

(T19)討論 Area(分區)。討論 ActionResult 的 SubClass。實作不同的 ActionResult  
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924

---

(T19)討論 Area(分區)。討論 ActionResult 的 SubClass。實作不同的 ActionResult

---

## 0. Summary

- 
- 1. New Project - OnlineGame
  - 1.1. New Project - OnlineGame.Web
    - 1.1.1. App\_Start/FilterConfig.cs
    - 1.1.2. App\_Start/RouteConfig.cs
    - 1.1.3. Global.asax.cs
    - 1.1.4. Web.config
    - 1.1.5. Add Customized Error View and Error Controller
      - 1.1.5.1. Controllers/ErrorController.cs
      - 1.1.5.2. Views/Shared/Error.cshtml
      - 1.1.5.3. Views/Shared/UnauthorizedError.cshtml
      - 1.1.5.4. Views/Shared/NotFound.cshtml
    - 1.1.6. WebShared/CustomizeCacheAttribute.cs
      - 1.1.6.1. WebShared/CustomizeCacheAttribute.cs
      - 1.1.6.2. The way to use WebShared/CustomizeCacheAttribute.cs

- - 2. OnlineGame.Web
    - 2.1. Introduction of Area
    - 2.2. Add Gamer Area
      - 2.2.1. Areas/Gamer/Controllers/HomeController.cs
      - 2.2.2. Areas/Gamer/Views/Home/Index.cshtml
    - 2.3. Add VipGamer Area
      - 2.3.1. Areas/VipGamer/Controllers/HomeController.cs
      - 2.3.2. Areas/VipGamer/Views/Home/Index.cshtml
    - 2.4. Add Admin Area
      - 2.4.1. Areas/Admin/Controllers/HomeController.cs
      - 2.4.2. Areas/Admin/Views/Home/Index.cshtml
    - 2.5. No Area
      - 2.5.1. Controllers/HomeController.cs
      - 2.5.2. Views/Home/Index.cshtml
    - 2.6. Run the Web Application
    - 2.7. Fix the Issue
      - 2.7.1. App\_Start/RouteConfig.cs
    - 2.8. Navigate to Gamer Area Index View
    - 2.9. Fix Issues
      - 2.9.1. Areas/Admin/AdminAreaRegistration.cs
      - 2.9.2. Areas/Gamer/GamerAreaRegistration.cs
      - 2.9.3. Areas/VipGamer/VipGamerAreaRegistration.cs
      - 2.9.4. Navigate to Area Index View
    - 2.10. Views/Home/Index.cshtml - ActionLink for Area
- 

## 0. Summary

In this tutorial, we will discuss

## \* Area

-----  
使用 Area 將網站畫地分區。

\* 龐大複雜的 Web Application 通常需要 Area 來將網站畫地分區。

\* 每個 Area 都有屬於自己的 Model，View，Controller 結構。

\* 每個 Area 內的 MVC 都不會和其他 Area 的 MVC 互相影響，可以獨立作業。

1.

### ActionResult

ActionResult is an abstract class.

Reference:

[https://msdn.microsoft.com/en-us/library/system.web.mvc.actionresult\(v=vs.118\).aspx](https://msdn.microsoft.com/en-us/library/system.web.mvc.actionresult(v=vs.118).aspx)

<https://github.com/icsharpcode/ILSpy>

E.g.

```
//public ActionResult Index()  
//{  
//    return View(); // returns ViewResult object  
//    //return Json("{Id:15,Name:AAA}"); // returns JsonResult object  
//}
```

The following are sub class of ActionResult and  
the helper methods used to return the specific sub-type

1.1.

#### HttpNotFoundResult

Helper Methods : Controller.HttpNotFound Method

Returns an object to indicate that the requested resource can not be found.

1.2.

#### HttpUnauthorizedResult

Helper Methods : N/A

It represents the result of an unauthorized HTTP request.

1.3.

#### JavaScriptResult

Helper Methods : N/A, JavaScript

Returns a piece of JavaScript code that can be executed on the client

1.4.

#### FileContentResult

Helper Methods : N/A, File

Returns a file to the client

1.5.

#### FilePathResult

Helper Methods : N/A, File

Returns a file to the client, which is provided by the given path.

1.6.

#### FileStreamResult

Helper Methods : N/A, File

Returns a file to the client, which is provided by a Stream.

1.7.

#### PartialViewResult

Helper Methods : PartialView

Returns a Specified partial view.

## 1.8.

ViewResult

### Helper Methods : View

Returns a Specified view.

## 1.9.

ContentResult

### Helper Methods : Content

Write content to the response stream without requiring a view.

## 1.10.

JsonResult

### Helper Methods : Json

Returns a JsonResult which serializes an object in JSON format.

## 1.11.

EmptyResult

### Helper Methods : N/A

An empty response is returned. Used when the action method returns void.

## 1.12.

RedirectResult

### Helper Methods : Redirect

Performs an HTTP redirection to a specific new URL.

## 1.13.

RedirectToRouteResult

### Helper

### Methods : RedirectToAction, RedirectToRoute, RedirectToActionPermanent, RedirectToRoutePermanent

Performs an HTTP redirection to another action method that is determined by the routing engine, based on given route data

---

## 2.

Area structure is usually used in big and complex web application structure.

Each area has its own set of Models, Views, Controllers, Routes.

E.g.

We are going to create the following areas.

A.

Gamer Area

The area allows gamers to create their profile.

B.

VipGamer Area

The area allows VIP gamers to create their profile and get some game rewards.

C.

Admin Area

The area allows the administrator to maintain the game.

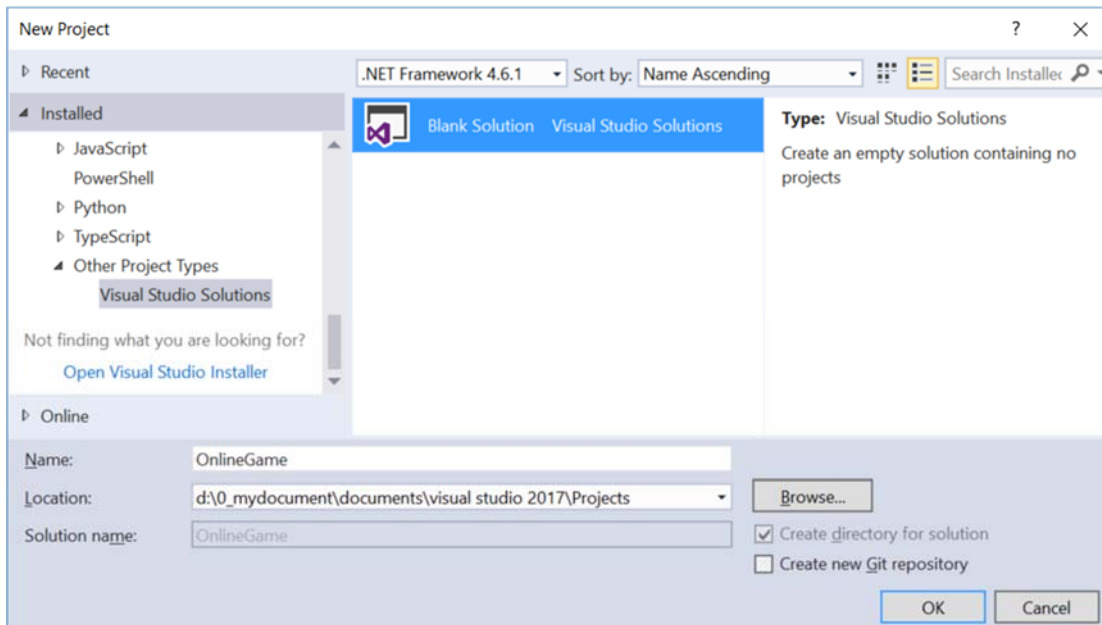
=====

# 1. New Project - OnlineGame

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

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

Name: **OnlineGame**



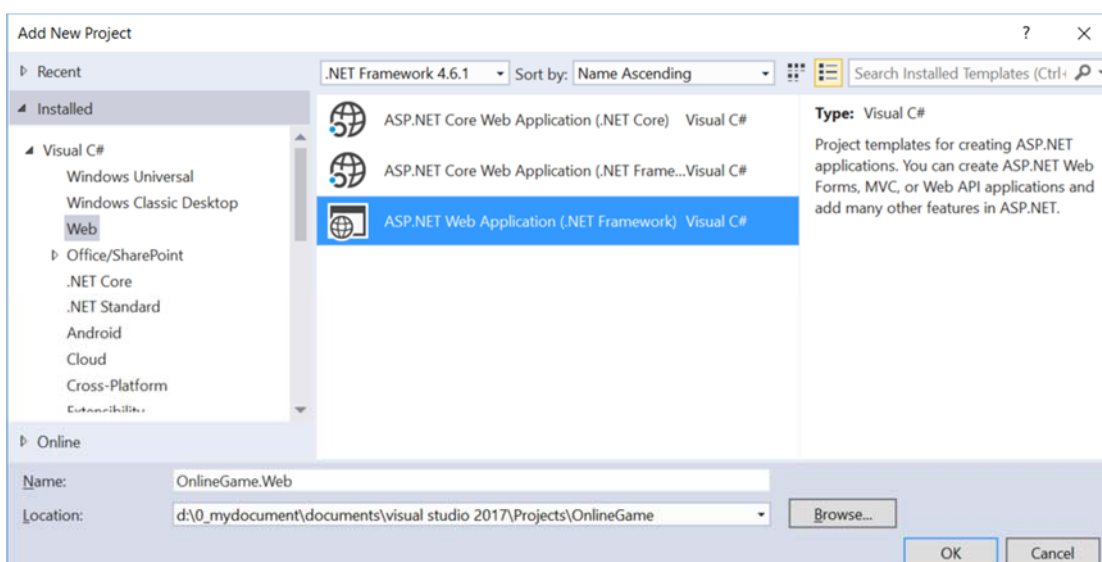
## 1.1. New Project - OnlineGame.Web

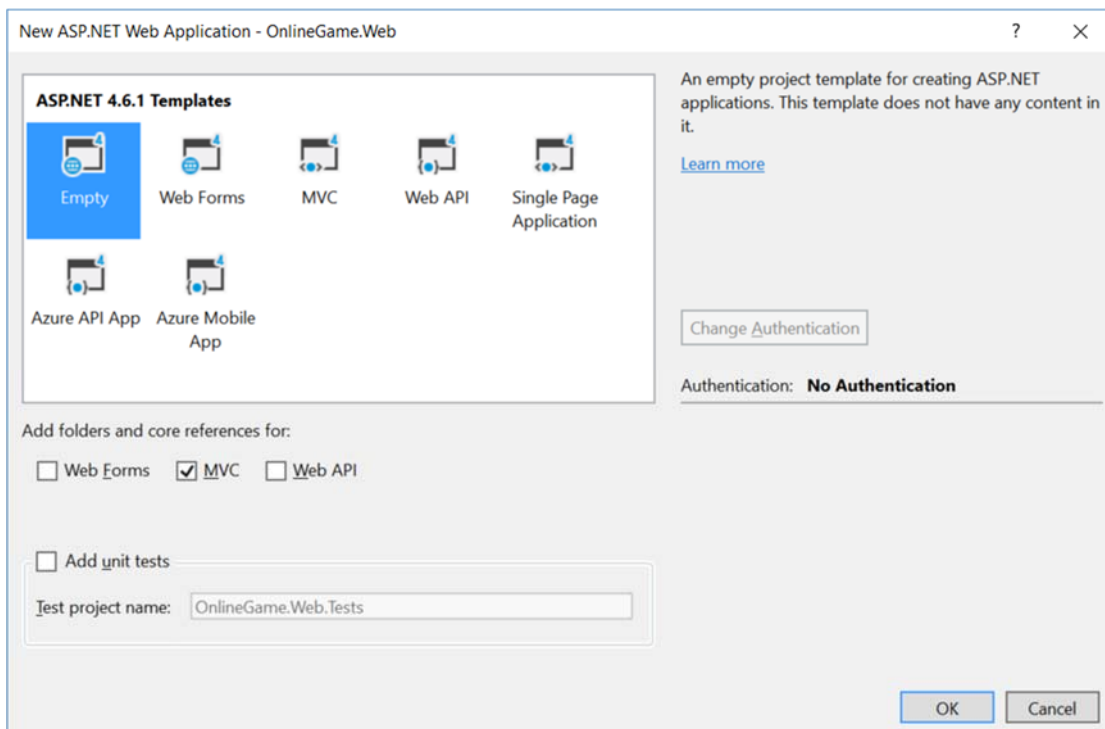
Solutions Name --> Add --> New Project -->

Visual C# --> Web --> ASP.NET Web Application (.Net Framework)  
-->

Name: **OnlineGame.Web**

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





### 1.1.1. App\_Start/FilterConfig.cs

```
using System.Web.Mvc;

namespace WebApplication1
{
    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
*/
```

## 1.1.2. App\_Start/RouteConfig.cs

```
using System.Web.Mvc;
using System.Web.Routing;
namespace OnlineGame.Web
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            //Handle the Route of the axd request file.
            //E.g. ASP.Net Tracing
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
            //Handle the Route called "Default".
            //The mapping URL is "{controller}/{action}/{id}"
            //Set the default value of Controller, action, and id.
            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", 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-ignoreroresource-axd-pathinfo>

This line can handle the axd file request route,

E.g. trace.axd

.axd files don't exist physically.

ASP.NET uses URLs with .axd extensions

(ScriptResource.axd and WebResource.axd) internally,

and they are handled by an HttpHandler.

Therefore, you should keep this rule,

to prevent ASP.NET MVC from trying to handle the request

instead of letting the dedicated HttpHandler do it.

2.2.

trace.axd

Reference:

<https://msdn.microsoft.com/en-us/library/wwh16c6c.aspx>

trace.axd trace details for a specific request.

If you want to enable trace.axd,

then you have to go to Web.config

Add <trace enabled="true" pageOutput="false"/> under <system.web>

Then run the project, type the following URL

<http://localhost/OnlineGame.Web/trace.axd>

This will return ASP.NET trace, trace.axd.

If you do not have

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

then you can not enable the trace.axd.

```
*/
```

## 1.1.3. Global.asax.cs

```
using System.Web.Mvc;
```

```
using System.Web.Routing;
```

```
using WebApplication1;
```

```
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);
```

```
        }
```

```
    }
```

```
}
```

## 1.1.4. Web.config



```
11 <appSettings>
12   <add key="webpages:Version" value="3.0.0.0" />
13   <add key="webpages:Enabled" value="false" />
14   <add key="ClientValidationEnabled" value="true" />
15   <add key="UnobtrusiveJavaScriptEnabled" value="true" />
16 </appSettings>
17 <system.web>
18   <outputCache>
19     <outputCacheSettings>
20       <outputCacheProfiles>
21         <clear/>
22         <add name="outputCacheProfile1" duration="60" varyByParam="none" />
23       </outputCacheProfiles>
24     </outputCacheSettings>
25   </outputCache>
26   <customErrors mode="On">
27     <error statusCode="401" redirect="Error/UnauthorizedError" />
28     <error statusCode="404" redirect="Error/NotFound" />
29     <error statusCode="500" redirect="Error/InternalServerError" />
30   </customErrors>
31   <globalization culture="en-au" />
32   <compilation debug="true" targetFramework="4.6.1" />
33   <httpRuntime targetFramework="4.6.1" />
34 </system.web>
```

```
<system.web>
  <outputCache>
    <outputCacheSettings>
      <outputCacheProfiles>
        <clear/>
        <add name="outputCacheProfile1" duration="60" varyByParam="none" />
      </outputCacheProfiles>
    </outputCacheSettings>
  </outputCache>
  <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>
```

## 1.1.5. Add Customized Error View and Error Controller

### 1.1.5.1. 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>
*/

```

### 1.1.5.2. Views/Shared/Error.cshtml

```
@{
    ViewBag.Title = "Error";
}
<h2>Something occurs, please contact support.</h2>
```

### 1.1.5.3. Views/Shared/UnauthorizedError.cshtml

```
@{
    ViewBag.Title = "UnauthorizedError";
}
<h2>Error UnauthorizedError statusCode=401</h2>
You are trying to access something which you are not allowed to access.
```

<http://localhost/onlinegame.web/Error/UnauthorizedError>  
**Error UnauthorizedError statusCode=401**  
 You are trying to access something which you are not allowed to access.

### 1.1.5.4. Views/Shared/NotFound.cshtml

```
@{
    ViewBag.Title = "NotFound";
}
<h2>Error NotFound statusCode=404</h2>
The request can not be found.
```

<http://localhost/onlinegame.web/Error/NotFound>  
**Error NotFound statusCode=404**  
 The request can not be found.

### 1.1.5.5. Views/Shared/InternalServerError.cshtml

```
@{
    ViewBag.Title = "InternalServerError";
}
<h2>Error InternalServerError statusCode=500</h2>
The developers did something wrong, not users fault.
```

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

## Error InternalServerError statusCode=500

The developers did something wrong, not users fault.

### 1.1.6. WebShared/CustomizeCacheAttribute.cs

#### 1.1.6.1. WebShared/CustomizeCacheAttribute.cs

```
using System.Web.Mvc;
using System.Web.Configuration;
namespace OnlineGame.Web.WebShared
{
    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>
*/
```

### 1.1.6.2. The way to use WebShared/CustomizeCacheAttribute.cs

```
[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 throw exception
[CustomizeCache("outputCacheProfile1")]
public string GetGamerCount2()
{
    return $"Gamer Count = {db.Gamers.Count()} At {DateTime.Now}";
}
```

## 2. OnlineGame.Web

### 2.1. Introduction of Area

Area structure is usually used in big and complex web application structure.

Each area has its own set of Models, Views, Controllers, Routes.

E.g.

We are going to create the following areas.

A.

Gamer Area

The area allows gamers to create their profile.

B.

VipGamer Area

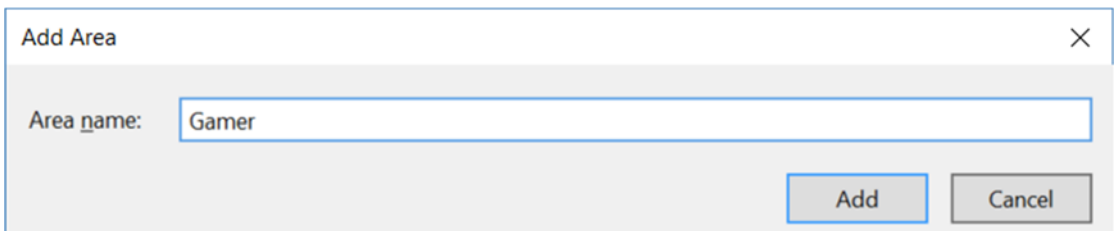
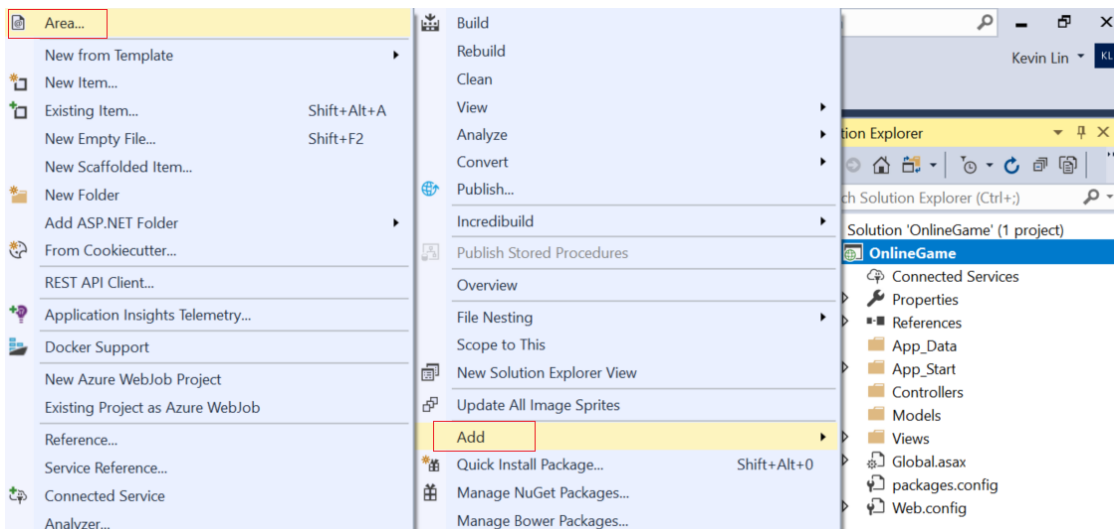
The area allows VIP gamers to create their profile and get some game rewards.

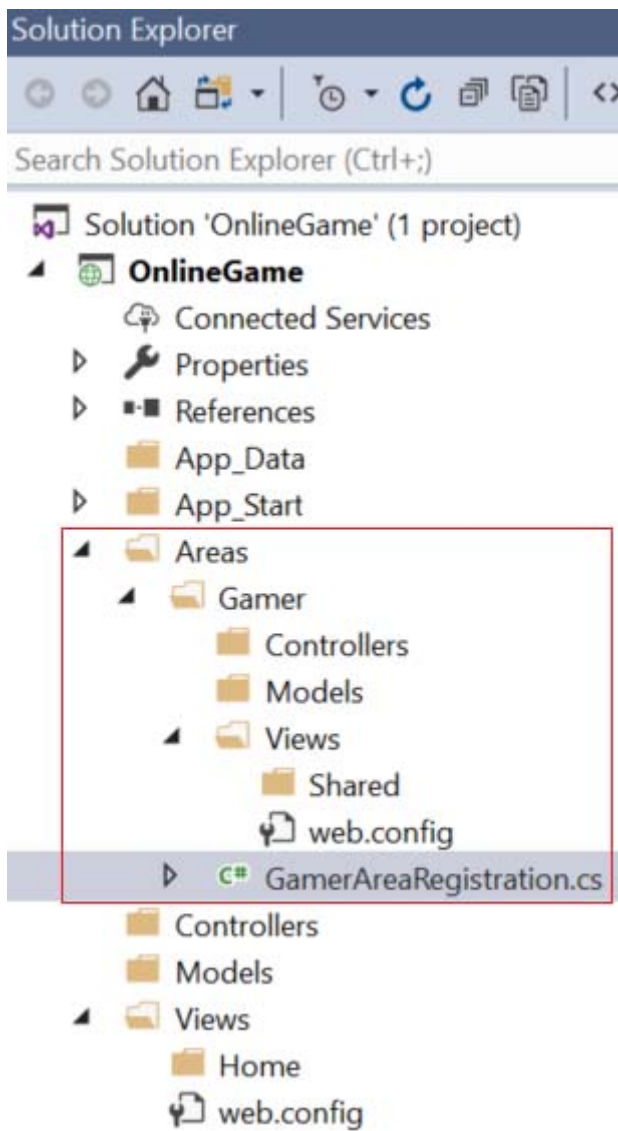
C.

Admin Area

The area allows the administrator to maintain the game.

## 2.2. Add Gamer Area





### 2.2.1. Areas/Gamer/Controllers/HomeController.cs

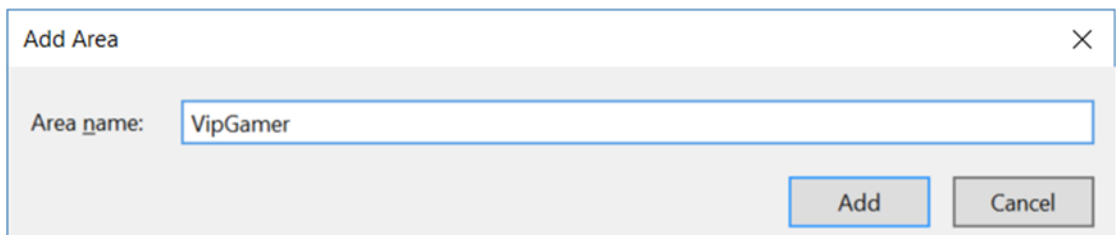
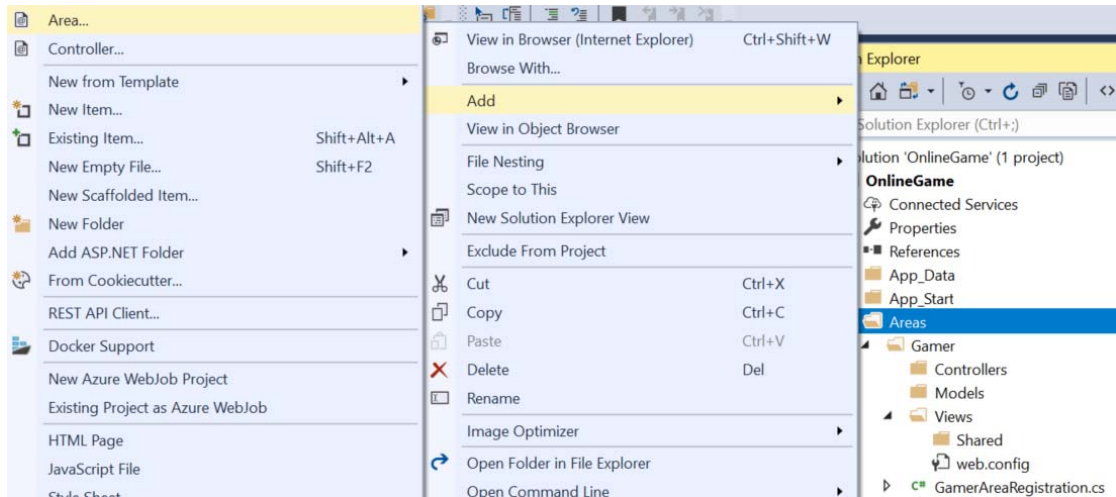
```
using System.Web.Mvc;
namespace OnlineGame.Web.Areas.Gamer.Controllers
{
    public class HomeController : Controller
    {
        // GET: Gamer/Home
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

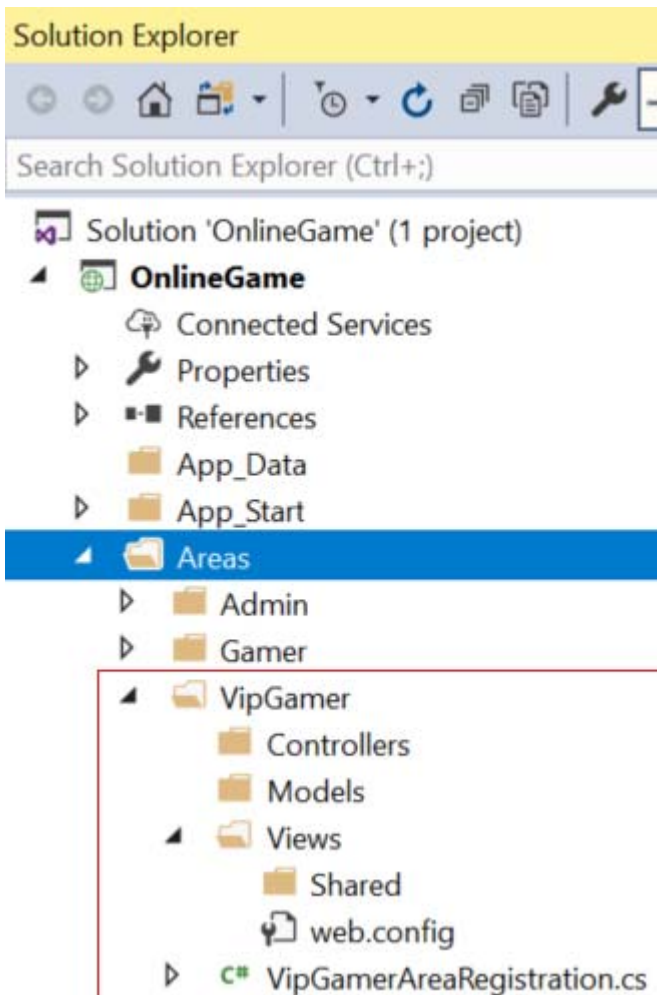
### 2.2.2. Areas/Gamer/Views/Home/Index.cshtml

```
@{
    ViewBag.Title = "Index";
}
```

}  
<h2>Areas/Gamer/Views/Home/Index.cshtml</h2>

## 2.3. Add VipGamer Area





### 2.3.1. Areas/VipGamer/Controllers/HomeController.cs

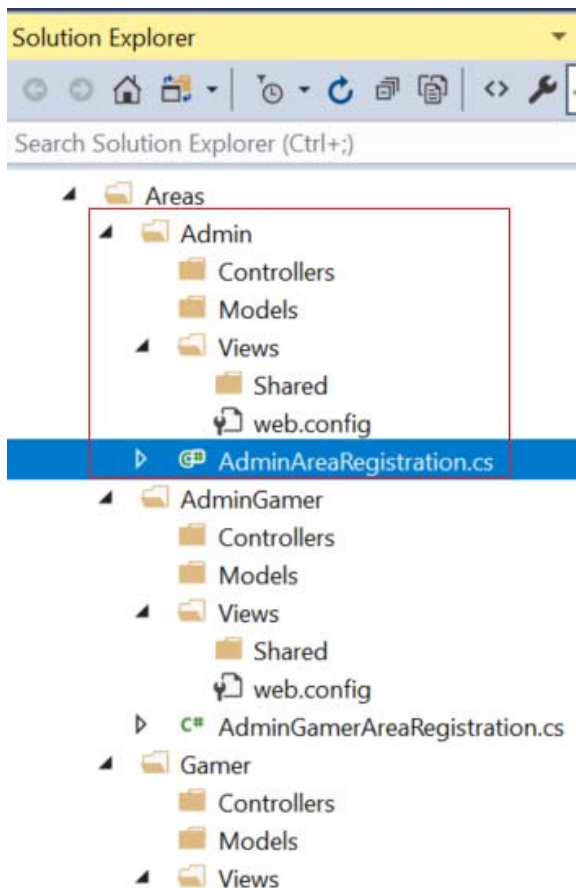
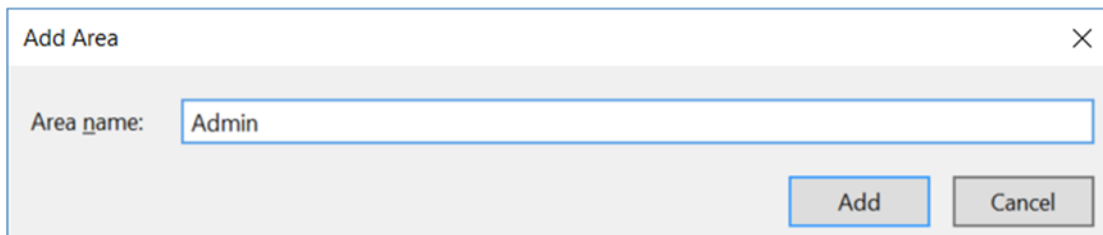
```
using System.Web.Mvc;
namespace OnlineGame.Web.Areas.VipGamer.Controllers
{
    public class HomeController : Controller
    {
        // GET: VipGamer/Home
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

### 2.3.2. Areas/VipGamer/Views/Home/Index.cshtml

```
@{
    ViewBag.Title = "Index";
}
<h2>Areas/VipGamer/Views/Home/Index.cshtml</h2>
```

## 2.4. Add Admin Area





### 2.4.1. Areas/Admin/Controllers/HomeController.cs

```
using System.Web.Mvc;
namespace OnlineGame.Web.Areas.Admin.Controllers
{
    public class HomeController : Controller
    {
        // GET: Admin/Home
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

### 2.4.2. Areas/Admin/Views/Home/Index.cshtml

```
@{
    ViewBag.Title = "Index";
}
<h2>Areas/Admin/Views/Home/Index.cshtml</h2>
```

## 2.5. No Area

### 2.5.1. Controllers/HomeController.cs

```
using System.Web.Mvc;
namespace OnlineGame.Web.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

### 2.5.2. Views/Home/Index.cshtml

```
@{
    ViewBag.Title = "Index";
}
<h2>Views/Home/Index.cshtml</h2>
```

## 2.6. Run the Web Application

<http://localhost:63486/>



## Server Error in '/' Application.

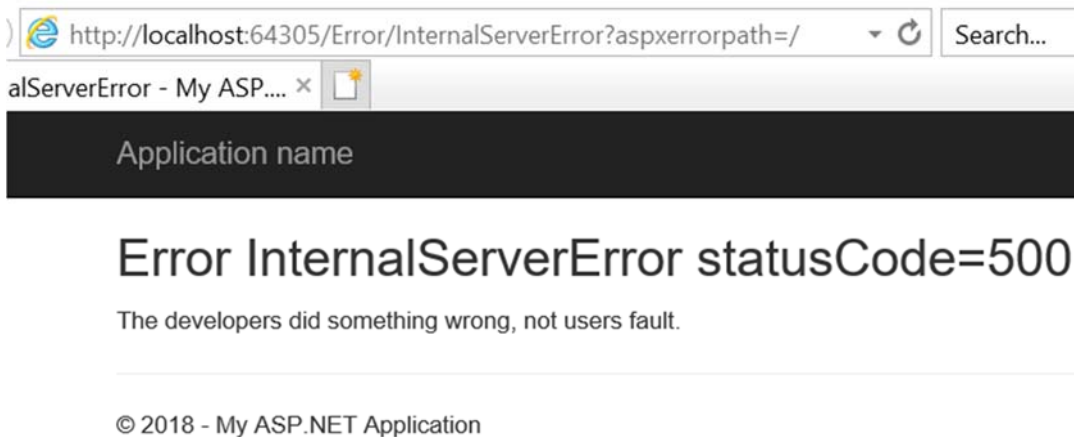
*Multiple types were found that match the controller named 'Home'. This can happen if the route that services this request ('{controller}/{action}/{id}') does not specify namespaces to search for a controller that matches the request. If this is the case, register this route by calling an overload of the 'MapRoute' method that takes a 'namespaces' parameter.*

*The request for 'Home' has found the following matching controllers:*

*OnlineGame.Controllers.HomeController  
OnlineGame.Areas.VipGamer.Controllers.HomeController  
OnlineGame.Areas.Gamer.Controllers.HomeController  
OnlineGame.Areas.Admin.Controllers.HomeController*

Or

It will show the following if you implement ErrorController



## 2.7. Fix the Issue

### 2.7.1. App\_Start/RouteConfig.cs

```
using System.Web.Mvc;
using System.Web.Routing;
namespace OnlineGame.Web
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            //Handle the Route of the axd request file.
            //E.g. ASP.Net Tracing
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        }
    }
}
```

```

//Handle the Route called "Default".
//The mapping URL is "{controller}/{action}/{id}"
//Set the default value of Controller, action, and id.
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },
    namespaces: new[] { "OnlineGame.Web.Controllers" }
);
}
}
}

```

```

/*
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}");

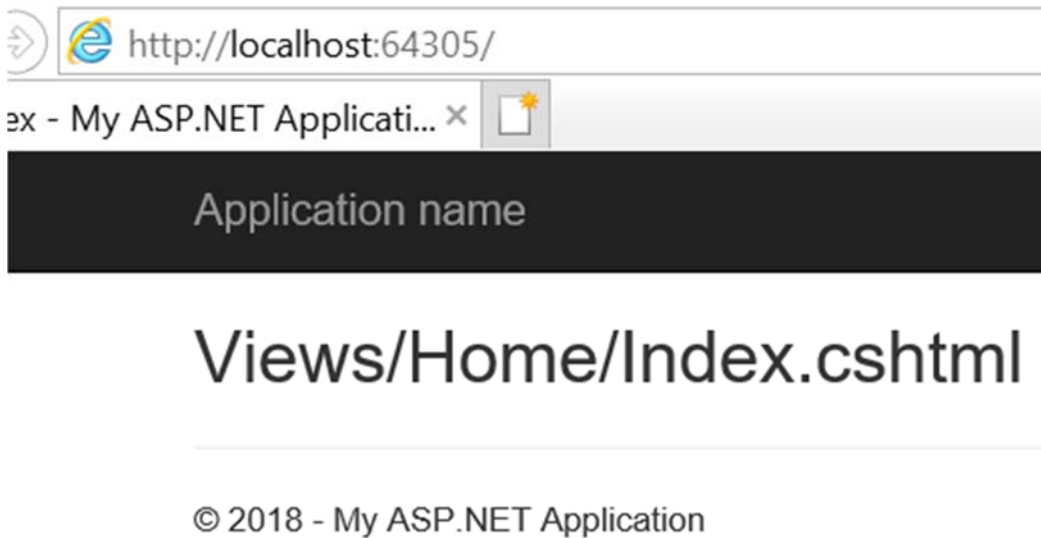
2.1.  
Reference:  
<https://stackoverflow.com/questions/9016650/what-is-routes-ignoreroresource-axd-pathinfo>

This line can handle the axd file request route,  
E.g. trace.axd  
.axd files don't exist physically.  
[ASP.NET](#) uses URLs with .axd extensions  
(ScriptResource.axd and WebResource.axd) internally,  
and they are handled by an HttpHandler.  
Therefore, you should keep this rule,  
to prevent [ASP.NET](#) MVC from trying to handle the request  
instead of letting the dedicated HttpHandler do it.

2.2.  
trace.axd  
Reference:  
<https://msdn.microsoft.com/en-us/library/wwh16c6c.aspx>  
trace.axd trace details for a specific request.  
If you want to enable trace.axd,  
then you have to go to Web.config  
Add <trace enabled="true" pageOutput="false"/> under <system.web>  
Then run the project, type the following URL  
<http://localhost/OnlineGame.Web/trace.axd>

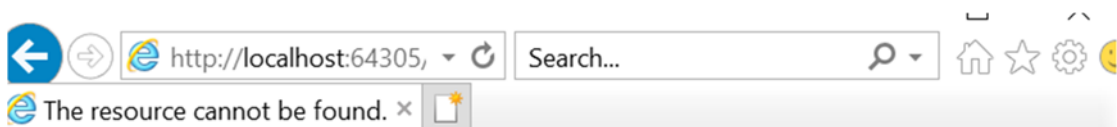
```
This will return ASP.NET trace, trace.axd.  
If you do not have  
// routes.IgnoreRoute("{resource}.axd/{*pathInfo}");  
then you can not enable the trace.axd.  
*/
```

Run Again, it will show the view.



## 2.8. Navigate to Gamer Area Index View

<http://localhost:64305/gamer>



### Server Error in '/' Application.

*The resource cannot be found.*

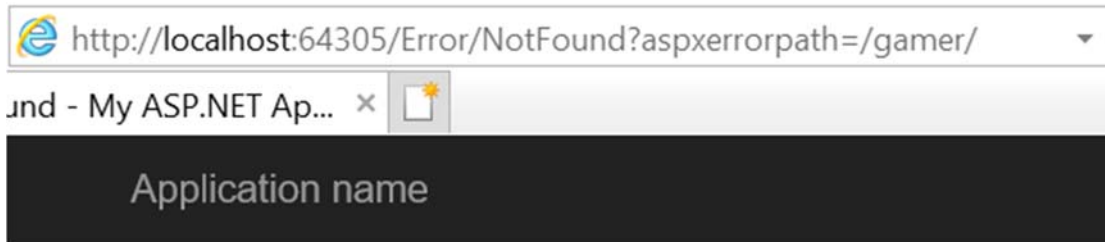
**Description:** HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

**Requested URL:** /gamer

**Version Information:** Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.7.2633.0

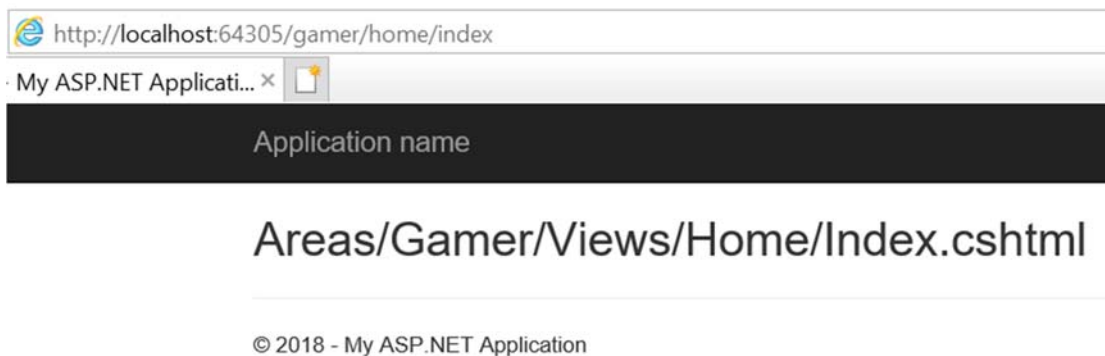
Or

It will show the following if you implement ErrorController



The correct ways to call "Gamer Area Home Controller Index Action".

<http://localhost:64305/gamer/home/index>



## 2.9. Fix Issues

### 2.9.1. Areas/Admin/AdminAreaRegistration.cs

```
using System.Web.Mvc;
namespace OnlineGame.Web.Areas.Admin
{
    public class AdminAreaRegistration : AreaRegistration
    {
        public override string AreaName
        {
            get
            {
                return "Admin";
            }
        }
        public override void RegisterArea(AreaRegistrationContext context)
        {
            context.MapRoute(
                "Admin_default",
```

```

        "Admin/{controller}/{action}/{id}",
        new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
}
}

```

## 2.9.2. Areas/Gamer/GamerAreaRegistration.cs

```

using System.Web.Mvc;
namespace OnlineGame.Web.Areas.Gamer
{
    public class GamerAreaRegistration : AreaRegistration
    {
        public override string AreaName
        {
            get
            {
                return "Gamer";
            }
        }
        public override void RegisterArea(AreaRegistrationContext context)
        {
            context.MapRoute(
                "Gamer_default",
                "Gamer/{controller}/{action}/{id}",
                new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}

```

## 2.9.3. Areas/VipGamer/VipGamerAreaRegistration.cs

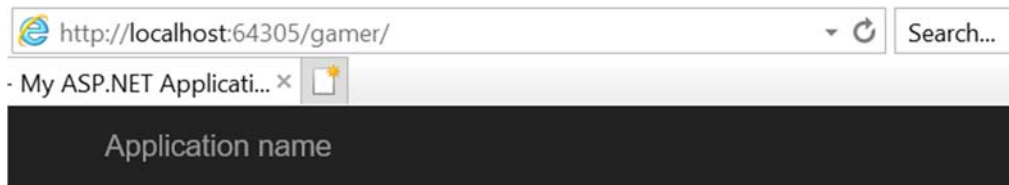
```

using System.Web.Mvc;
namespace OnlineGame.Web.Areas.VipGamer
{
    public class VipGamerAreaRegistration : AreaRegistration
    {
        public override string AreaName
        {
            get
            {
                return "VipGamer";
            }
        }
        public override void RegisterArea(AreaRegistrationContext context)
        {
            context.MapRoute(
                "VipGamer_default",
                "VipGamer/{controller}/{action}/{id}",
                new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}

```

## 2.9.4. Navigate to Area Index View

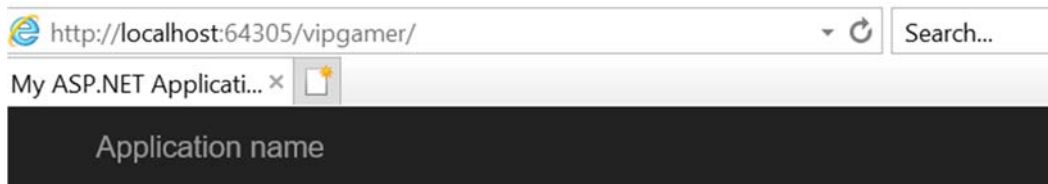
<http://localhost:64305/gamer/>



Areas/Gamer/Views/Home/Index.cshtml

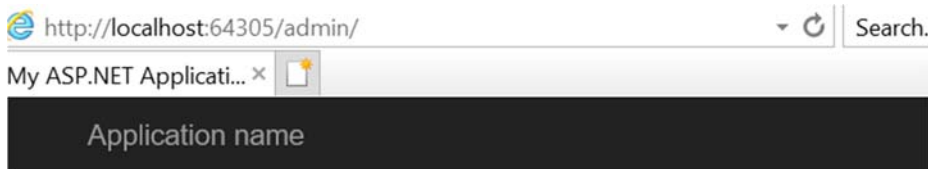
© 2018 - My ASP.NET Application

<http://localhost:64305/vipgamer/>



Areas/VipGamer/Views/Home/Index.cshtml

<http://localhost:64305/admin/>



Areas/Admin/Views/Home/Index.cshtml

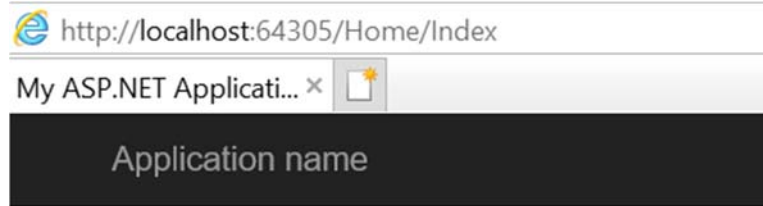
## 2.10. Views/Home/Index.cshtml - ActionLink for Area

```
@{
    ViewBag.Title = "Index";
}
<h2>Views/Home/Index.cshtml</h2>
<div>
    @Html.ActionLink("Views/Home/Index.cshtml", "Index", "Home", new { area = "" }, null) <br />
    @Html.ActionLink("Areas/Gamer/Views/Home/Index.cshtml", "Index", "Home", new { area
= "Gamer" }, null) <br />
    @Html.ActionLink("Areas/VipGamer/Views/Home/Index.cshtml", "Index", "Home", new { area
= "VipGamer" }, null) <br />
```



```
@Html.ActionLink("Areas/Admin/Views/Home/Index.cshtml", "Index", "Home", new { area  
= "Admin" }, null) <br />  
</div>
```

<http://localhost:64305/Home/Index>



## Views/Home/Index.cshtml

Views/Home/Index.cshtml  
Areas/Gamer/Views/Home/Index.cshtml  
Areas/VipGamer/Views/Home/Index.cshtml  
Areas/Admin/Views/Home/Index.cshtml

© 2018 - My ASP.NET Application