationSummaryTitleString = "Please check the following fields.";

    }

}

2.6. Global.asax.cs

using System.Web.Mvc;

using System.Web.Routing;

using OnlineGame.Web;

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();

            //Register HandleErrorAttribute to global filter

            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

            //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);

            ////Enable Unobtrusive validation

            //HtmlHelper.UnobtrusiveJavaScriptEnabled = true;

            //HtmlHelper.ClientValidationEnabled = true;

        }

    }

}

2.7. 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](http://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](http://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](http://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](http://asp.net/) trace, trace.axd.

If you do not have

// routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

then you can not enable the trace.axd.

\*/

2.8. Error Handling

2.8.1. Web.config

Check the following code in  Web.config

<customErrors mode="On">

  <error statusCode="401" redirect="Error/UnauthorizedError" />

  <error statusCode="404" redirect="Error/NotFound" />

  <error statusCode="500" redirect="Error/InternalServerError" />

</customErrors>

2.8.2. App\_Start/FilterConfig.cs

using System.Web.Mvc;

namespace OnlineGame.Web

{

    public class FilterConfig

    {

        public static void RegisterGlobalFilters(GlobalFilterCollection filters)

        {

            filters.Add(new HandleErrorAttribute());

        }

    }

}

/\*

1.

Register Customized Error View

1.1.

Register HandleErrorAttribute to global filter

In Global.asax,

//FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

We pass the GlobalFilters.Filters to

//public static void RegisterGlobalFilters(GlobalFilterCollection filters)

Here, we register "HandleErrorAttribute" to global filter.

1.2.

In Web.Config, add the customErrors mode="On"

//<system.web>

//    <customErrors mode="On">

//    </customErrors>

1.3.

Create error view, Views/Shared/Error.cshtml

\*/

2.8.3. Global.asax.cs

Check the following code in  Global.asax.cs

//Register HandleErrorAttribute to global filter

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

2.8.4. Controllers/ErrorController.cs

using System.Web.Mvc;

namespace OnlineGame.Web.Controllers

{

    public class ErrorController : Controller

    {

        //error statusCode="401"

        [HttpGet]

        public ActionResult UnauthorizedError()

        {

            return View();

        }

        //error statusCode="404"

        [HttpGet]

        public ActionResult NotFound()

        {

            return View();

        }

        //error statusCode="500"

        [HttpGet]

        public ActionResult InternalServerError()

        {

            return View();

        }

    }

}

/\*

1.

In the Web.config

//<customErrors mode="On" defaultRedirect="Error/DefaultError">

//    <error statusCode="401" redirect="Error/UnauthorizedError" />

//    <error statusCode="404" redirect="Error/NotFound" />

//    <error statusCode="500" redirect="Error/InternalServerError" />

//</customErrors>

We notice that it will still show the Views/Shared/Error.cshtml

when exception occurs.

Thus, we can delete Views/Shared/DefaultError.cshtml.

We also can delete DefaultError() in ErrorController.cs

In the Web.config, we can set as the following.

//<customErrors mode="On">

//    <error statusCode="401" redirect="Error/UnauthorizedError" />

//    <error statusCode="404" redirect="Error/NotFound" />

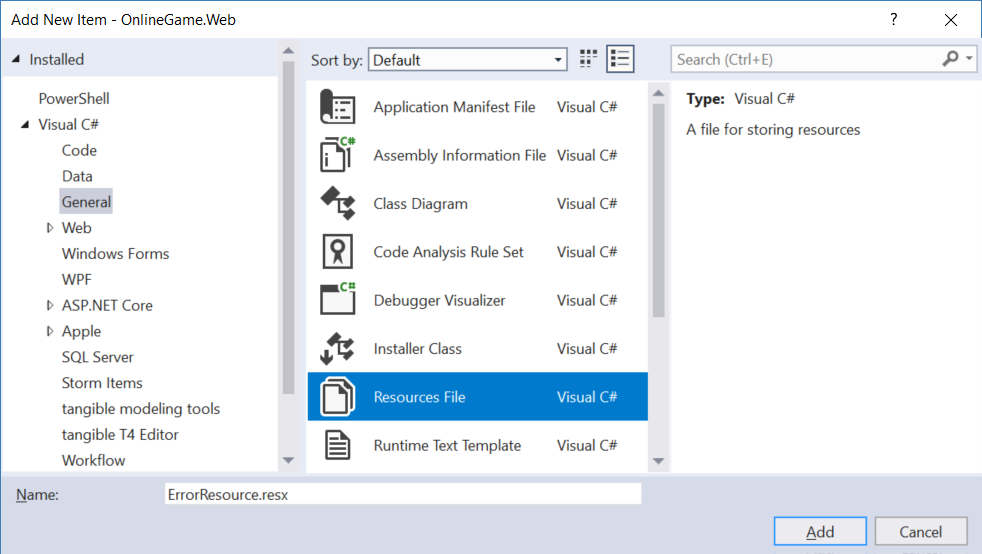
//    <error statusCode="500" redirect="Error/InternalServerError" />

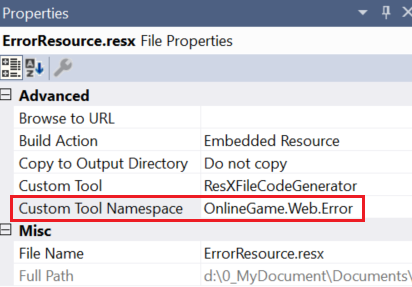
//</customErrors>

\*/

2.8.5. Error View

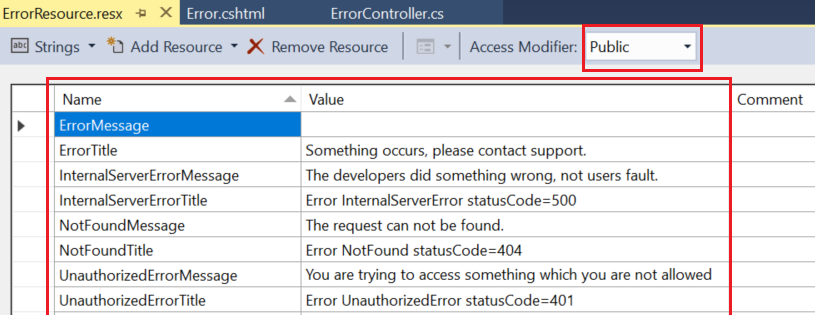
2.8.5.1. Views/Shared/Resource/ErrorResource.resx (NameSpace: OnlineGame.Web.Error)





NameSpace

**OnlineGame.Web.Error**



|  |  |
| --- | --- |
| ErrorMessage |  |
| ErrorTitle | Something occurs, please contact support. |
| InternalServerErrorMessage | The developers did something wrong, not users fault. |
| InternalServerErrorTitle | Error InternalServerError statusCode=500 |
| NotFoundMessage | The request can not be found. |
| NotFoundTitle | Error NotFound statusCode=404 |
| UnauthorizedErrorMessage | You are trying to access something which you are not allowed to access. |
| UnauthorizedErrorTitle | Error UnauthorizedError statusCode=401 |

ErrorMessage

ErrorTitle

Something occurs, please contact support.

InternalServerErrorMessage

The developers did something wrong, not users fault.

InternalServerErrorTitle

Error InternalServerError statusCode=500

NotFoundMessage

The request can not be found.

NotFoundTitle

Error NotFound statusCode=404

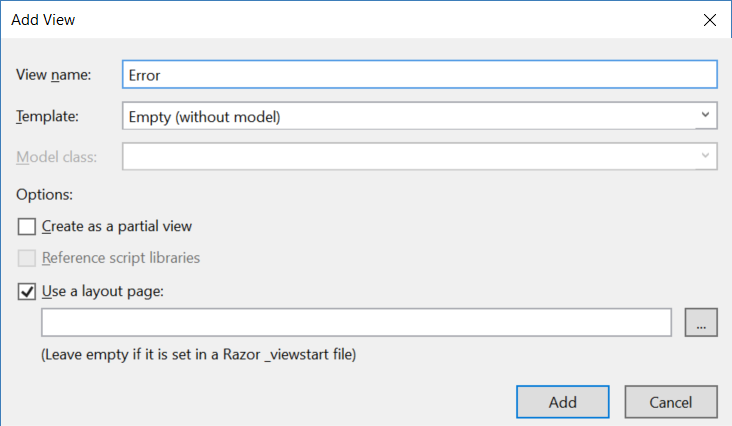
UnauthorizedErrorMessage

You are trying to access something which you are not allowed to access.

UnauthorizedErrorTitle

Error UnauthorizedError statusCode=401

2.8.5.2. Views/Shared/Error.cshtml



@using OnlineGame.Web.Error

@{

    ViewBag.Title = "Error";

}

<h2>@ErrorResource.ErrorTitle</h2>

<div>

    @ErrorResource.ErrorMessage

</div>

2.8.5.3. Views/Shared/UnauthorizedError.cshtml

@using OnlineGame.Web.Error

@{

    ViewBag.Title = "UnauthorizedError";

}

<h2>@ErrorResource.UnauthorizedErrorTitle</h2>

<div>

    @ErrorResource.UnauthorizedErrorMessage

</div>

<http://localhost/onlinegame.web/Error/UnauthorizedError>



2.8.5.4. Views/Shared/NotFound.cshtml

@using OnlineGame.Web.Error

@{

    ViewBag.Title = "NotFound";

}

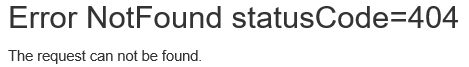
<h2>@ErrorResource.NotFoundTitle</h2>

<div>

    @ErrorResource.NotFoundMessage

</div>

<http://localhost/onlinegame.web/Error/NotFound>



2.7.5.5. Views/Shared/InternalServerError.cshtml

@using OnlineGame.Web.Error

@{

    ViewBag.Title = "InternalServerError";

}

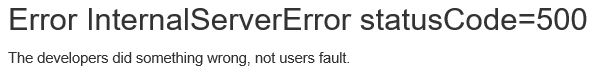
<h2>@ErrorResource.InternalServerErrorTitle</h2>

<div>

    @ErrorResource.InternalServerErrorMessage

</div>

<http://localhost/onlinegame.web/Error/InternalServerError>



2.9. WebShare/Attribute

2.9.1. CustomizeCacheAttribute

2.9.1.1. WebShare/Attribute/CustomizeCacheAttribute.cs

using System.Web.Mvc;

using System.Web.Configuration;

namespace OnlineGame.Web.WebShare.Attribute

{

    public class CustomizeCacheAttribute : OutputCacheAttribute

    {

        public CustomizeCacheAttribute(string cacheProfileName)

        {

            OutputCacheSettingsSection cacheSettings =

                (OutputCacheSettingsSection)WebConfigurationManager

                .GetSection("system.web/caching/outputCacheSettings");

            OutputCacheProfile cacheProfile = cacheSettings.OutputCacheProfiles[cacheProfileName];

            Duration = cacheProfile.Duration;

            VaryByParam = cacheProfile.VaryByParam;

            VaryByCustom = cacheProfile.VaryByCustom;

        }

    }

}

/\*

In Web.config

//<system.web>

//    <caching>

//        <outputCacheSettings>

//        <outputCacheProfiles>

//            <clear/>

//            <add name="outputCacheProfile1" duration="60" varyByParam="none"/>

//        </outputCacheProfiles>

//        </outputCacheSettings>

//    </caching>

//    <customErrors mode="On">

//        <error statusCode="401" redirect="Error/UnauthorizedError" />

//        <error statusCode="404" redirect="Error/NotFound" />

//        <error statusCode="500" redirect="Error/InternalServerError" />

//    </customErrors>

//    <globalization culture="en-au" />

//    <compilation debug="true" targetFramework="4.6.1" />

//    <httpRuntime targetFramework="4.6.1" />

//</system.web>

\*/

2.9.1.2. The way to use WebShare/Attribute/CustomizeCacheAttribute.cs

The following the controller action method to show you how to use CustomizeCacheAttribute

[HttpGet]

//[OutputCache(Duration = 60)]

[OutputCache(CacheProfile = "outputCacheProfile1")]

public async Task<ActionResult> Index4()

{

    return View(await db.Gamers.ToListAsync());

}

//[ChildActionOnly] make the action to be accessible only by a child request,

//so no one can make a direct URL request to this action.

[ChildActionOnly]

[HttpGet]

//[OutputCache(Duration = 60)]

//[OutputCache(CacheProfile = "outputCacheProfile1")]   //This will thrwo exception

[CustomizeCache("outputCacheProfile1")]

public string GetGamerCount2()

{

    return $"Gamer Count = {db.Gamers.Count()} At {DateTime.Now}";

}

2.9.2. DateRangeAttribute, BetweenMinimumDateAndTodayAttribute, BeforeTodayAttribute

2.9.2.1. WebShare/Attribute/DateRangeAttribute.cs

using System;

using System.ComponentModel.DataAnnotations;

using System.Globalization;

namespace OnlineGame.Web.WebShare.Attribute

{

    public class DateRangeAttribute : ValidationAttribute

    {

        //private const string DateFormat = "dd/MM/yyyy";

        //private static string DateFormat = System.Configuration.ConfigurationManager.AppSettings["DateDormat"];

        private const string DateFormat = WebShareConst.DateFormat;

        private const string DefaultErrorMessage = "{0}, '{1}' must be a date between {2:d} and {3:d}.";

        public DateTime MinDate { get; set; }

        public DateTime MaxDate { get; set; }

        public DateRangeAttribute(string minDate, string maxDate)

            : base(DefaultErrorMessage)

        {

            MinDate = ParseDate(minDate);

            MaxDate = ParseDate(maxDate);

        }

        public override bool IsValid(object value)

        {

            if (!(value is DateTime))

            {

                return false;

            }

            DateTime dateValue = (DateTime)value;

            return MinDate <= dateValue && dateValue <= MaxDate;

        }

        public override string FormatErrorMessage(string name)

        {

            return String.Format(CultureInfo.CurrentCulture,

                ErrorMessageString, name, DateFormat, MinDate, MaxDate);

        }

        private static DateTime ParseDate(string dateValue)

        {

            return DateTime.ParseExact(dateValue, DateFormat,

                CultureInfo.InvariantCulture);

        }

    }

}

2.9.2.2. WebShare/Attribute/BetweenMinimumDateAndTodayAttribute.cs

using System;

using System.ComponentModel.DataAnnotations;

using System.Globalization;

namespace OnlineGame.Web.WebShare.Attribute

{

    public class BetweenMinimumDateAndTodayAttribute : ValidationAttribute

    {

        //private const string DateFormat = "dd/MM/yyyy";

        //private static string DateFormat = System.Configuration.ConfigurationManager.AppSettings["DateDormat"];

        private const string DateFormat = WebShareConst.DateFormat;

        private const string DefaultErrorMessage = "{0}, '{1}' must be a date between {2:d} and {3:d}.";

        public DateTime MinDate { get; set; }

        private DateTime MaxDate = DateTime.Now;

        public BetweenMinimumDateAndTodayAttribute(string minDate)

            : base(DefaultErrorMessage)

        {

            MinDate = ParseDate(minDate);

        }

        public override bool IsValid(object value)

        {

            if (!(value is DateTime))

            {

                return false;

            }

            DateTime dateValue = (DateTime)value;

            return MinDate <= dateValue && dateValue <= MaxDate;

        }

        public override string FormatErrorMessage(string name)

        {

            return String.Format(CultureInfo.CurrentCulture,

                ErrorMessageString, DateFormat, name, MinDate, MaxDate);

        }

        private static DateTime ParseDate(string dateValue)

        {

            return DateTime.ParseExact(dateValue, DateFormat,

                CultureInfo.InvariantCulture);

        }

    }

}

2.9.2.3. WebShare/Attribute/BeforeTodayAttribute.cs

using System;

using System.ComponentModel.DataAnnotations;

using System.Globalization;

namespace OnlineGame.Web.WebShare.Attribute

{

    public class BeforeTodayAttribute : ValidationAttribute

    {

        //private const string DateFormat = "dd/MM/yyyy";

        //private static string DateFormat = System.Configuration.ConfigurationManager.AppSettings["DateDormat"];

        private const string DateFormat = WebShareConst.DateFormat;

        private const string DefaultErrorMessage = "{0}, '{1}' must be a date between {2:d} and {3:d}.";

        private DateTime MinDate = DateTime.MinValue;

        private DateTime MaxDate = DateTime.Now;

        public BeforeTodayAttribute()

            : base(DefaultErrorMessage)

        {

        }

        public override bool IsValid(object value)

        {

            if (!(value is DateTime))

            {

                return false;

            }

            DateTime dateValue = (DateTime)value;

            return MinDate <= dateValue && dateValue <= MaxDate;

        }

        public override string FormatErrorMessage(string name)

        {

            return String.Format(CultureInfo.CurrentCulture,

                ErrorMessageString, DateFormat, name, MinDate, MaxDate);

        }

        private static DateTime ParseDate(string dateValue)

        {

            return DateTime.ParseExact(dateValue, DateFormat,

                CultureInfo.InvariantCulture);

        }

    }

}

2.9.2.4. The way to use DateRangeAttribute, BetweenMinimumDateAndTodayAttribute, BeforeTodayAttribute

The following the controller action method to show you how to use this attributes

//[Range(typeof(DateTime), "1/1/1970", "1/1/2001", ErrorMessage = "Date is out of Range")]    //Error - Client Side validation will never pass

//[Range(typeof(DateTime), "1/1/1970", "1/1/2001")] //Error - Client Side validation will never pass

//[DateRange("01/01/1970", "01/01/2001")]

//[DateRange("01/01/1970", DateTime.Now.ToShortDateString())] //Error :  An attribute argument must be a constant expression

//[BetweenMinimumDateAndToday("01/01/1970")]

[BeforeToday]

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

[DisplayFormat(DataFormatString = WebShareConst.DateStringFormat, ApplyFormatInEditMode = true)]

[Required]

public DateTime DateOfBirth { get; set; }

2.9.3. RemoteClientServerAttribute.cs

2.9.3.1. WebShare/Attribute/RemoteClientServerAttribute.cs

using System;

using System.Linq;

using System.Web.Mvc;

using System.ComponentModel.DataAnnotations;

using System.Reflection;

namespace OnlineGame.Web.WebShare.Attribute

{

    public class RemoteClientServerAttribute : RemoteAttribute

    {

        //Implement IsValid() will get the server side validation.

        protected override ValidationResult IsValid(object value, ValidationContext validationContext)

        {

            //1.

            //Using reflection to get the controller that is used by this attribute.

            //In all types of the Executing Assembly, I want the type

            //which the type name is equals to the controller name in route data.

            Type controller = Assembly.GetExecutingAssembly().GetTypes()

                .FirstOrDefault(type =>

                String.Equals(type.Name,

                $"{RouteData["controller"].ToString()}Controller",

                StringComparison.CurrentCultureIgnoreCase));

            if (controller != null)

            {

                //2.

                // Get the action method that is used by this attribute and has validation logic.

                MethodInfo action = controller.GetMethods()

                    .FirstOrDefault(method =>

                    String.Equals(method.Name,

                    RouteData["action"].ToString(),

                    StringComparison.CurrentCultureIgnoreCase));

                if (action != null)

                {

                    // Create an instance of the controller class

                    object instance = Activator.CreateInstance(controller);

                    // Invoke the action method that is used by this attribute and has validation logic.

                    // action.Invoke(object obj, object[] parameters)

                    //In our case, it will invoke "IsEmailAvailable" action of "GamerController".

                    object response = action.Invoke(instance, new[] { value });

                    //if (response is JsonResult), then get data.

                    //if (response is JsonResult)

                    //object jsonData = ((JsonResult)response).Data;

                    var result = response as JsonResult;

                    object jsonData = result?.Data;

                    //If the data is true, it means pass the validation.

                    //Otherwise return new ValidationResult(ErrorMessage);

                    if (jsonData is bool)

                    {

                        return (bool)jsonData ?

                            ValidationResult.Success :

                            new ValidationResult(ErrorMessage);

                    }

                }

            }

            //3.

            //3.1.

            //// return ValidationResult.Success;

            //If we don't find the controller which passed from the attribute,

            //or if we don't find the action that used by the attribute and has validation logic.

            //Then we return validation pass.

            //That means we ignore this validation attribute.

            //3.2.

            // If you want the validation to fail

            ////return new ValidationResult(base.ErrorMessageString);

            return ValidationResult.Success;

        }

        public RemoteClientServerAttribute(string routeName)

            : base(routeName)

        {

        }

        public RemoteClientServerAttribute(string action, string controller)

            : base(action, controller)

        {

        }

        public RemoteClientServerAttribute(string action, string controller,

            string areaName) : base(action, controller, areaName)

        {

        }

    }

}

/\*

1.

//Type controller = Assembly.GetExecutingAssembly().GetTypes()

//    .FirstOrDefault(type =>

//    String.Equals(type.Name,

//    $"{RouteData["controller"].ToString()}Controller",

//    StringComparison.CurrentCultureIgnoreCase));

Using reflection to get the controller that is used by this attribute.

In all types of the Executing Assembly, I want the type

which the type name is equals to the controller name in route data.

1.1.

//Assembly.GetExecutingAssembly().GetTypes().FirstOrDefault

In all types of the Executing Assembly

1.2.

//$"{RouteData["controller"].ToString()}Controller"

{RouteData["controller"].ToString()} is the controller name in route data.

//$"{RouteData["controller"].ToString()}Controller"

{NameOfController}+"Controller".

2.

//MethodInfo action = controller.GetMethods()

//    .FirstOrDefault(method =>

//    String.Equals(method.Name,

//    RouteData["action"].ToString(),

//    StringComparison.CurrentCultureIgnoreCase));

2.1.

Get the action method that is used by this attribute and has validation logic.

2.2.

//RouteData["action"].ToString()

RouteData["action"].ToString() is the action name in route data.

The action method uses this attribute and has validation logic.

\*/

2.9.3.2. The way to use RemoteClientServerAttribute

Please check the Models/Gamer/GamerMetaData.cs

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = "The email has already been taken.")]

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

//[RemoteClientServer("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

////Remote attribute can only do the client side validation.

////RemoteClientServer is a customize attribute which can do both client side and server side validation.

////Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

////If you really want to use it, please use two different model classes for Edit and Create mode.

//[RegularExpression(@"^[\w-\.\_\+%]+@(?:[\w-]+\.)+[\w]{2,6}$", ErrorMessage = "Email is not valid")]  //EmailAddress Regular Expression

//[RegularExpression(WebShareConst.EmailRegularExpression, ErrorMessage = "Email is not valid")]  //EmailAddress Regular Expression

[RegularExpression(WebShareConst.EmailRegularExpression, ErrorMessage = WebShareConst.EmailNotValid)]  //EmailAddress Regular Expression

[Required]

public string EmailAddress { get; set; }

/\*

6.

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = "The email has already been taken.")]

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

//[RemoteClientServer("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

//// Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

//// You may use two different model class for create and edit mode.

//public string EmailAddress { get; set; }

6.1.

Remote attribute can only do the client side validation.

RemoteClientServer is a customize attribute which can do both client side and server side validation.

Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

If you really want to use it, please use two different model classes for Edit and Create mode.

6.2.

RemoteAttribute uses AJAX to make an asynchronous call to the server-side method.

If the user disables javascript function of the browser, then it will not work.

Thus, we should also have server-side validation.

6.3.

server-side validation.

In the Create HttpPost Method.

It is hard to read that validation logic is in the controller.

Using validation attributes is always the preferred method.

You may use two different model class for create and edit mode.

//[HttpPost]

//[ValidateAntiForgeryToken]

//public async Task<ActionResult> Create([Bind(Include = "Id,Name,Gender,City,DateOfBirth,EmailAddress,ConfirmEmailAddress,Score,ProfileUrl,GameMoney,RolePhoto,RolePhotoAltText,TeamId")] Gamer gamer)

//{

//    //If the Email already exists, then add Model validation error

//    if (db.Gamers.Any(g => g.EmailAddress == gamer.EmailAddress))

//    {

//        //AddModelError(Key, ErrorMessage)

//        ModelState.AddModelError("EmailAddress", WebShareConst.EmailHasBeenTaken);

//    }

//    //It is hard to read that validation logic is in controller.

//    //Using validation attributes is always preferred method.

//    if (!ModelState.IsValid)

//    {

//        ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name", gamer.TeamId);

//        return View(gamer);

//    }

//    db.Gamers.Add(gamer);

//    await db.SaveChangesAsync();

//    return RedirectToAction("Index");

//}

6.4.

Another way to has both server side and client side validation,

you have to create your own RemoteClientServerAttribute which extend RemoteAttribute.

\*/

2.10.  Enable client side validation

When we use automatically generate code,

it will also generate the code to enable client side validation.

Let's discuss how it works.

2.10.1. Enable Unobtrusive Validation

Use one of the following methods can enable Unobtrusive Validation.

In our case, we just use the "Web.config" to enable Unobtrusive Validation

2.10.1.1. Enable Unobtrusive Validation Method1 - Web.config

<appSettings>

  <add key="webpages:Version" value="3.0.0.0" />

  <add key="webpages:Enabled" value="false" />

  <add key="ClientValidationEnabled" value="true" />

  <add key="UnobtrusiveJavaScriptEnabled" value="true" />

  <add key="ApplicationName" value="Online Game Web" />

</appSettings>

2.10.1.2. Enable client side validation Method2 - Global.asax.cs

using System.Web.Mvc;

using System.Web.Routing;

using OnlineGame.Web;

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();

            //Register HandleErrorAttribute to global filter

            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

            //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);

            ////Enable Unobtrusive validation

            //HtmlHelper.UnobtrusiveJavaScriptEnabled = true;

            //HtmlHelper.ClientValidationEnabled = true;

        }

    }

}

2.10.1.3. Enable client side validation Method3 - In the View

@{

    ViewBag.Title = "Edit";

    ////Enable client side validation

    //Html.EnableClientValidation(true);

    //Html.EnableUnobtrusiveJavaScript(true);

}

2.10.2. In the View

In our case, please check the following file

Views/Gamer/Create.cshtml

Views/Gamer/Edit.cshtml

In our case, we already move these 3 lines to Views/Shared/\_Layout.cshtml

<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>

2.10.3. How it works

2.10.3.1. How it works - Models/Gamer/Gamer.cs

[MetadataType(typeof(GamerMetaData))]

public partial class Gamer

2.10.3.2. How it works - Models/Gamer/GamerMetaData.cs

Look At Name Property.

[StringLength(20, MinimumLength = 2)]

//[RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$")] //First Name Last Name

//[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression)] //First Name Last Name

//[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = "Please enter first name or first name and last name.")] //First Name Last Name

[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = WebShareConst.FirstNameLastNameNotValid)] //First Name Last Name

[Required]

public string Name { get; set; }

2.10.3.3. How it works - Edit.cshtml

Run the Edit.cshtml, and inspect the html when it is running.

<input class="form-control text-box single-line valid"

data-val="true"

data-val-length="The field Name must be a string with a minimum length of 2 and a maximum length of 20." data-val-length-max="20" data-val-length-min="2"

data-val-regex="Please enter first name or first name and last name." data-val-regex-pattern="^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$"

data-val-required="The Name field is required."

id="Name" name="Name" type="text" value="NameOne ABB">

-------------------------------

A.

data-val="true"

means the unobtrusive validation of this element is enabled.

B.

[StringLength(20, MinimumLength = 2)]

-->

data-val-length="The field Name must be a string with a minimum length of 2 and a maximum length of 20." data-val-length-max="20" data-val-length-min="2"

C.

//[RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$")] //First Name Last Name

//[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression)] //First Name Last Name

//[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = "Please enter first name or first name and last name.")] //First Name Last Name

[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = WebShareConst.FirstNameLastNameNotValid)] //First Name Last Name

-->

data-val-regex="Please enter first name or first name and last name." data-val-regex-pattern="^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$"

D.

[Required]

-->

data-val-required="The Name field is required."

2.11.  Enable client side validation Summary

2.11.1. \_Layout.cshtml

A.

"Site.css" must below "bootstrap.min.css",

so "Site.css" will override "bootstrap.min.css".

B.

The "Application Name" action link should set to Gamer controller and Index action.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>@ViewBag.Title - My [ASP.NET](http://asp.net/) Application</title>

    <link href="~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />

    <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />

    <script src="~/Scripts/modernizr-2.6.2.js"></script>

</head>

<body>

    <div class="navbar navbar-inverse navbar-fixed-top">

        <div class="container">

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                @Html.ActionLink(System.Configuration.ConfigurationManager.AppSettings["ApplicationName"], "Index", "Gamer", new { area = "" }, new { @class = "navbar-brand" })

            </div>

            <div class="navbar-collapse collapse">

                <ul class="nav navbar-nav"></ul>

            </div>

        </div>

    </div>

    <div class="container body-content">

        @RenderBody()

        <hr />

        <footer>

            <p>&copy; @DateTime.Now.Year - My [ASP.NET](http://asp.net/) Application</p>

        </footer>

    </div>

    <script src="~/Scripts/jquery-1.10.2.min.js"></script>

    <script src="~/Scripts/bootstrap.min.js"></script>

   <script src="~/Scripts/jquery-ui-1.12.1.min.js"></script>

    <script src="~/Scripts/jquery.validate.min.js"></script>

    <script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

    <script src="~/Scripts/operation.js"></script>

</body>

</html>

2.11.2. Content/Site.css

body {

    padding-top: 50px;

    padding-bottom: 20px;

}

/\* Set padding to keep content from hitting the edges \*/

.body-content {

    padding-left: 15px;

    padding-right: 15px;

}

/\* Set width on the form input elements since they're 100% wide by default \*/

input,

select,

textarea {

    max-width: 280px;

}

/\*Override Bootstrap ------------------- \*/

/\*.text-danger {

    color: #b94a48;

}\*/

/\*validation summary ------------------- \*/

.validation-summary-valid {

    display: none;

}

/\*The validation summary span when validation is valid.\*/

.validation-summary-errors {

    color: #b94a48;

    font-weight: bold;

}

/\*The validation summary span when validation is invalid.\*/

/\*field validation ------------------- \*/

.field-validation-valid {

    display: none;

}

/\*The validation span below input textbox when validation is valid.\*/

.field-validation-error {

    color: #b94a48;

}

/\*The validation span below input textbox when validation is invalid.\*/

.input-validation-error {

    border: 1px solid #b94a48;

}

/\*The input textbox when validation is invalid.\*/

2.11.3. In the View - validation Summary

In our case, please check the following file

Views/Gamer/Create.cshtml

Views/Gamer/Edit.cshtml

<div>

    @Html.ValidationSummary(false, WebShareConst.ValidationSummaryTitleString)

    @\*@Html.ValidationSummary(false, "Please check the following fields.")\*@

</div>

@\*

1.

//@Html.ValidationSummary(false, "Please check the following fields.")

//@Html.ValidationSummary(false, @WebShareConst.ValidationSummaryTitleString)

1st parameter is excludePropertyErrors

true to have the summary display model-level errors only,

or false to have the summary display all errors.

\*@

2.11.4. In the View - Field validation

You may specify the Error Message of "ValidationMessageFor".

@Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "", new { @class = "text-danger" })

When Error Message is empty string, it will display default error message.

@Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "\*", new { @class = "text-danger" })

This will Error Message string is not empty string,

it will display what you type in that Error Message string.

<div class="form-group">

    @Html.LabelFor(model => model.ConfirmEmailAddress, new { @class = "control-label col-md-2" })

    <div class="col-md-10">

        @Html.EditorFor(model => model.ConfirmEmailAddress, new { htmlAttributes = new { @class = "form-control" } })

        @Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "", new { @class = "text-danger" })

        @\*@Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "\*", new { @class = "text-danger" })\*@

    </div>

</div>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.12. Fix the Date format dd/mm/yyy issue - Jquery Validation in Chrome and Safari won't validate with dd/mm/yyyy

Fixed

<https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

Reference:

<https://www.nuget.org/packages/jquery-globalize/1.3.0>

<https://github.com/globalizejs/globalize>

<http://devdens.blogspot.com.au/2011/11/jquery-validation-fix-for-date-format_29.html>

<https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

<https://stackoverflow.com/questions/14020117/dd-mm-yyyy-date-validation-error-with-google-chrome>

2.12.1. Induction

2.12.1.1. Web.config

<appSettings>

  <add key="webpages:Version" value="3.0.0.0" />

  <add key="webpages:Enabled" value="false" />

  <add key="ClientValidationEnabled" value="true" />

  <add key="UnobtrusiveJavaScriptEnabled" value="true" />

  <add key="ApplicationName" value="Online Game Web" />

</appSettings>

<system.web>

  <caching>

    <outputCacheSettings>

      <outputCacheProfiles>

        <clear />

        <add name="outputCacheProfile1" duration="60" varyByParam="none" />

      </outputCacheProfiles>

    </outputCacheSettings>

  </caching>

  <customErrors mode="On">

    <error statusCode="401" redirect="Error/UnauthorizedError" />

    <error statusCode="404" redirect="Error/NotFound" />

    <error statusCode="500" redirect="Error/InternalServerError" />

  </customErrors>

  <globalization culture="en-au" uiCulture="en-AU" />

  <compilation debug="true" targetFramework="4.6.1" />

  <httpRuntime targetFramework="4.6.1" />

</system.web>

2.12.1.2. WebShare/WebShareConst.cs

public const string DateFormat = "dd/MM/yyyy";

public const string DateStringFormat = "{0:dd/MM/yyyy}";

2.12.1.3. Models/Gamer/Gamer.cs

[MetadataType(typeof(GamerMetaData))]

public partial class Gamer

2.12.1.4. Models/Gamer/GamerMetaData.cs

//[Range(typeof(DateTime), "1/1/1970", "1/1/2001", ErrorMessage = "Date is out of Range")]    //Error - Client Side validation will never pass

//[Range(typeof(DateTime), "1/1/1970", "1/1/2001")] //Error - Client Side validation will never pass

//[DateRange("01/01/1970", "01/01/2001")]

//[DateRange("01/01/1970", DateTime.Now.ToShortDateString())] //Error :  An attribute argument must be a constant expression

//[BetweenMinimumDateAndToday("01/01/1970")]

[BeforeToday]

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

[DisplayFormat(DataFormatString = WebShareConst.DateStringFormat, ApplyFormatInEditMode = true)]

[Required]

public DateTime DateOfBirth { get; set; }

/\*

3.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

//public DateTime DateOfBirth { get; set; }

We discuss this at T009 and T010 already.

3.1.

DisplayFormat attribute

3.1.1.

//[DisplayFormat(DataFormatString = "{0:d}")]

Display only the date part. E.g. 29/04/1986

3.1.2.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy HH:mm:ss}")]

Display in 24 hour notation. E.g. 29/04/1986 13:00:00

3.1.3.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy hh:mm:ss tt}")]

Display in 12 hour notation. E.g. 29/04/1986 1:00:00 PM

3.2.

DisplayFormatAttribute attribute

//[DisplayFormatAttribute(DataFormatString="{0:d}")]

Display only the date part. E.g. 29/04/1986

\*/

2.12.1.5. Views/Gamer/Edit.cshtml   Or  Views/Gamer/Create.cshtml

<div class="form-group">

    @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })

    <div class="col-md-10">

        @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-control datepicker" } })

        @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })

    </div>

</div>

2.12.2. Fix the Date format dd/mm/yyy issue

If you run the view in IE or FireFox,

then it will work as we expect.

However, if you run the view in Google Chrome or Apple Safari,

**Jquery Validation in Chrome and Safari won't validate with dd/mm/yyyy**.

You are not alone, if you google "**validation dd/mm/yyyy**" you will find a lot of solutions.

The following is the link which fixes my issue.

Fixed

<https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

The following is the link which also helps me

Reference:

<https://www.nuget.org/packages/jquery-globalize/1.3.0>

<https://github.com/globalizejs/globalize>

<http://devdens.blogspot.com.au/2011/11/jquery-validation-fix-for-date-format_29.html>

<https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

<https://stackoverflow.com/questions/14020117/dd-mm-yyyy-date-validation-error-with-google-chrome>

2.12.2.1. Scripts/operation.js

//DateFormate : dd/mm/yyyy

//Reference:

//<https://www.w3schools.com/jsref/jsref_obj_regexp.asp>

//<https://stackoverflow.com/questions/6906725/unobtrusive-validation-in-chrome-wont-validate-with-dd-mm-yyyy>

jQuery.validator.methods.date = function (value, element) {

    var dateRegex = /^(0?[1-9]\/|[12]\d\/|3[01]\/){2}(19|20)\d\d$/;

    return this.optional(element) || dateRegex.test(value);

};

$(function () {

});

2.12.2.2. Views/Shared/\_Layout.cshtml

<link href="~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />

<link href="~/Content/Site.css" rel="stylesheet" type="text/css" />

<script src="~/Scripts/modernizr-2.6.2.js"></script>

<script src="~/Scripts/jquery-1.10.2.min.js"></script>

<script src="~/Scripts/bootstrap.min.js"></script>

<script src="~/Scripts/jquery-ui-1.12.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

<script src="~/Scripts/operation.js"></script>

2.12.3. DatePicker

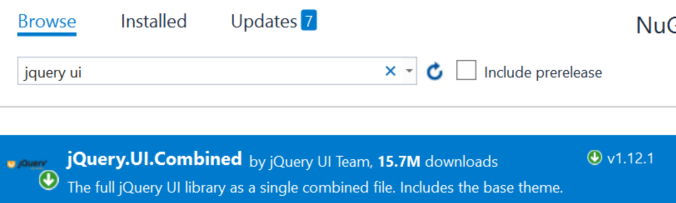
Reference:

ASP.NET MVC 3: Integrating with the jQuery UI date picker and adding a jQuery validate date range validator

<https://blogs.msdn.microsoft.com/stuartleeks/2011/01/25/asp-net-mvc-3-integrating-with-the-jquery-ui-date-picker-and-adding-a-jquery-validate-date-range-validator/>

2.12.3.1. Install Jquery UI

Install Jquery UI from NuGet Package



2.12.3.2. Views/Shared/\_Layout.cshtml

@using OnlineGame.Web.WebShare

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>@ViewBag.Title - My [ASP.NET](http://asp.net/) Application</title>

    <link href="~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />

    <link href="~/Content/themes/base/jquery-ui.min.css" rel="stylesheet" />

    <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />

    @\*<script src="~/Scripts/jquery-1.10.2.min.js"></script>\*@

    <script src="~/Scripts/jquery-1.12.4.min.js"></script>

    <script src="~/Scripts/modernizr-2.6.2.js"></script>

    <script src="~/Scripts/bootstrap.min.js"></script>

    <script src="~/Scripts/jquery-ui-1.12.1.min.js"></script>

    <script src="~/Scripts/jquery.validate.min.js"></script>

    <script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

    <script src="~/Scripts/operation.js"></script>

</head>

<body>

    <div class="navbar navbar-inverse navbar-fixed-top">

        <div class="container">

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                @Html.ActionLink(System.Configuration.ConfigurationManager.AppSettings["ApplicationName"], "Index", "Gamer", new { area = "" }, new { @class = "navbar-brand" })

            </div>

            <div class="navbar-collapse collapse">

                <ul class="nav navbar-nav"></ul>

            </div>

        </div>

    </div>

    <div class="container body-content">

        @RenderBody()

        <hr />

        <footer>

            <p>&copy; @DateTime.Now.Year - My [ASP.NET](http://asp.net/) Application</p>

        </footer>

    </div>

    <script type="text/javascript">

        $(function () {

            $('.datepicker').datepicker(

                {

                    //dateFormat: "dd/mm/yy"

                    dateFormat: "@WebShareConst.DateFormatJavascriptString"

                });

        });

    </script>

</body>

</html>

2.12.3.3. WebShare/WebShareConst.cs

public const string DateFormat = "dd/MM/yyyy";

public const string DateFormatJavascriptString = "dd/mm/yy";

public const string DateStringFormat = "{0:dd/MM/yyyy}";

2.12.3.4. Views/Gamer/Edit.cshtml  Or  Views/Gamer/Create.cshtml

<div class="form-group">

    @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })

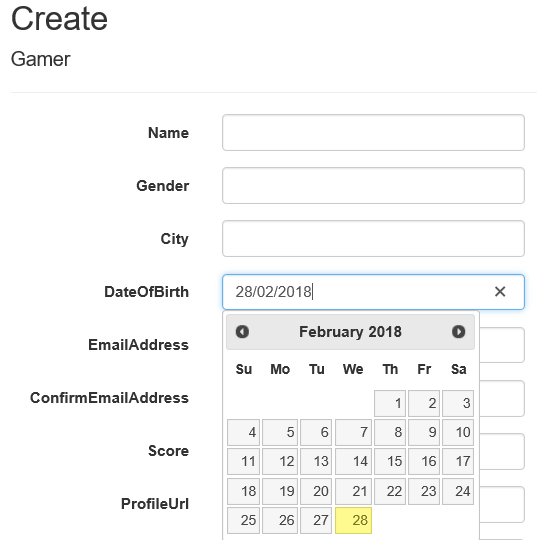
    <div class="col-md-10">

        @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-control datepicker" } })

        @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })

    </div>

</div>



2.12.3.5. Client Side Date Range

Please follow the instruction here.

<https://blogs.msdn.microsoft.com/stuartleeks/2011/01/25/asp-net-mvc-3-integrating-with-the-jquery-ui-date-picker-and-adding-a-jquery-validate-date-range-validator/>

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](http://ado.net/) Entity Data Model

Name:

**OnlineGameDataModel**

-->

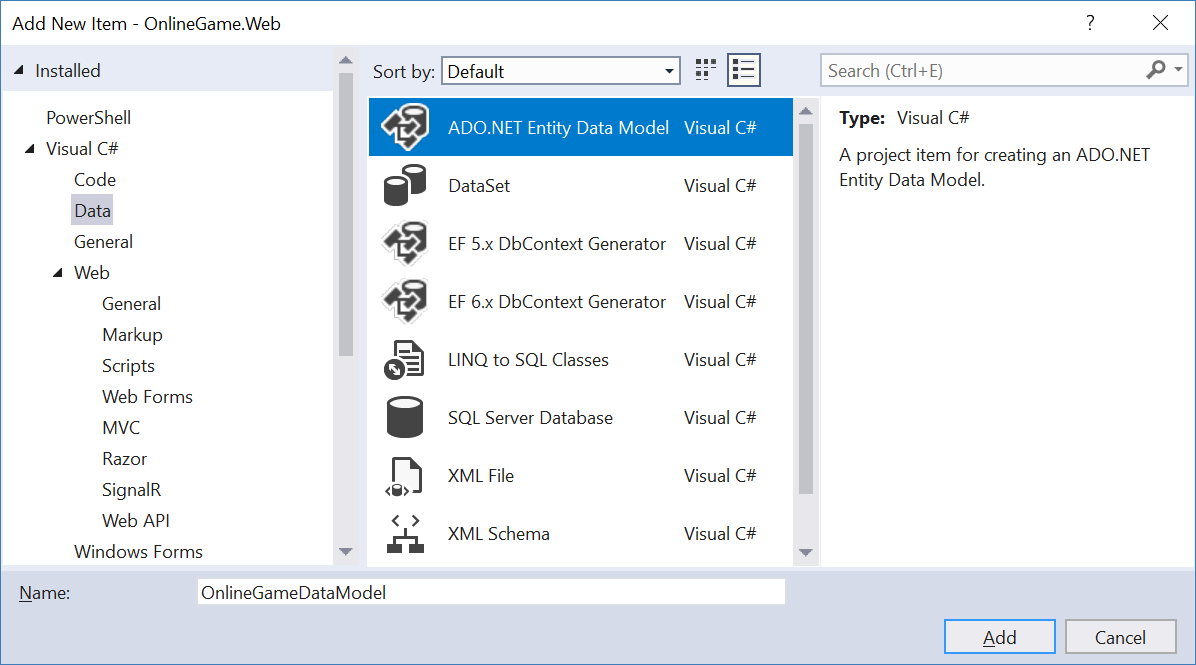
EF Designer from database

....

-->

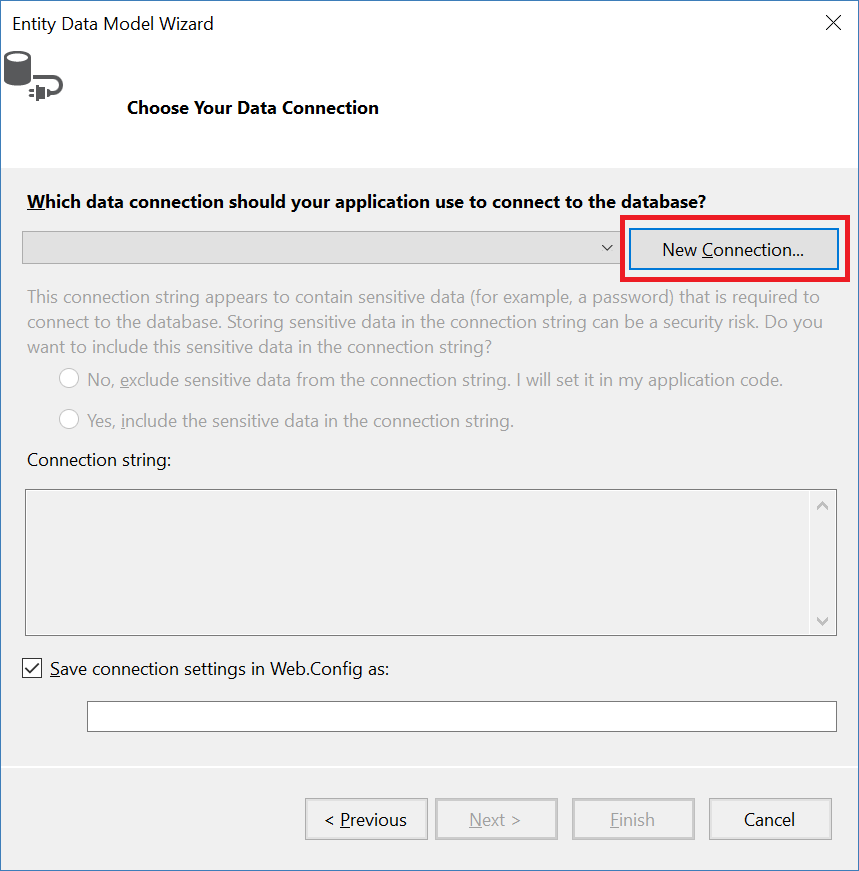
Save Connection settings in Web.Config as:

**OnlineGameContext**



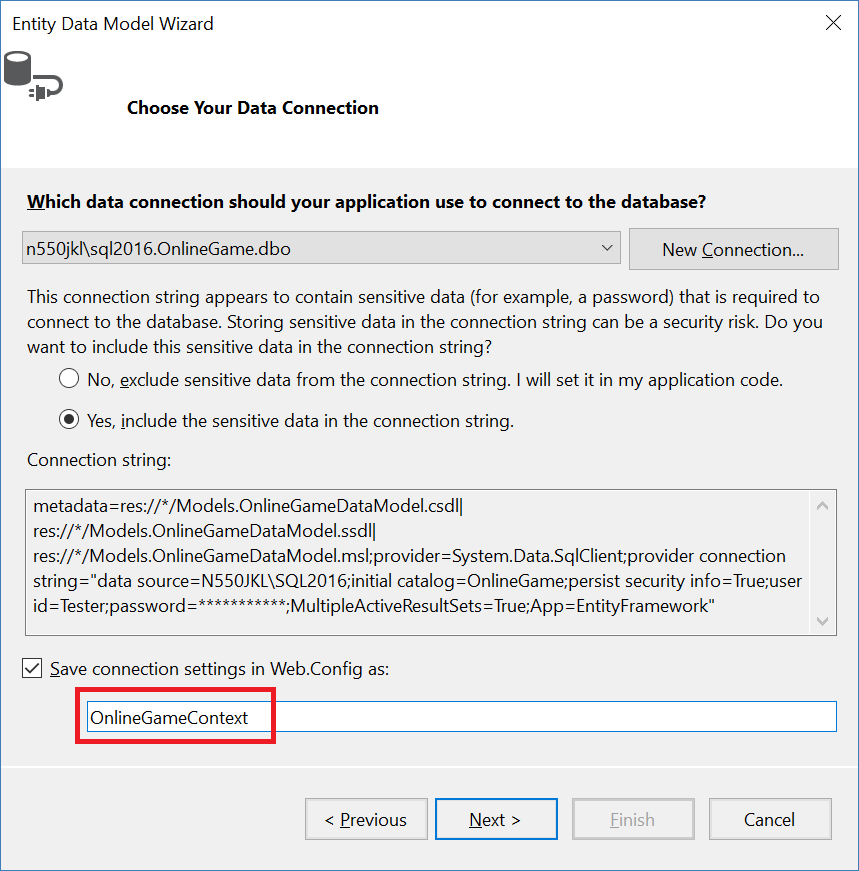
Graphical user interface, text, application, email

Description automatically generated



Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

3.2. Controllers/GamerController.cs

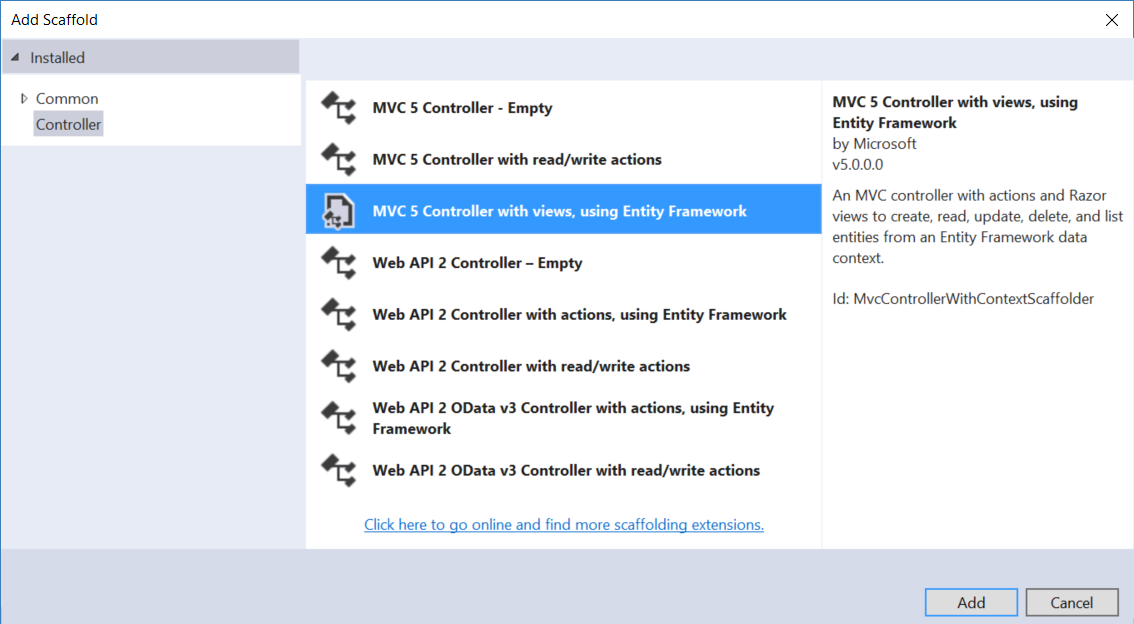
Controllers --> Right click --> Add --> Controller

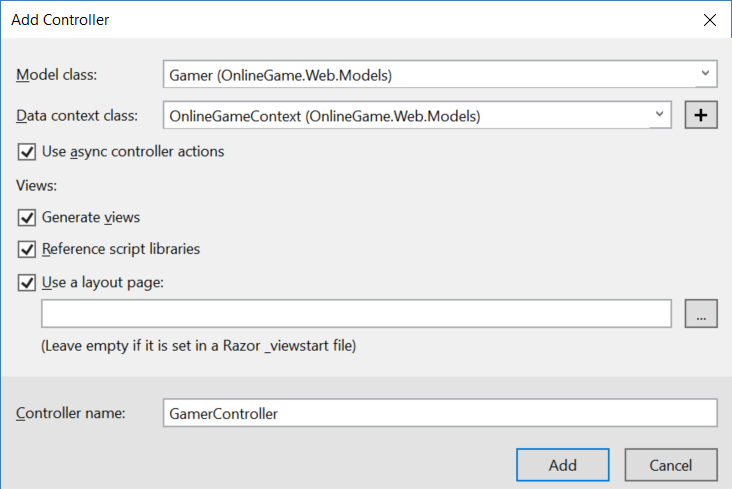
-->

**MVC 5 Controller with views, using Entity Framework**

Graphical user interface, application

Description automatically generated





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

Graphical user interface, text, application

Description automatically generated with medium confidence

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

4. OnlineGame.Web

4.1. Models/Gamer/Gamer.cs

using System.ComponentModel.DataAnnotations;

namespace OnlineGame.Web.Models

{

    [MetadataType(typeof(GamerMetaData))]

    public partial class Gamer

    {

        [Compare("EmailAddress")]

        [Required]

        public string ConfirmEmailAddress { get; set; }

        //ConfirmEmailAddress property will not save into database.

    }

}

/\*

1.

//[Compare("EmailAddress")]

//public string ConfirmEmailAddress { get; set; }

Reference:

<https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation>

Validates two properties in a model match.

\*/

4.2. Models/Gamer/GamerMetaData.cs

using System;

using System.ComponentModel.DataAnnotations;

using System.Web.Mvc;

using OnlineGame.Web.WebShare;

using OnlineGame.Web.WebShare.Attribute;

namespace OnlineGame.Web.Models

{

    public class GamerMetaData

    {

        //private static readonly string DateFormat = System.Configuration.ConfigurationManager.AppSettings["DateStringFormat"];

        ////It will not work.

        ////DisplayFormat attribute DataFormatString property only take "DateStringFormat" constant string.

        [StringLength(20, MinimumLength = 2)]

        //[RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$")] //First Name Last Name

        //[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression)] //First Name Last Name

        //[RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = "Please enter first name or first name and last name.")] //First Name Last Name

        [RegularExpression(WebShareConst.FirstNameLastNameRegularExpression, ErrorMessage = WebShareConst.FirstNameLastNameNotValid)] //First Name Last Name

        [Required]

        public string Name { get; set; }

        //[Remote("IsEmailAvailable", "Gamer", ErrorMessage = "The email has already been taken.")]

        //[Remote("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

        //[RemoteClientServer("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

        ////Remote attribute can only do the client side validation.

        ////RemoteClientServer is a customize attribute which can do both client side and server side validation.

        ////Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

        ////If you really want to use it, please use two different model classes for Edit and Create mode.

        //[RegularExpression(@"^[\w-\.\_\+%]+@(?:[\w-]+\.)+[\w]{2,6}$", ErrorMessage = "Email is not valid")]  //EmailAddress Regular Expression

        //[RegularExpression(WebShareConst.EmailRegularExpression, ErrorMessage = "Email is not valid")]  //EmailAddress Regular Expression

        [RegularExpression(WebShareConst.EmailRegularExpression, ErrorMessage = WebShareConst.EmailNotValid)]  //EmailAddress Regular Expression

        [Required]

        public string EmailAddress { get; set; }

        [Range(1, 1000000)]

        [Required]

        public int GameMoney { get; set; }

        //[Range(typeof(DateTime), "1/1/1970", "1/1/2001", ErrorMessage = "Date is out of Range")]    //Error - Client Side validation will never pass

        //[Range(typeof(DateTime), "1/1/1970", "1/1/2001")] //Error - Client Side validation will never pass

        //[DateRange("01/01/1970", "01/01/2001")]

        //[DateRange("01/01/1970", DateTime.Now.ToShortDateString())] //Error :  An attribute argument must be a constant expression

        //[BetweenMinimumDateAndToday("01/01/1970")]

        [BeforeToday]

        //[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

        [DisplayFormat(DataFormatString = WebShareConst.DateStringFormat, ApplyFormatInEditMode = true)]

        [Required]

        public DateTime DateOfBirth { get; set; }

    }

}

/\*

1.

//[StringLength(20, MinimumLength = 2)]

attribute check if the input string has certain length.

If you leave the input textbox as blank, then it does not care.

Thus we need [Required] attribute to ensure the field always has value.

---------------------------------------------

2.

//[Range(1, 1000000)]

//public int GameMoney { get; set; }

The field must be between 1 to 1000000.

---------------------------------------------

3.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

//public DateTime DateOfBirth { get; set; }

We discuss this at T009 and T010 already.

3.1.

DisplayFormat attribute

3.1.1.

//[DisplayFormat(DataFormatString = "{0:d}")]

Display only the date part. E.g. 29/04/1986

3.1.2.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy HH:mm:ss}")]

Display in 24 hour notation. E.g. 29/04/1986 13:00:00

3.1.3.

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy hh:mm:ss tt}")]

Display in 12 hour notation. E.g. 29/04/1986 1:00:00 PM

3.2.

DisplayFormatAttribute attribute

//[DisplayFormatAttribute(DataFormatString="{0:d}")]

Display only the date part. E.g. 29/04/1986

---------------------------------------------

4.

////[Range(typeof(DateTime), "1/1/1970", "1/1/2001", ErrorMessage = "Date is out of Range")]    //Error - Client Side validation will never pass

////[Range(typeof(DateTime), "1/1/1970", "1/1/2001")] //Error - Client Side validation will never pass

////[DateRange("01/01/1970", "01/01/2001")]

////[DateRange("01/01/1970", DateTime.Now.ToShortDateString())] //Error :  An attribute argument must be a constant expression

////[BetweenMinimumDateAndToday("01/01/1970")]

//[BeforeToday]

//[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

//public DateTime DateOfBirth { get; set; }

Validating a datetime range in EF annotation

We can’t pass a DateTime in as attribute arguments must be const.

We have to build our own Customize Validation Attribute

Reference:

<https://blogs.msdn.microsoft.com/stuartleeks/2011/01/25/asp-net-mvc-3-integrating-with-the-jquery-ui-date-picker-and-adding-a-jquery-validate-date-range-validator/>

[https://forums.asp.net/t/1831436.aspx?Validating+a+datetime+range+in+EF+annotation](https://forums.asp.net/t/1831436.aspx?Validating%2Ba%2Bdatetime%2Brange%2Bin%2BEF%2Bannotation)

<https://stackoverflow.com/questions/13183647/date-range-validation-with-entity-framework-4-data-annotations>

4.1.

[DateRange], [BetweenMinimumDateAndToday], [BeforeToday] are Customize Validation Attributes

Please have a look.

---------------------------------------------

5.

RegularExpression

//[RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$")] //First Name Last Name

//public string Name { get; set; }

//[RegularExpression(@"^[\w-\.\_\+%]+@(?:[\w-]+\.)+[\w]{2,6}$", ErrorMessage = "Please enter a valid email address")]  //EmailAddress Regular Expression

//public string EmailAddress { get; set; }

Reference:

<https://regexr.com/>

<https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>

---------------------------------------------

6.

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = "The email has already been taken.")]

//[Remote("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

//[RemoteClientServer("IsEmailAvailable", "Gamer", ErrorMessage = WebShareConst.EmailHasBeenTaken)]

//// Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

//// You may use two different model class for create and edit mode.

//public string EmailAddress { get; set; }

6.1.

Remote attribute can only do the client side validation.

RemoteClientServer is a customize attribute which can do both client side and server side validation.

Don't add Remote attribute in shared model, it will affect both Edit and Create mode.

If you really want to use it, please use two different model classes for Edit and Create mode.

6.2.

RemoteAttribute uses AJAX to make an asynchronous call to the server-side method.

If the user disables javascript function of the browser, then it will not work.

Thus, we should also have server-side validation.

6.3.

server-side validation.

In the Create HttpPost Method.

It is hard to read that validation logic is in the controller.

Using validation attributes is always the preferred method.

You may use two different model class for create and edit mode.

//[HttpPost]

//[ValidateAntiForgeryToken]

//public async Task<ActionResult> Create([Bind(Include = "Id,Name,Gender,City,DateOfBirth,EmailAddress,ConfirmEmailAddress,Score,ProfileUrl,GameMoney,RolePhoto,RolePhotoAltText,TeamId")] Gamer gamer)

//{

//    //If the Email already exists, then add Model validation error

//    if (db.Gamers.Any(g => g.EmailAddress == gamer.EmailAddress))

//    {

//        //AddModelError(Key, ErrorMessage)

//        ModelState.AddModelError("EmailAddress", WebShareConst.EmailHasBeenTaken);

//    }

//    //It is hard to read that validation logic is in controller.

//    //Using validation attributes is always preferred method.

//    if (!ModelState.IsValid)

//    {

//        ViewBag.TeamId = new SelectList(db.Teams, "Id", "Name", gamer.TeamId);

//        return View(gamer);

//    }

//    db.Gamers.Add(gamer);

//    await db.SaveChangesAsync();

//    return RedirectToAction("Index");

//}

6.4.

Another way to has both server side and client side validation,

you have to create your own RemoteClientServerAttribute which extend RemoteAttribute.

\*/

4.3. Controllers/GamerController.cs

using System.Data.Entity;

using System.Linq;

using System.Threading.Tasks;

using[System.Net](http://system.net/);

using System.Web.Mvc;

using System.Web.Script.Serialization;

using OnlineGame.Web.Models;

using OnlineGame.Web.WebShare;

namespace OnlineGame.Web.Controllers

{

    public class GamerController : Controller

    {

        private OnlineGameContext \_db = new OnlineGameContext();

        public JsonResult IsEmailAvailable(string emailAddress)

        {

            JsonResult jsonResult = Json(!\_db.Gamers.Any(g => g.EmailAddress == emailAddress),

                JsonRequestBehavior.AllowGet);

            //JSONResult to String

            //Reference: <https://stackoverflow.com/questions/4571985/jsonresult-to-string>

            string jsonResultString = new JavaScriptSerializer().Serialize(jsonResult.Data);

            return jsonResult;

        }

        // GET: Gamer

        [HttpGet]

        public async Task<ActionResult> Index()

        {

            IQueryable<Gamer> gamers = \_db.Gamers.Include(g => g.Team);

            return View(await gamers.ToListAsync());

        }

        // GET: Gamer/Details/5

        [HttpGet]

        public async Task<ActionResult> Details(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Gamer gamer = await \_db.Gamers.FindAsync(id);

            if (gamer == null)

            {

                return HttpNotFound();

            }

            return View(gamer);

        }

        // GET: Gamer/Create

        [HttpGet]

        public ActionResult Create()

        {

            ViewBag.TeamId = new SelectList(\_db.Teams, "Id", "Name");

            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]

        public async Task<ActionResult> Create([Bind(Include = "Id,Name,Gender,City,DateOfBirth,EmailAddress,ConfirmEmailAddress,Score,ProfileUrl,GameMoney,RolePhoto,RolePhotoAltText,TeamId")] Gamer gamer)

        {

            //Validation Logic

            //If the Email already exists, then add Model validation error

            if (\_db.Gamers.Any(g => g.EmailAddress == gamer.EmailAddress))

            {

                //AddModelError(Key, ErrorMessage)

                ModelState.AddModelError("EmailAddress", WebShareConst.EmailHasBeenTaken);

            }

            //It is hard to read that validation logic is in controller.

            //Using validation attributes is always preferred method.

            //You may use two different model class for create and edit mode.

            if (!ModelState.IsValid)

            {

                ViewBag.TeamId = new SelectList(\_db.Teams, "Id", "Name", gamer.TeamId);

                return View(gamer);

            }

            \_db.Gamers.Add(gamer);

            await \_db.SaveChangesAsync();

            return RedirectToAction("Index");

        }

        // GET: Gamer/Edit/5

        [HttpGet]

        public async Task<ActionResult> Edit(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Gamer gamer = await \_db.Gamers.FindAsync(id);

            if (gamer == null)

            {

                return HttpNotFound();

            }

            gamer.ConfirmEmailAddress = gamer.EmailAddress;

            ViewBag.TeamId = new SelectList(\_db.Teams, "Id", "Name", gamer.TeamId);

            return View(gamer);

        }

        // 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]

        public async Task<ActionResult> Edit([Bind(Include = "Id,Name,Gender,City,DateOfBirth,EmailAddress, ConfirmEmailAddress,Score,ProfileUrl,GameMoney,RolePhoto,RolePhotoAltText,TeamId")] Gamer gamer)

        {

            if (!ModelState.IsValid)

            {

                ViewBag.TeamId = new SelectList(\_db.Teams, "Id", "Name", gamer.TeamId);

                return View(gamer);

            }

            \_db.Entry(gamer).State = EntityState.Modified;

            await \_db.SaveChangesAsync();

            return RedirectToAction("Index");

        }

        // GET: Gamer/Delete/5

        [HttpGet]

        public async Task<ActionResult> Delete(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Gamer gamer = await \_db.Gamers.FindAsync(id);

            if (gamer == null)

            {

                return HttpNotFound();

            }

            return View(gamer);

        }

        // POST: Gamer/Delete/5

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public async Task<ActionResult> DeleteConfirmed(int id)

        {

            Gamer gamer = await \_db.Gamers.FindAsync(id);

            \_db.Gamers.Remove(gamer);

            await \_db.SaveChangesAsync();

            return RedirectToAction("Index");

        }

        protected override void Dispose(bool disposing)

        {

            if (disposing)

            {

                \_db.Dispose();

            }

            base.Dispose(disposing);

        }

    }

}

4.4. Views/Gamer/Edit.cshtml

@using OnlineGame.Web.WebShare

@model OnlineGame.Web.Models.Gamer

@{

    ViewBag.Title = "Edit";

    ////Enable client side validation

    //Html.EnableClientValidation(true);

    //Html.EnableUnobtrusiveJavaScript(true);

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <h4>Gamer</h4>

        <hr />

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        @Html.HiddenFor(model => model.Id)

        <div class="form-group">

            @Html.LabelFor(model => model.Name, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Gender, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Gender, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Gender, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.City, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-control datepicker" } })

                @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.EmailAddress, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.EmailAddress, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.EmailAddress, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.ConfirmEmailAddress, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.ConfirmEmailAddress, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "", new { @class = "text-danger" })

                @\*@Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "\*", new { @class = "text-danger" })\*@

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Score, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Score, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Score, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.ProfileUrl, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.ProfileUrl, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.ProfileUrl, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.GameMoney, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.GameMoney, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.GameMoney, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.RolePhoto, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.RolePhoto, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.RolePhoto, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.RolePhotoAltText, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.RolePhotoAltText, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.RolePhotoAltText, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.TeamId, "TeamId", new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.DropDownList("TeamId", null, new { @class = "form-control" })

                @Html.ValidationMessageFor(model => model.TeamId, "", new { @class = "text-danger" })

            </div>

        </div>

        <div>

            @Html.ValidationSummary(false, WebShareConst.ValidationSummaryTitleString)

        </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>\*@

@\*<script type="text/javascript" language="javascript">

    jQuery.validator.methods.date = function (value, element) {

        var dateRegex = /^(0?[1-9]\/|[12]\d\/|3[01]\/){2}(19|20)\d\d$/;

        return this.optional(element) || dateRegex.test(value);

    };

</script>\*@

@\*

1.

//@Html.ValidationSummary(false, "Please check the following fields.")

//@Html.ValidationSummary(false, WebShareConst.ValidationSummaryTitleString)

1st parameter is excludePropertyErrors

true to have the summary display model-level errors only,

or false to have the summary display all errors.

\*@

4.5. Views/Gamer/Create.cshtml

@using OnlineGame.Web.WebShare

@model OnlineGame.Web.Models.Gamer

@{

    ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <h4>Gamer</h4>

        <hr />

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">

            @Html.LabelFor(model => model.Name, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Gender, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Gender, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Gender, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.City, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.DateOfBirth, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.DateOfBirth, new { htmlAttributes = new { @class = "form-control datepicker" } })

                @Html.ValidationMessageFor(model => model.DateOfBirth, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.EmailAddress, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.EmailAddress, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.EmailAddress, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.ConfirmEmailAddress, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.ConfirmEmailAddress, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "", new { @class = "text-danger" })

                @\*@Html.ValidationMessageFor(model => model.ConfirmEmailAddress, "\*", new { @class = "text-danger" })\*@

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Score, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Score, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Score, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.ProfileUrl, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.ProfileUrl, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.ProfileUrl, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.GameMoney, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.GameMoney, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.GameMoney, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.RolePhoto, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.RolePhoto, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.RolePhoto, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.RolePhotoAltText, new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.RolePhotoAltText, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.RolePhotoAltText, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.TeamId, "TeamId", new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.DropDownList("TeamId", null, new { @class = "form-control" })

                @Html.ValidationMessageFor(model => model.TeamId, "", new { @class = "text-danger" })

            </div>

        </div>

        <div>

            @Html.ValidationSummary(false, WebShareConst.ValidationSummaryTitleString)

        </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>\*@