(T17)在IIS討論RequireHttps、SSL、HTTPS、Certicate  
CourseGUID: 8503b39c-5887-4634-8291-facfb3117924  
=======================================================================  
(T17)在IIS討論RequireHttps、SSL、HTTPS、Certicate  
=======================================================================  
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

1.1.5.5. Views/Shared/InternalServerError.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 - IIS

2.1. Run In IIS

2.2. Controllers/HomeController.cs

2.3. Views/Home/Index.cshtml

2.4. Views/Home/Index3.cshtml

2.5. Views/Home/Index5.cshtml

-----------

3. OnlineGame.Web - Certificate

3.1. Use IIS to Create Self-Signed Certificates

3.2. Add the Self-Signed Certificates to HTTPS site binding - Assign Web Applications with the specific certificate

3.3. Dis-allow a Web Application use HTTP Protocol

3.3.1. Remove HTTP Binding

3.3.2. Set the application always require SSL

3.4. Import / Export Certificates

3.4.1. Export Certificates

3.4.2. Import Certificates

3.5. Use MakeCert.exe to Create Self-Signed Certificates

3.6. Error Pages - Status Code: 403.4

3.7. Error Controller - Status Code: 403.4

3.7.1. Web.config

3.7.2. Controllers/ErrorController.cs

3.7.3. Views/Shared/RedirectToHttps.cshtml  
=======================================================================

0. Summary

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

In this tutorial, we will discuss

\* Please ensure you fully understand T013 Grid View before you continue.

\* Action filters

Reference:

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/controllers-and-routing/understanding-action-filters-cs>

An action filter is an attribute that you can apply to a controller action -- or an entire controller -- that modifies the way in which the action is executed.

    \* Authorize

    \* ChildActionOnly

    \* HandleError

    \* OutputCache

    \* **RequireHttps**

    \* ValidateInput

    \* ValidateAntiForgeryToken

\* HTTP

\* HTTPS

\* SSL, HTTPS with SSL

\* Certificates

\* The certificate authority

\* Create certificates by MakeCert.exe or IIS

\* Error Pages

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

第14章: 動作過濾器Action Filter 3 - RequireHttps搭配IIS，SSL，HTTPS，Certicate完全攻略。

\* 我遇過一些工作幾年的工程竟然不知道怎麼使用SSL和HTTPS來實現Secure Web Application。

\* 於是該章節從無到有寫出了一個範例，手把手帶你設定Certificate，完美介紹IIS和Secure Web Application之間的觀念。

\* 下次遇到公司開發的軟體產品突然不能跑，千萬別慌，可能你只是忘記設定Certificate。

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

Reference:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968(v=vs.85).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968%28v=vs.85%29.aspx)

<https://www.iis.net/downloads/microsoft/url-rewrite>

Reference:

保哥的Certificate的觀念補充

<https://blog.miniasp.com/post/2018/04/21/PKI-Digital-Certificate-Format-Convertion-Notes.aspx>

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

1.

Advantages of using HTTPS

HTTP(Hyper Text Transfer Protocol ) uses port 80.

HTTPS(Hyper Text Transfer Protocol Secure) uses port 443 and it is usually used with Secure Socket Layer(SSL).

The messages that are exchanged over the internet by HTTPS protocol are encrypted.

The server certificates installed on the IIS are responsible for encryption and decryption of  messages exchanged between the client and the server.

Compare HTTP, HTTPS usually need more time for Key negotiation (SSL handshake) which allows the server to authenticate itself to the client.

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

2.

Secure Socket Layer(SSL) is a security technology which uses server certificates for encryption and decryption in order to build an encrypted link between a server and a browser.

SSL certificate contains a public key and certificate issuer,

so client side browser can verify the certificate issuer as an official Certificate Authority

and then communicate with key and requested page's data by certificate.

With Secure Socket Layer(SSL), when a user requests a secure Web page,

the server creates a key and encrypts the page’s data and then send to the user depending on the user’s session.

On the client side, browser gets the key to decrypt the requested encrypts page’s data and then use the key to send a new request.

E.g.

When the client side browser trusts GoDaddy Certificate Authority and already trust any GoDaddy SSL certificate.

then the client side browser will also trust other GoDaddy SSL certificates.

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

3.

3.1.

The following are some trusted certificate authority who can issue server certificates.

A. GoDaddy

B. verisign

C. Thawte

D. Geotrust

E. Comodo

3.2.

**The certificate authority**  assures the client side browsers by verifying the server’s identity over the Internet.

When a client side browser requests over https,

it also requests the server certificate and sees if the site is in trusted sites list provided by the certificate authority.

If the server certificate was not already signed by the client side browser

or if the site is in trusted sites list provided by the certificate authority

or if any other problems,

a warning message is displayed.

3.3.

When we use makecert.exe to create certificate,

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

4.

There are 2 popular ways to create Self-Signed Certificates

Use MakeCert.exe

Use IIS

4.1.

Use MakeCert.exe

Reference:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968(v=vs.85).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968%28v=vs.85%29.aspx)

E.g.

**Makecert -r** **-pe** **-n "CN=YourComputerName" -b 01/01/2000** **-eku 1.3.6.1.5.5.7.3.1** **-e 01/01/2100 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12**

It need to run "Command Prompt" as admin.

The certificate created by makecert.exe will **not** be trusted by the certificate authority.

4.2.

Use IIS

The certificate created by IIS will be trusted by the certificate authority.

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

1. New Project - OnlineGame

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

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

-->

Name: **OnlineGame**

Graphical user interface, application

Description automatically generated

1.1. New Project - OnlineGame.Web

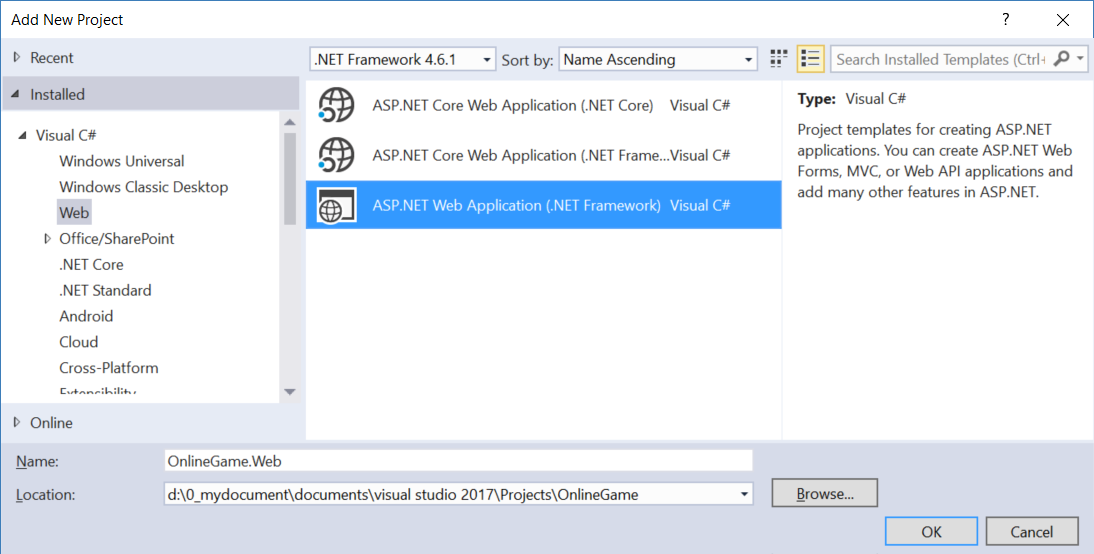
Solutions Name --> Add --> New Project -->

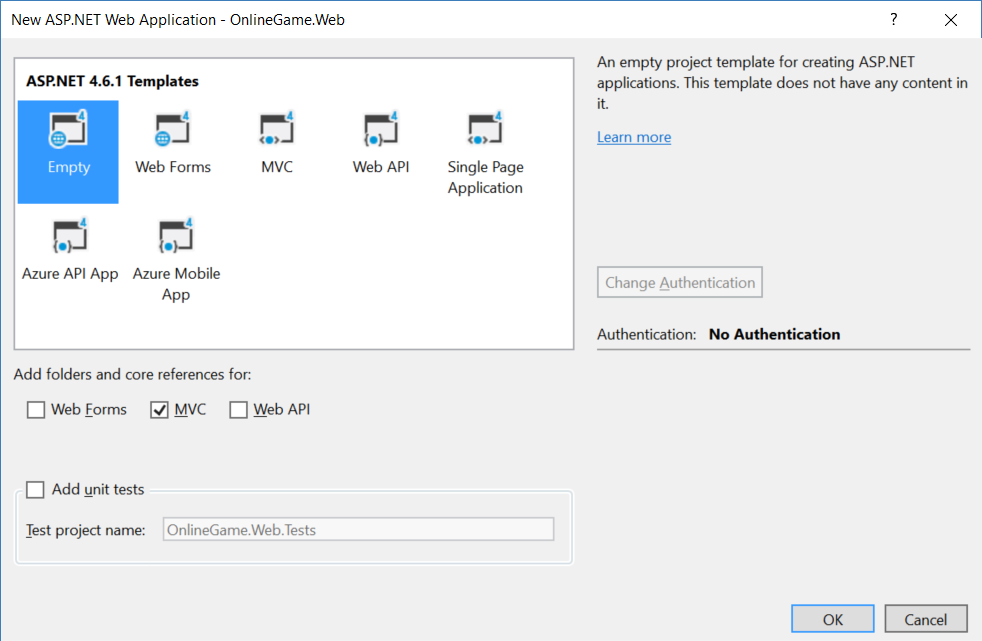
Visual C# --> Web --> [ASP.NET](http://asp.net/)Web Application (.Net Framework)

-->

Name: **OnlineGame.Web**

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





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](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 = "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-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.

\*/

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



<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>

1.1.5. 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" />

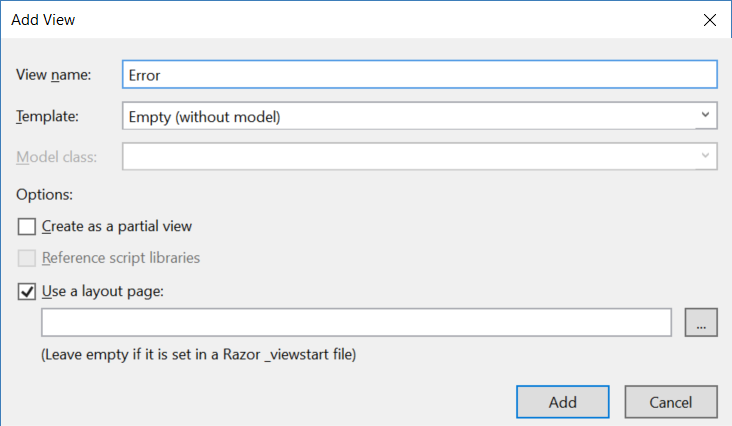
//    <error statusCode="404" redirect="Error/NotFound" />

//    <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>



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>

Text

Description automatically generated with medium confidence

1.1.5.5. Views/Shared/InternalServerError.cshtml

@{

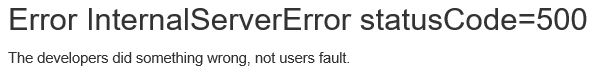
    ViewBag.Title = "InternalServerError";

}

<h2>Error InternalServerError statusCode=500</h2>

The developers did something wrong, not users fault.

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



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 thrwo exception

        [CustomizeCache("outputCacheProfile1")]

        public string GetGamerCount2()

        {

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

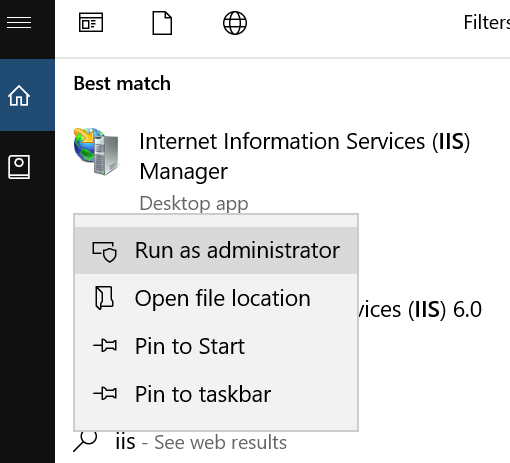
        }

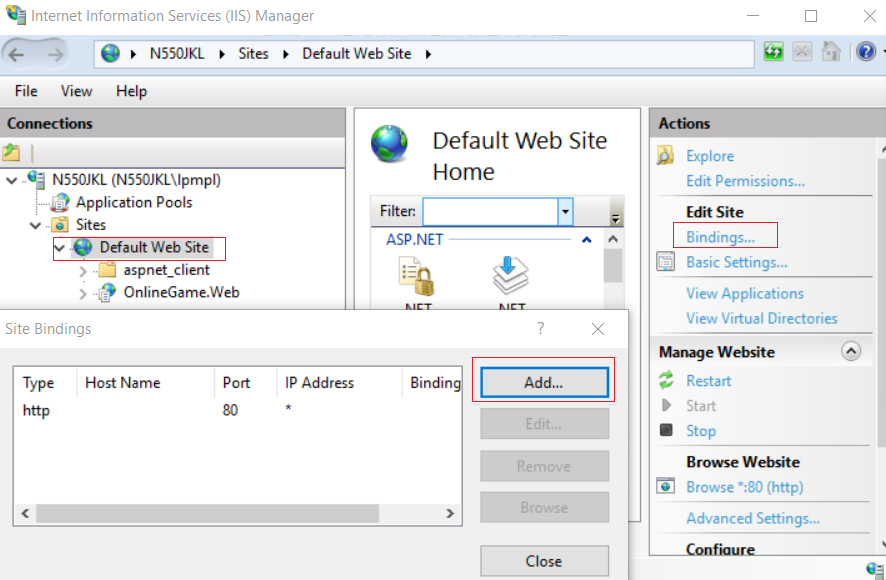
2. OnlineGame.Web - IIS

2.1. Run In IIS

Graphical user interface, text, application, email

Description automatically generated





Graphical user interface, text, application, email

Description automatically generated

Do not change anything,

This just shows you

HTTP use port 80

HTTPS use port 443

2.2. Controllers/HomeController.cs

using System.Web.Mvc;

//[RequireHttps] attribute will enforce to redirect to https

//It can apply to controller level or action level

//[RequireHttps]

namespace OnlineGame.Web.Controllers

{

    public class HomeController : Controller

    {

        // GET: Home

        public ActionResult Index()

        {

            return View();

        }

        public string Index2()

        {

            return "Index2 is accessible via HTTP or HTTPS";

        }

        public ActionResult Index3()

        {

            return View();

        }

             //[RequireHttps] attribute will enforce to redirect to https

        [RequireHttps]

        public string Index4()

        {

            return "Index4 is accessible via HTTPS";

        }

        [RequireHttps]

        public ActionResult Index5()

        {

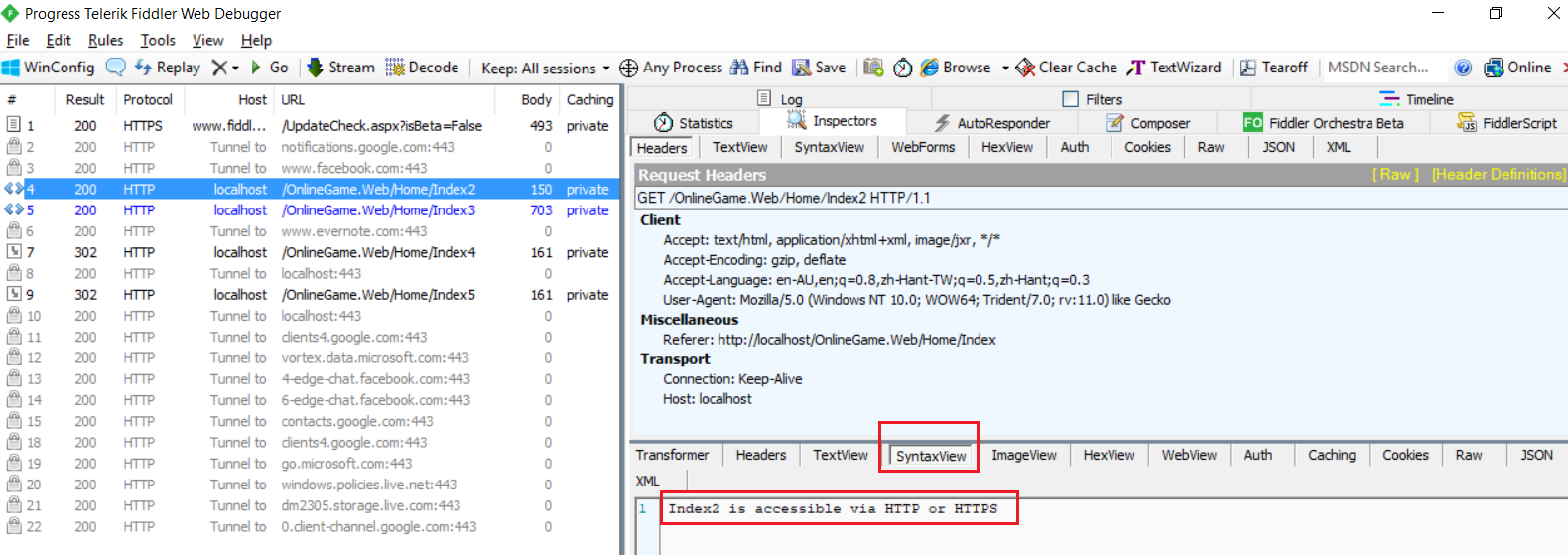
            return View();

        }

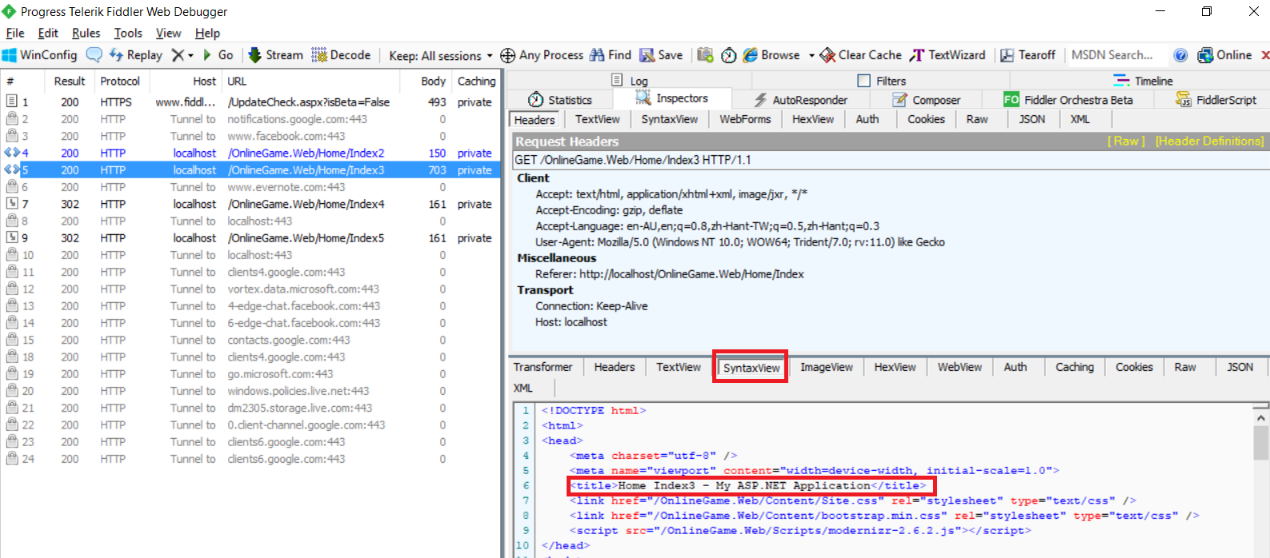
    }

}

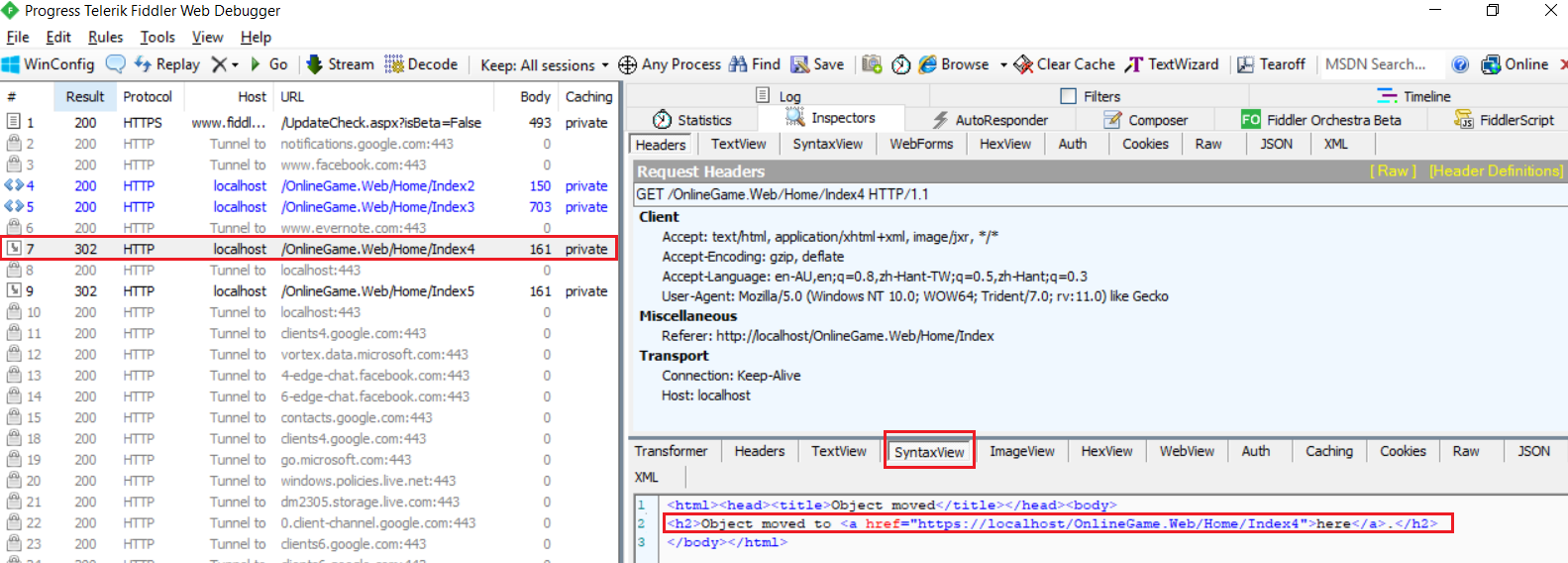
Http:// ... /Home/Index2

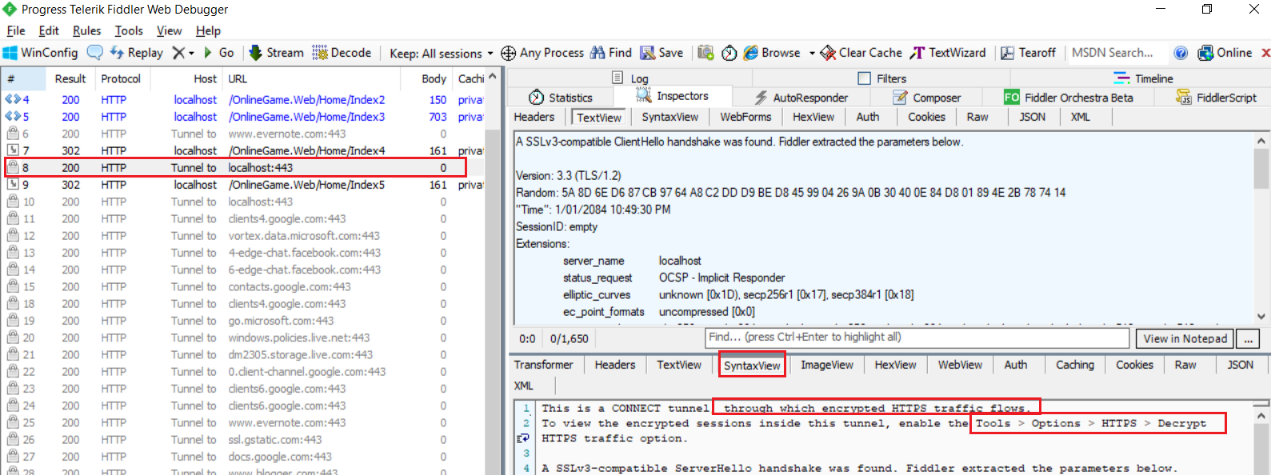


Http:// ... /Home/Index3

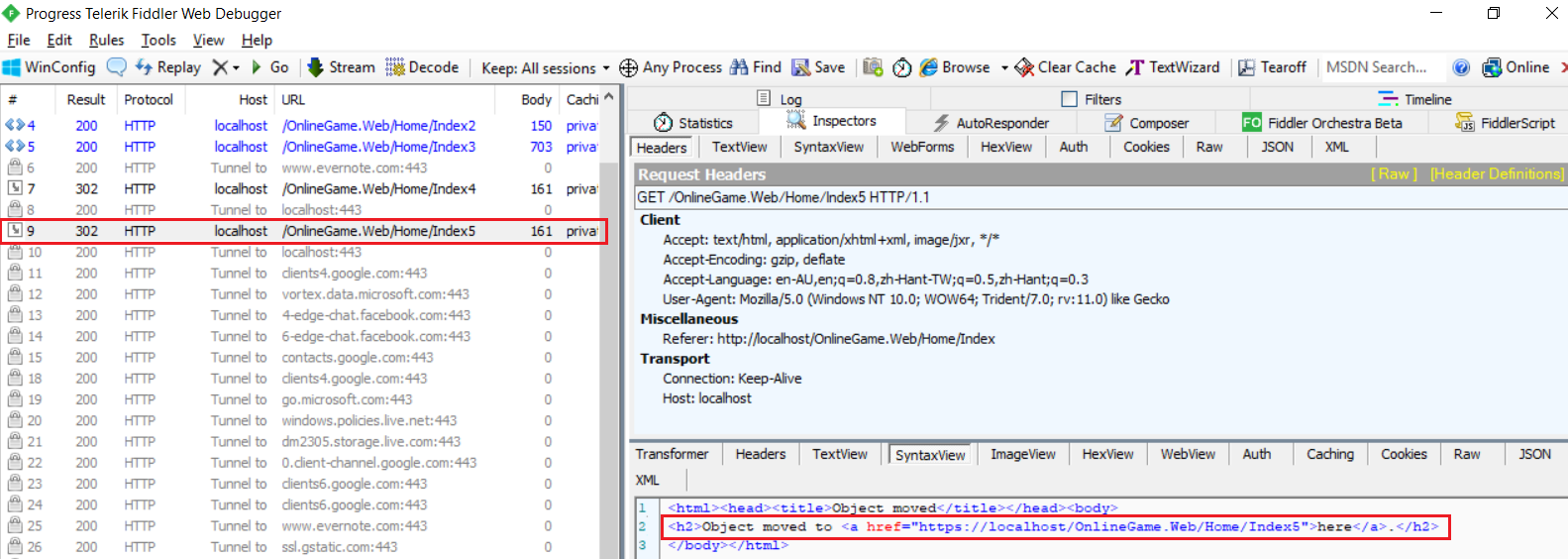


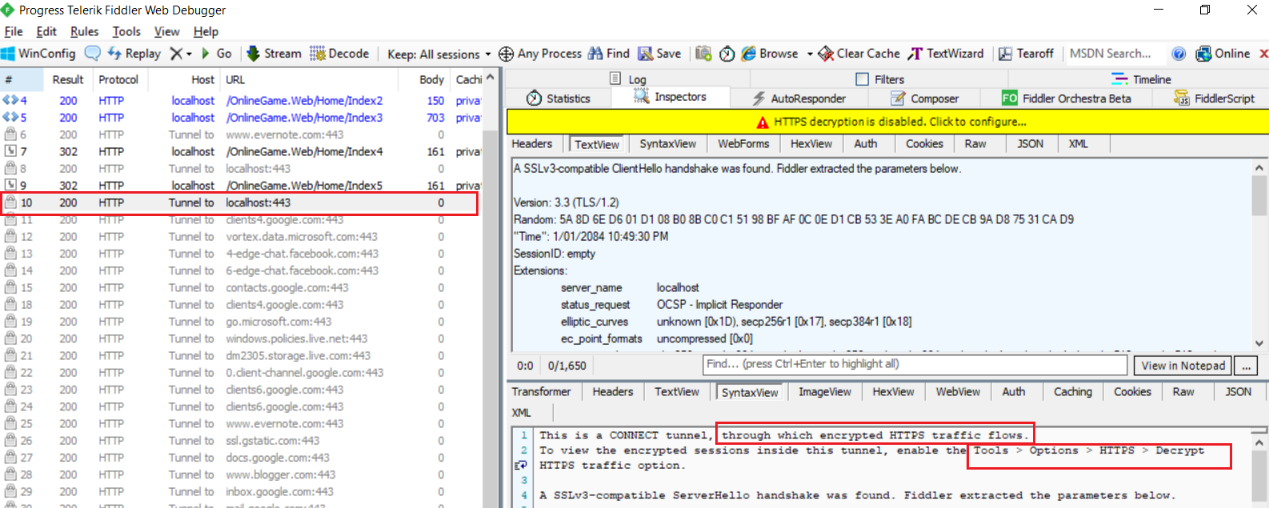
Https:// ... /Home/Index4





Https:// ... /Home/Index5





2.3. Views/Home/Index.cshtml

@{

    ViewBag.Title = "Home Index";

}

<h2>Home Index</h2>

Please try the following links via HTTP and HTTPS<br />

@Html.ActionLink("Index2", "Index2", "Home")<br/>

@Html.ActionLink("Index3", "Index3", "Home")<br />

@Html.ActionLink("Index4", "Index4", "Home")<br />

@Html.ActionLink("Index5", "Index5", "Home")<br />

2.4. Views/Home/Index3.cshtml

@{

    ViewBag.Title = "Home Index3";

}

<h2>Home Index3</h2>

2.5. Views/Home/Index5.cshtml

@{

    ViewBag.Title = "Home Index5";

}

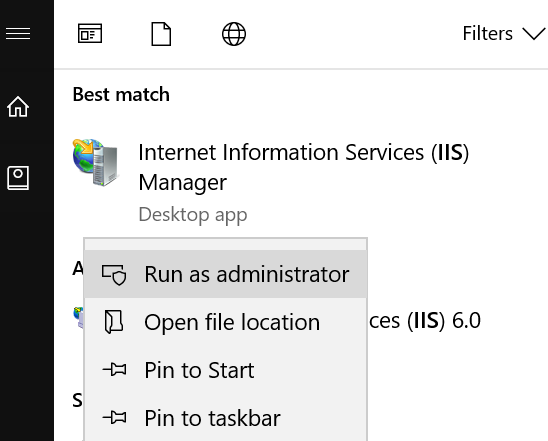
<h2>Home Index5</h2>

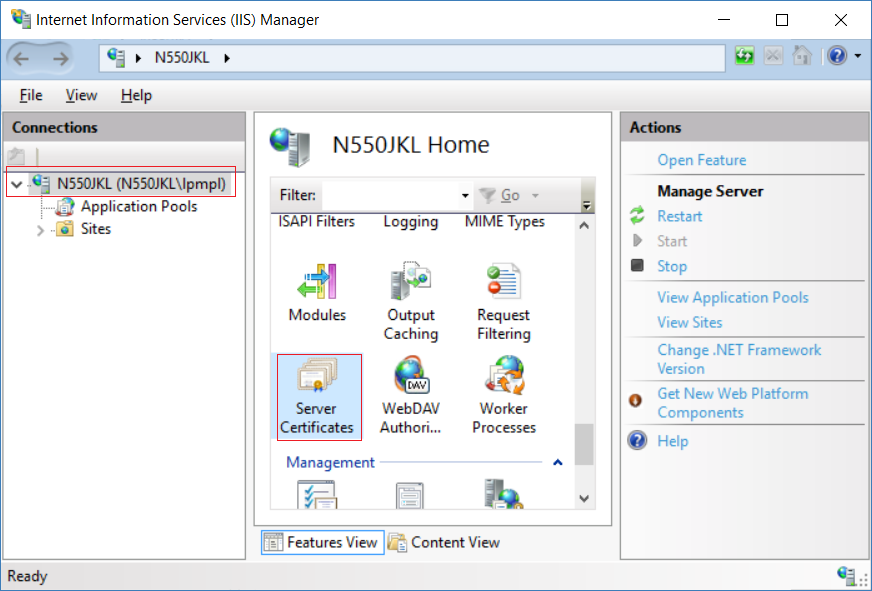
3. OnlineGame.Web - Certificate

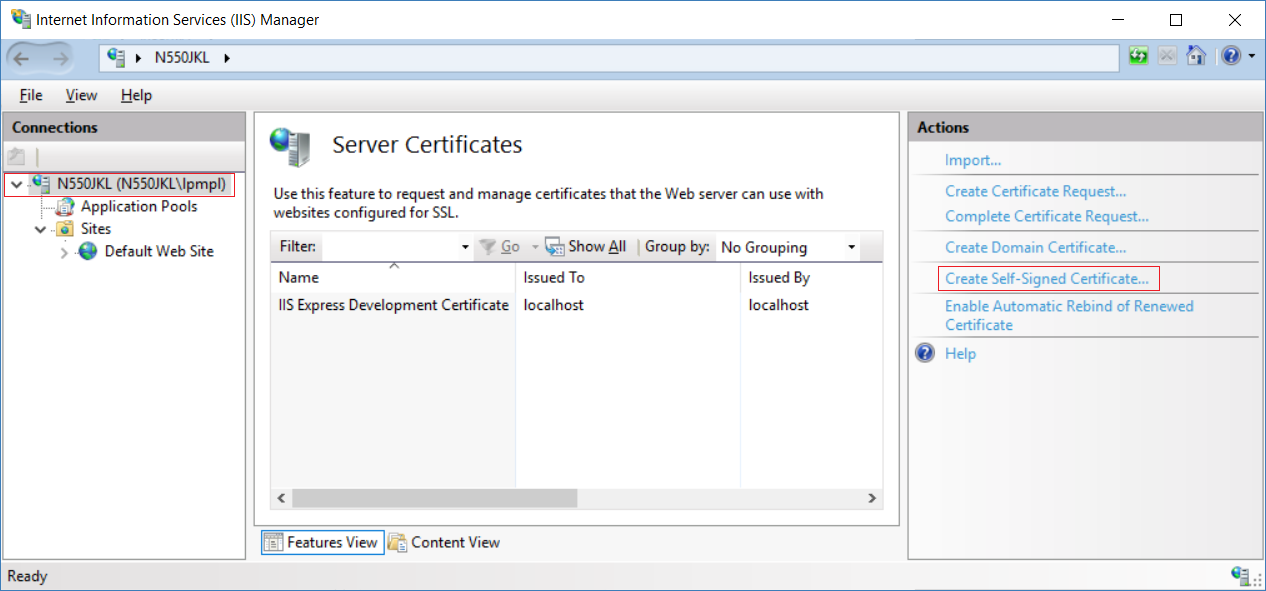
3.1. Use IIS to Create Self-Signed Certificates

There are 2 popular ways to create Self-Signed Certificates

* Use MakeCert.exe
* **Use IIS**

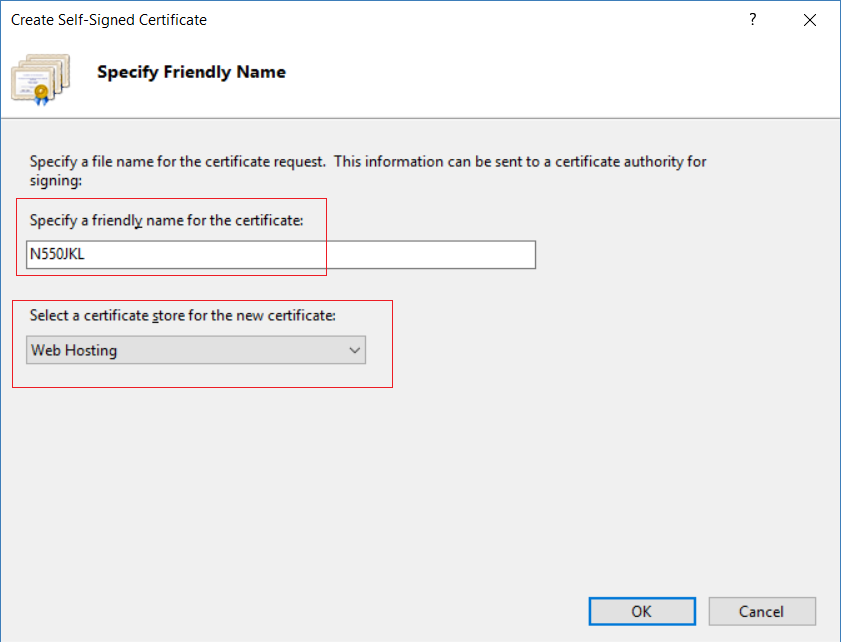


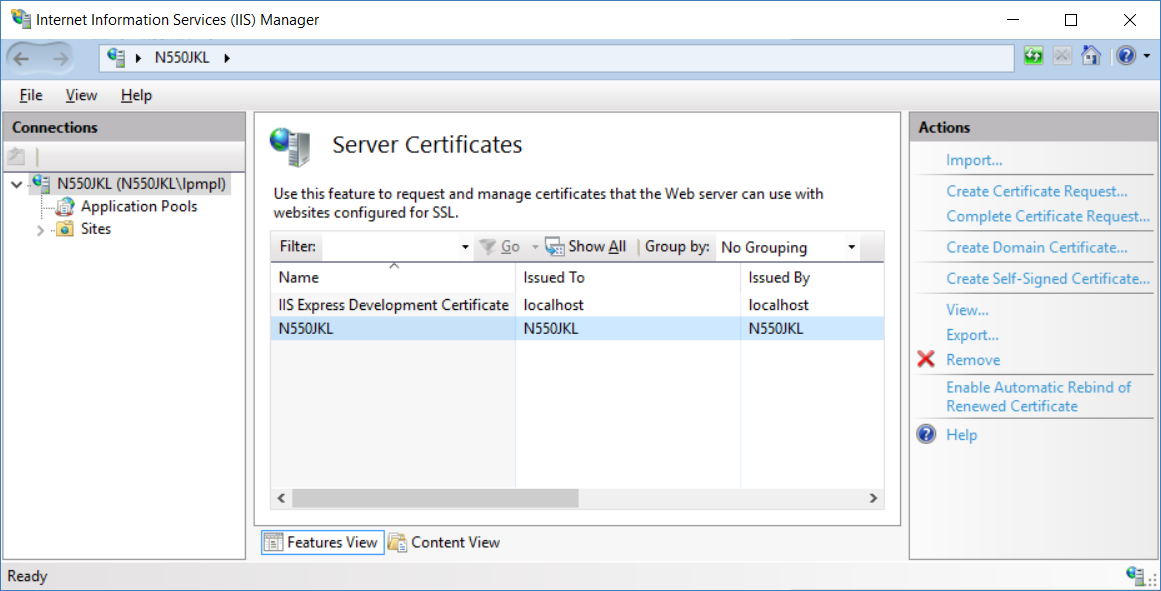




I used my computer name as my certificate name,

because my computer is the certificate issuer.



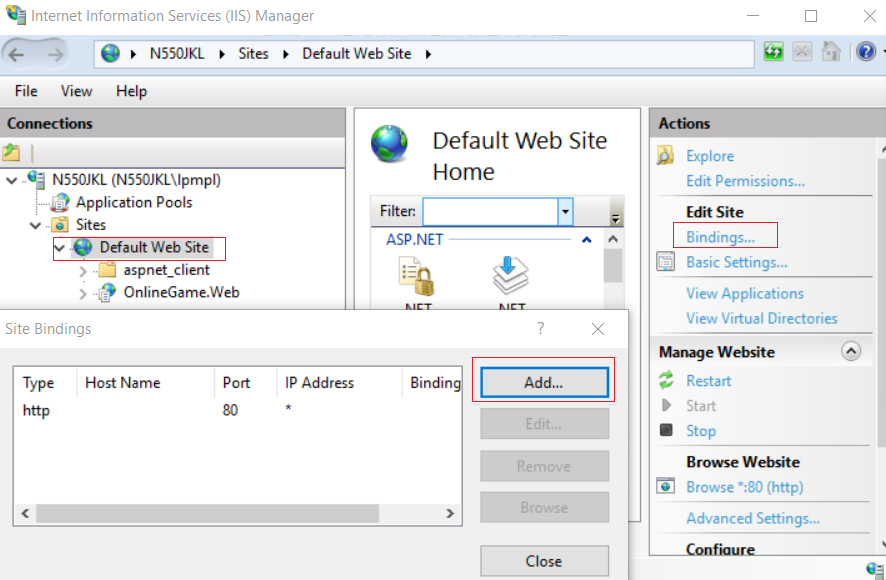


3.2. Add the Self-Signed Certificates to HTTPS site binding - Assign Web Applications with the specific certificate

Previously, we use IIS to create the Self-Signed Certificates.

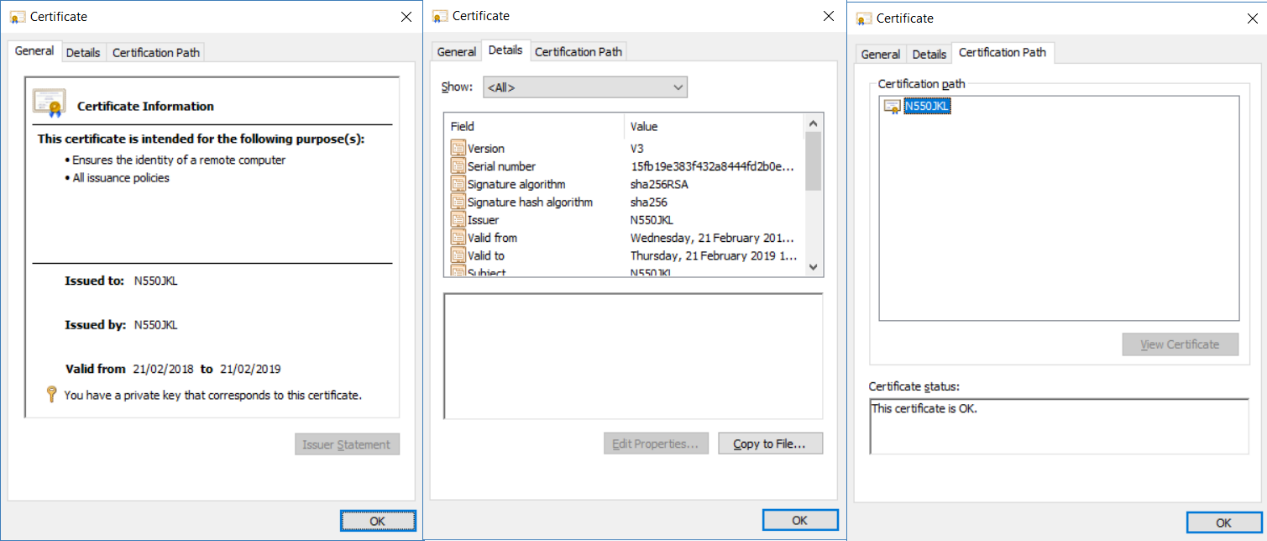
Now we need to assign the Self-Signed Certificates to our web application.

Firstly, we have to add the Self-Signed Certificates to HTTPS site binding



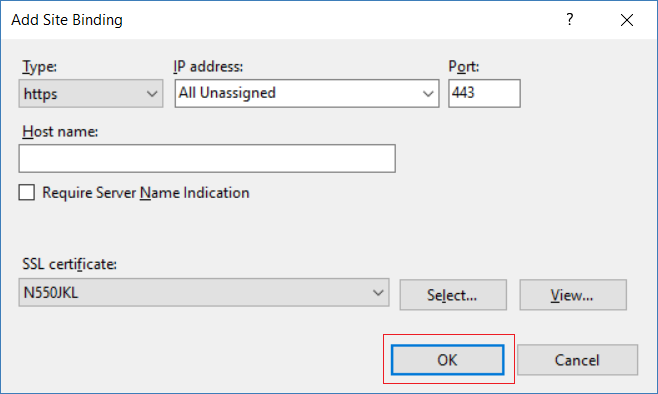
Graphical user interface, application

Description automatically generated



HTTP use port 80

HTTPS use port 443



Graphical user interface, text, application

Description automatically generated

Please try to use HTTP navigate your controller actions.

E.g.

Https:// ....  Home/Index

Https:// ....  Home/Index2

Https:// ....  Home/Index3

Https:// ....  Home/Index4

Https:// ....  Home/Index5

3.3. Dis-allow a Web Application use HTTP Protocol

3.3.1. Remove HTTP Binding

There are 2 ways to dis-allow a Web Application use HTTP Protocol

A.

Remove HTTP Binding.

This will dis-allow all Web Applications on that IIS using HTTP Protocol.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

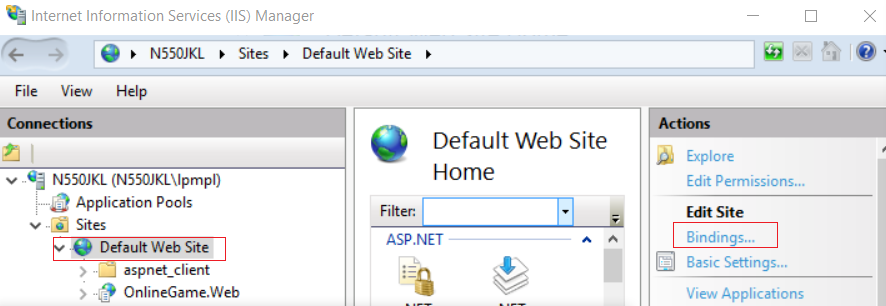
Description automatically generated

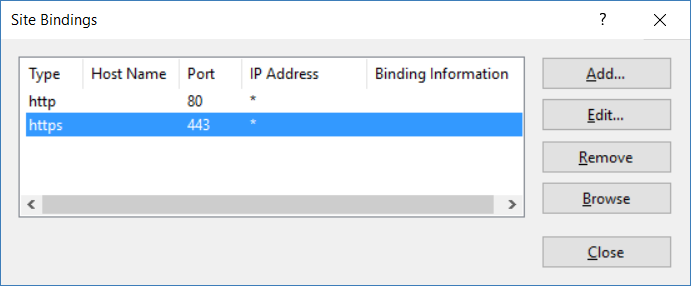
3.3.2. Set the application always require SSL

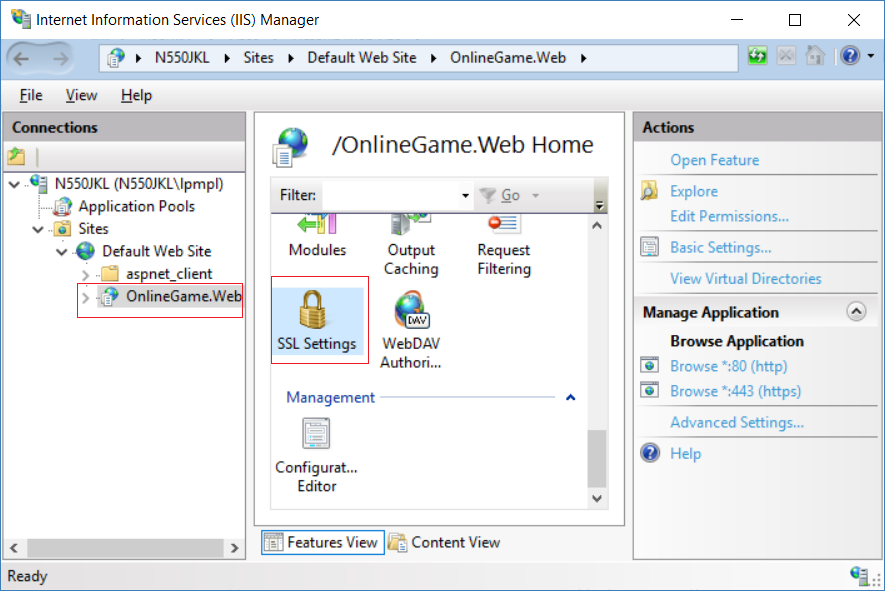
In Server Level, Let IIS still allow HTTP and HTTPS binding.

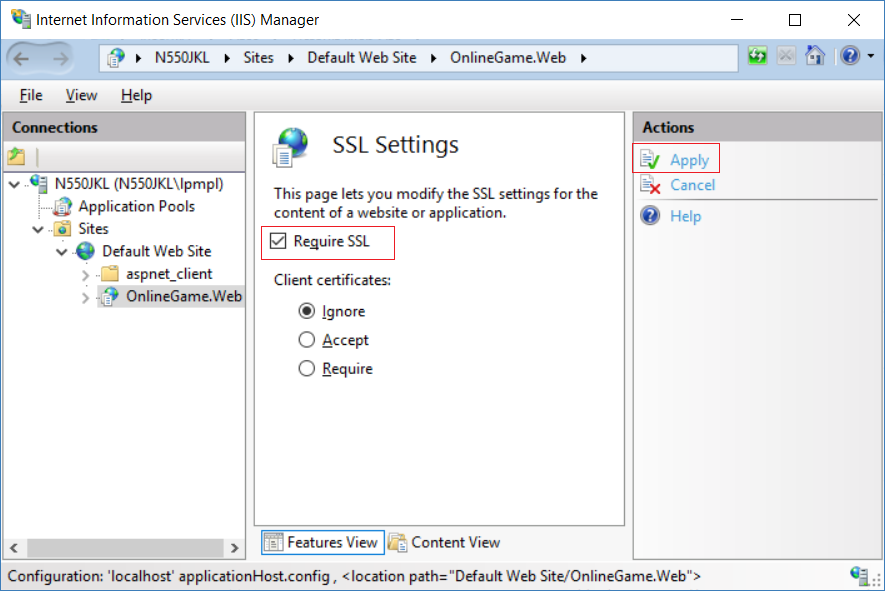
In the application level, set the application always require SSL

This will dis-allow only that Web Application on that IIS using HTTP Protocol.



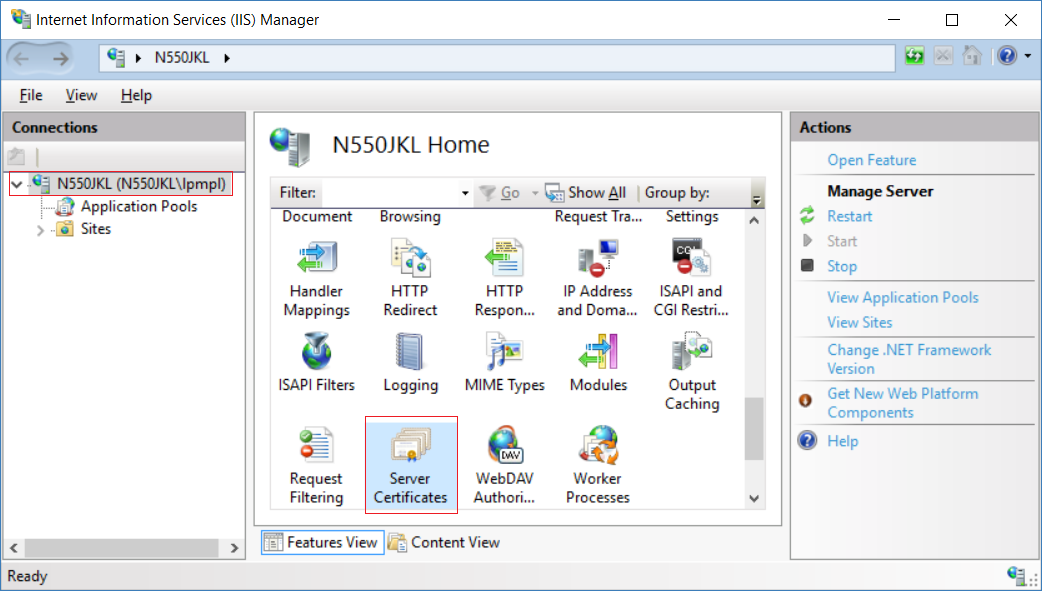


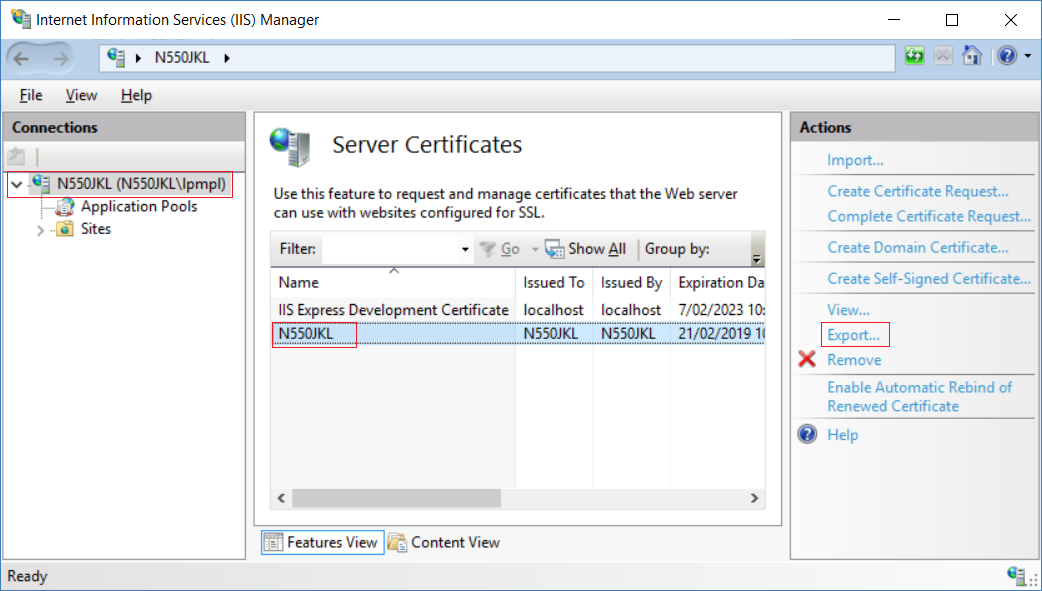




3.4. Import / Export Certificates

3.4.1. Export Certificates





C:\YourComputerName\_Certificate

1234

1234

-->

It will create

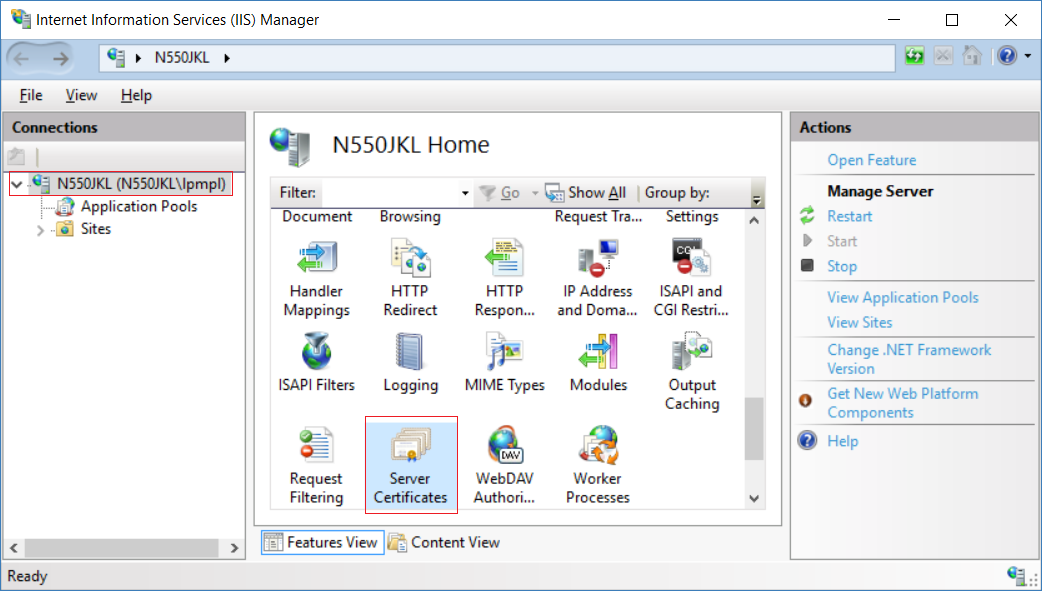
**C:\YourComputerName\_Certificate.pfx**

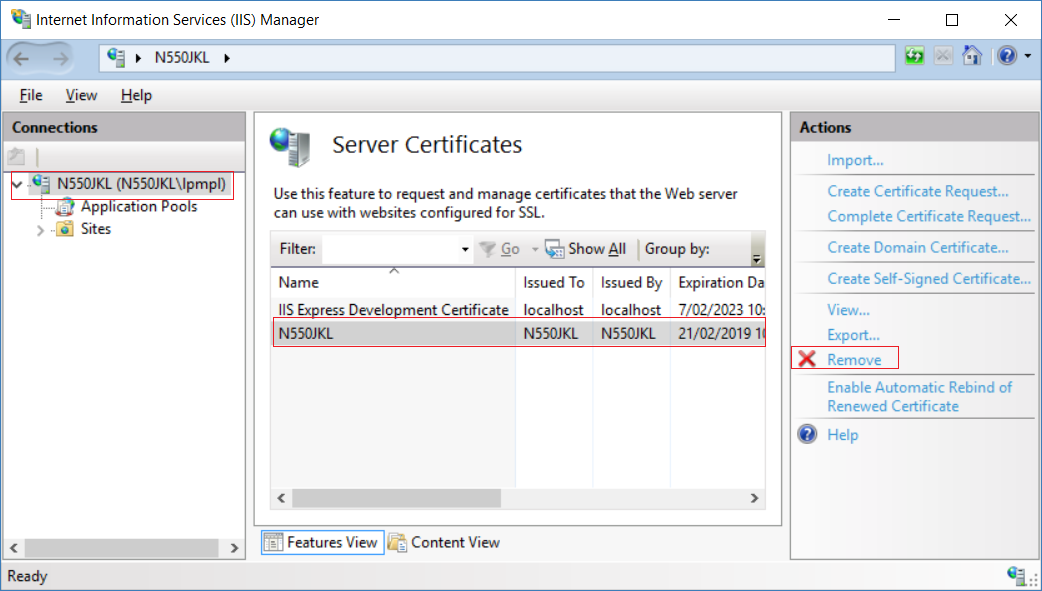


3.4.2. Import Certificates

Before we import the certificate, **C:\YourComputerName\_Certificate.pfx**

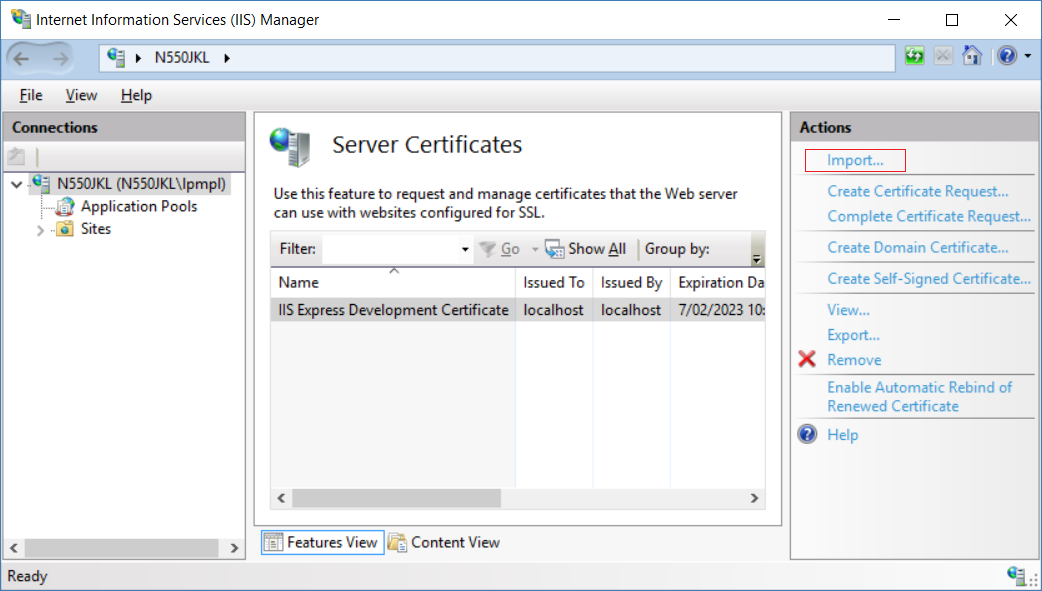
We have to delete the "YourComputerName" first.





After we delete the "YourComputerName".

We can import the certificate, **C:\YourComputerName\_Certificate.pfx**



C:\N550JKL\_Certificate.pfx

1234

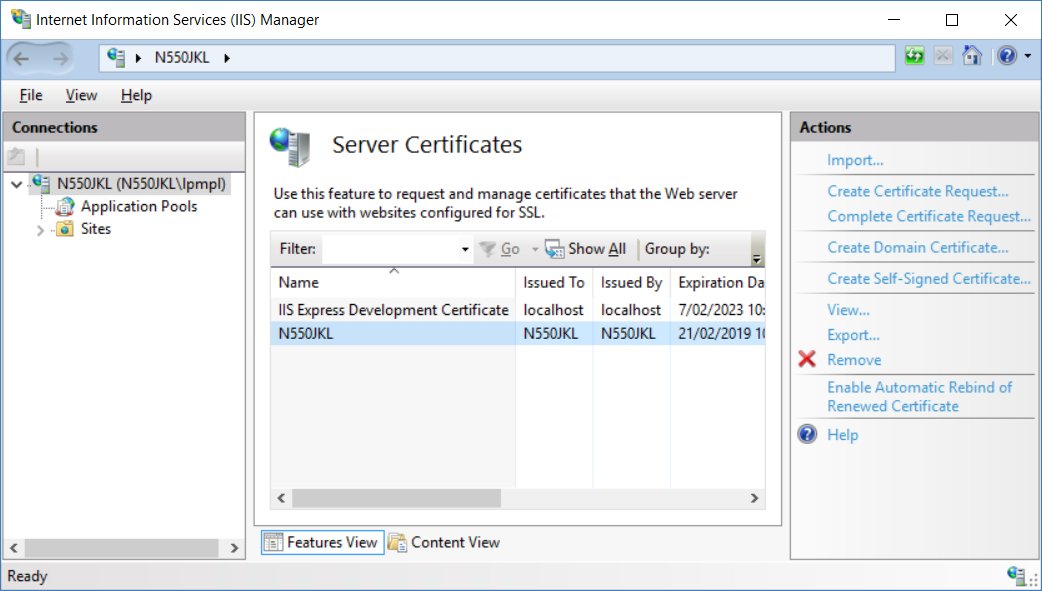
Select "Allow this certificate to be exported"

If we didn't select "Allow this certificate to be exported"

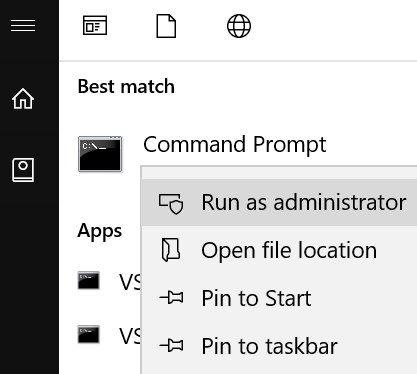
then we will not be able to export this certificate any more.

Graphical user interface, text, application, email

Description automatically generated

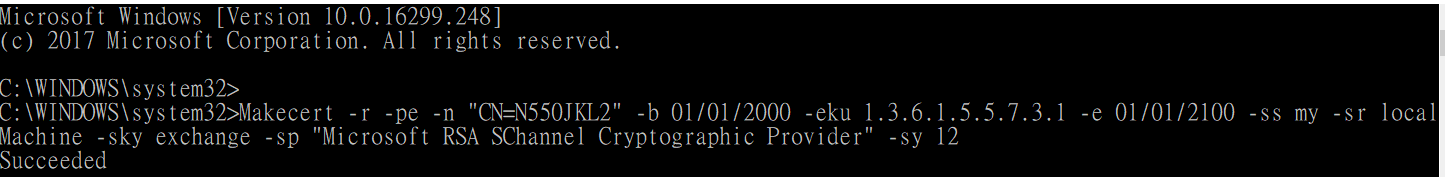


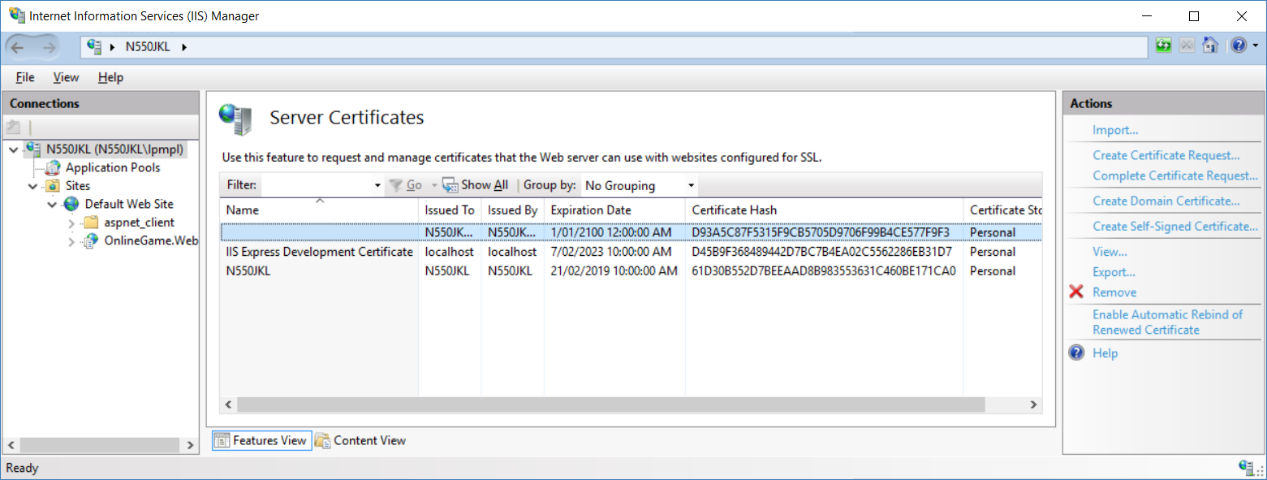
3.5. Use MakeCert.exe to Create Self-Signed Certificates

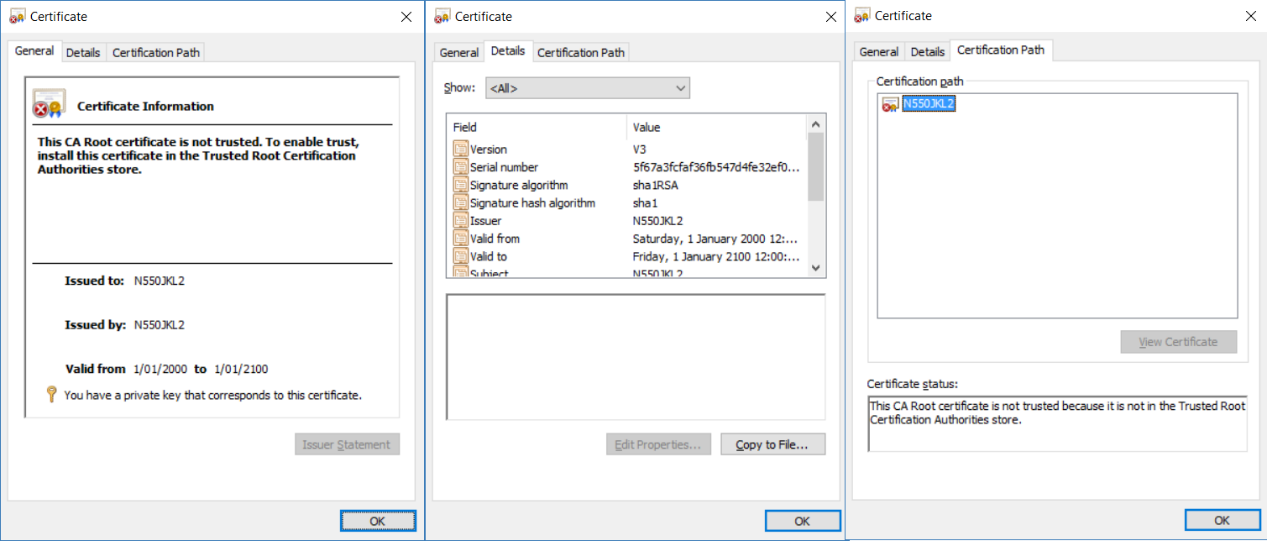


Makecert -r -pe -n "CN=N550JKL2" -b 01/01/2000 -eku 1.3.6.1.5.5.7.3.1 -e 01/01/2100 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12









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

3.

3.1.

The following are some trusted certificate authority who can issue server certificates.

A. GoDaddy

B. verisign

C. Thawte

D. Geotrust

E. Comodo

3.2.

The certificate authority assures the client side browsers by verifying the server’s identity over the Internet.

When a client side browser requests over https,

it also requests the server certificate and sees if the site is in trusted sites list provided by the certificate authority.

If the server certificate was not already signed by the client side browser

or if the site is in trusted sites list provided by the certificate authority

or if any other problems,

a warning message is displayed.

3.3.

When we use makecert.exe to create certificate,

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

4.

There are 2 popular ways to create Self-Signed Certificates

Use MakeCert.exe

Use IIS

4.1.

Use MakeCert.exe

Reference:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968(v=vs.85).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968%28v=vs.85%29.aspx)

E.g.

**Makecert -r** **-pe** **-n "CN=YourComputerName" -b 01/01/2000** **-eku 1.3.6.1.5.5.7.3.1** **-e 01/01/2100 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12**

It need to run "Command Prompt" as admin.

The certificate created by makecert.exe will **not** be trusted by the certificate authority.

4.2.

Use IIS

The certificate created by IIS will be trusted by the certificate authority.

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

Reference:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968(v=vs.85).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa386968%28v=vs.85%29.aspx)

Open Command Prompt

**Makecert -r** **-pe** **-n "CN=YourComputerName" -b 01/01/2000 -e 01/01/2100 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12**

**Makecert -r** **-pe** **-n "CN=YourComputerName" -b 01/01/2000** **-eku 1.3.6.1.5.5.7.3.1** **-e 01/01/2100 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12**

Note: Replace YourComputerName, with the name of your computer.

-r

Creates a self-signed certificate.

**-pe**

Marks the private key as exportable.

That means the private key can be part of certificate and we can export the certificate

-n "Name"

Name for the publisher's certificate.

This name must conform to the X.500 standard.

The simplest method is to use the "CN=MyName" format. For example: -n "CN=Test".

E.g. -n "CN=YourComputerName"

-b DateStart

Date the certificate first becomes valid.

The default is when the certificate is created.

The format of DateStartis mm/dd/yyyy.

E.g.

-b 01/01/2000

-e DateEnd

Date when the validity period ends. The default is the year 2039.

-e 01/01/2100

-ss SubjectCertStoreName

Name of the subject's certificate store where the generated certificate will be stored.

E.g.

-ss my

-sr SubjectCertStoreLocation

Registry location of the subject's certificate store.

SubjectCertStoreLocation must be either **LocalMachine**(registry key **HKEY\_LOCAL\_MACHINE**) or **CurrentUser** (registry key **HKEY\_CURRENT\_USER**). **CurrentUser** is the default.

E.g.

-sr localMachine

-sky SubjectKeySpec

Subject's key specification.

SubjectKeySpec must be one of three possible values:

\* **Signature** (**AT\_SIGNATURE** key specification)

\* **Exchange** (**AT\_KEYEXCHANGE** key specification)

\* An **integer**, such as **3**

E.g.

-sky exchange

-sp SubjectProviderName

**CryptoAPI provider** for subject. The default is the user's provider.

For information about CryptoAPI providers, see the CryptoAPI 2.0 documentation.

E.g.

-sp "Microsoft RSA SChannel

-sy nSubjectProviderType

CryptoAPI provider type for subject. The default is PROV\_RSA\_FULL.

For information about CryptoAPI provider types, see the CryptoAPI 2.0 documentation.

E.g.

-sy 12

-eku OID1, OID2 …

Inserts a list of one or more comma-separated, enhanced key usage object identifiers (OIDs) into the certificate. For example, -eku 1.3.6.1.5.5.7.3.2 inserts the client authentication OID. For definitions of allowable OIDs, see the Wincrypt.h file in CryptoAPI 2.0.

E.g.

-eku 1.3.6.1.5.5.7.3.1

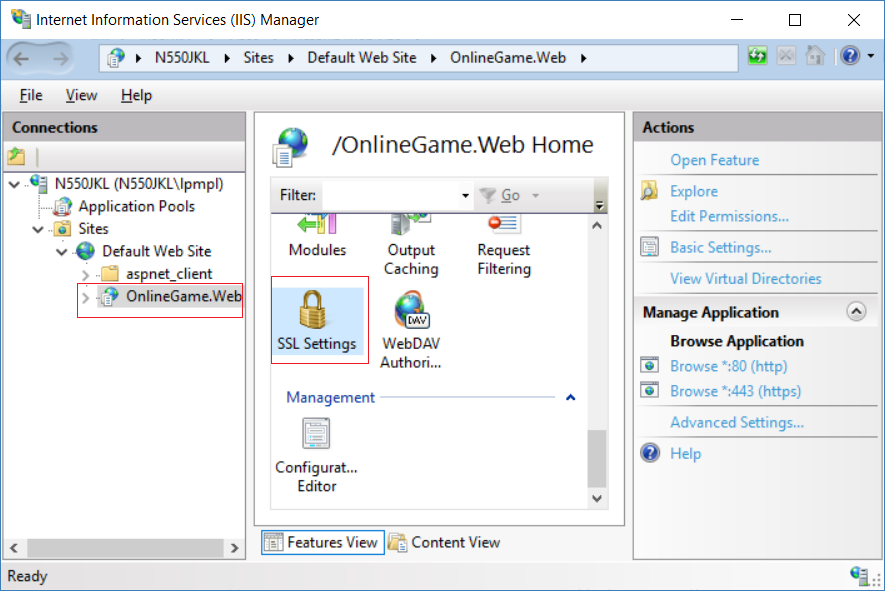
3.6. Error Pages - Status Code: 403.4

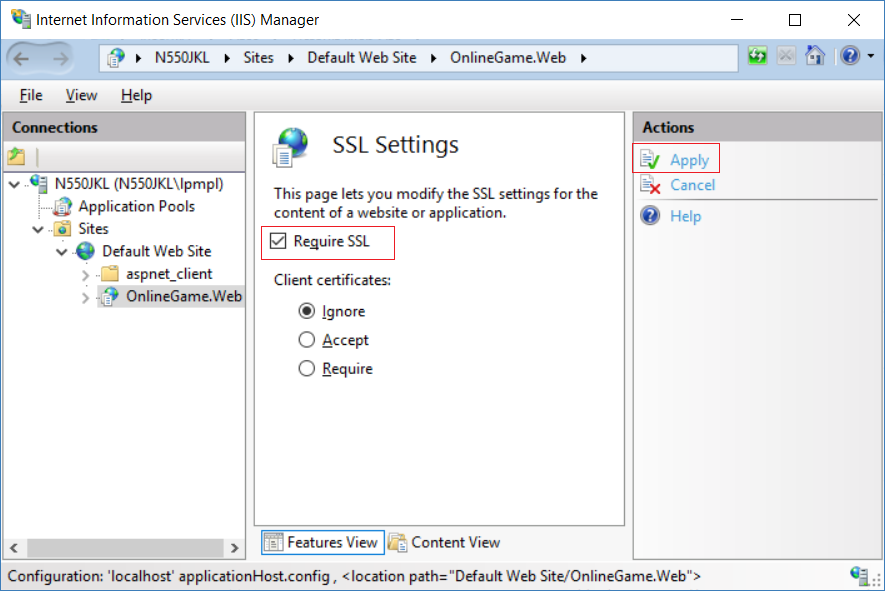
Error Pages is the place which lists a location of all Error Pages.

Let's try the following scenario.

A.

Enable the SSL required

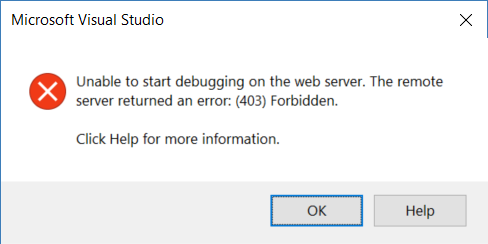




B.

Run application

The status code is actually **403.4**



C.

Create C:\inetpub\RedirectToHttps.html

The the following code will make HTTP redirect to HTTPS

<html>

<head>

    <title>

        Redirecting to HTTPS

    </title>

</head>

<script language="JavaScript">

    function redirectHttpToHttps() {

        var httpURL = window.location.hostname + window.location.pathname;

        var httpsURL = "https://" + httpURL;

        window.location = httpsURL;

    }

    redirectHttpToHttps();

</script>

<body>

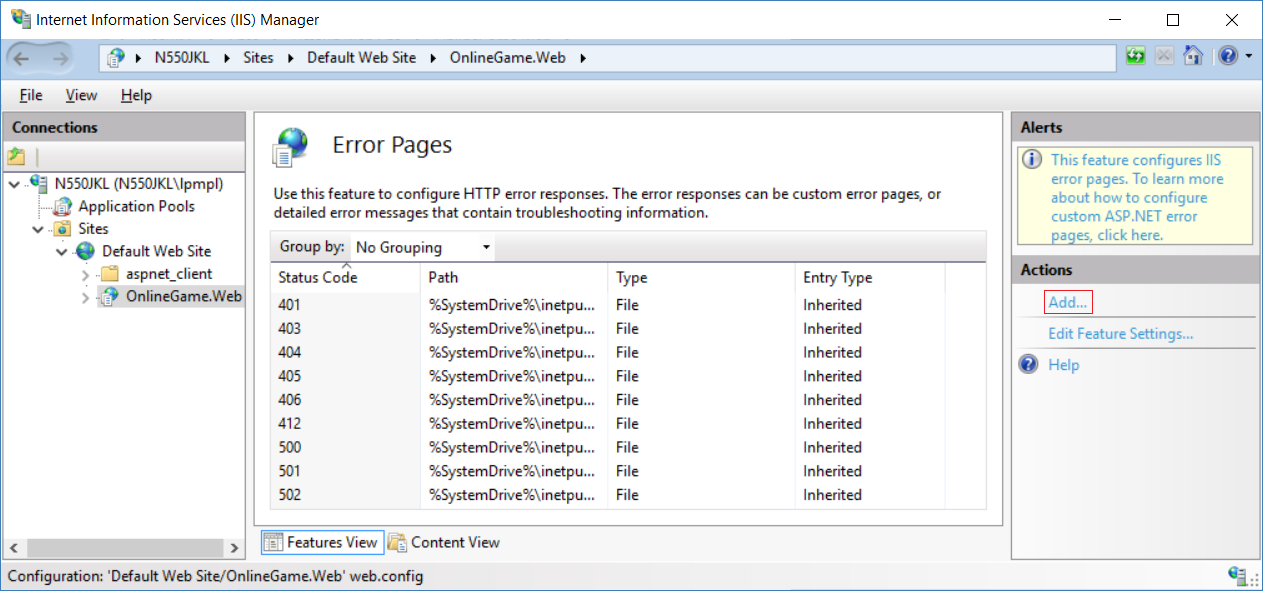
</body>

</html>

D.

Add Error Pages.





403.4

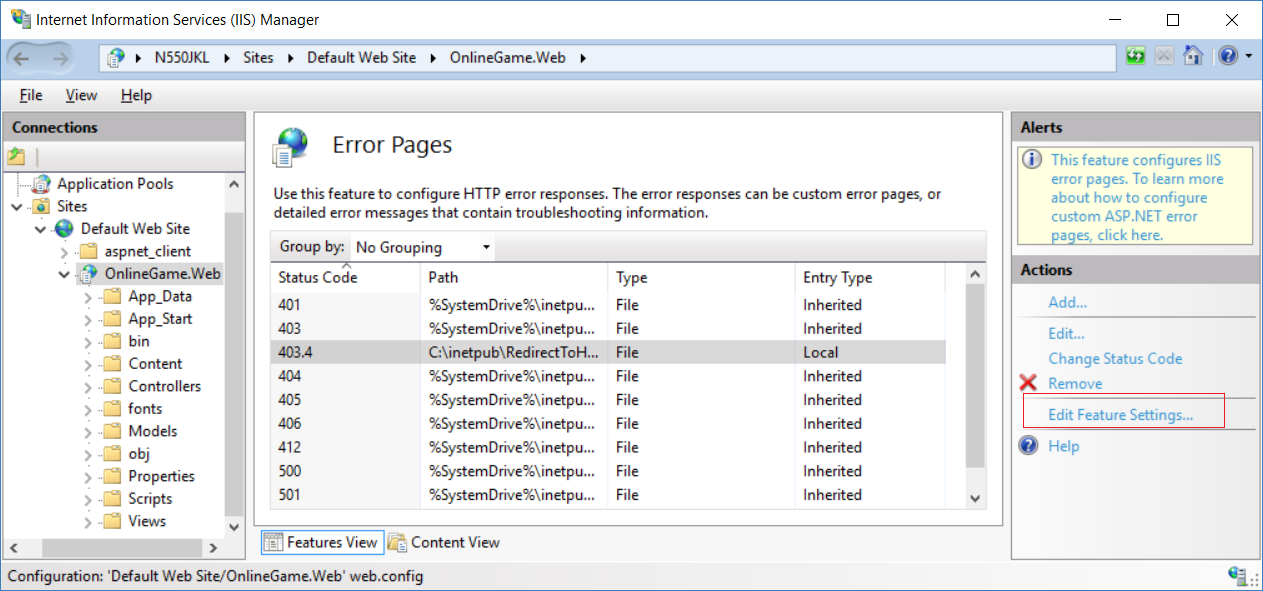
C:\inetpub\RedirectToHttps.html

Graphical user interface, text, application

Description automatically generated

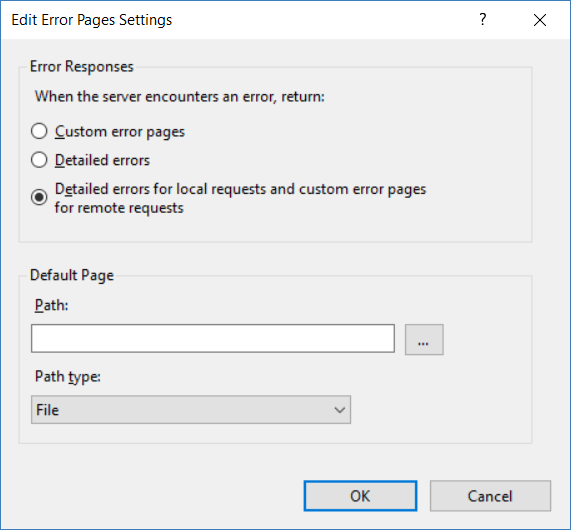
E.

Edit Feature Settings



By Default, it select "**Details error for local requests and custom error pages for remote requests**"

"Details Errors" means the default error pages which show all the details.



We tries to make HTTP request redirect to HTTPS.

For Testing, we select "**Custom Error Pages**"

Graphical user interface, application

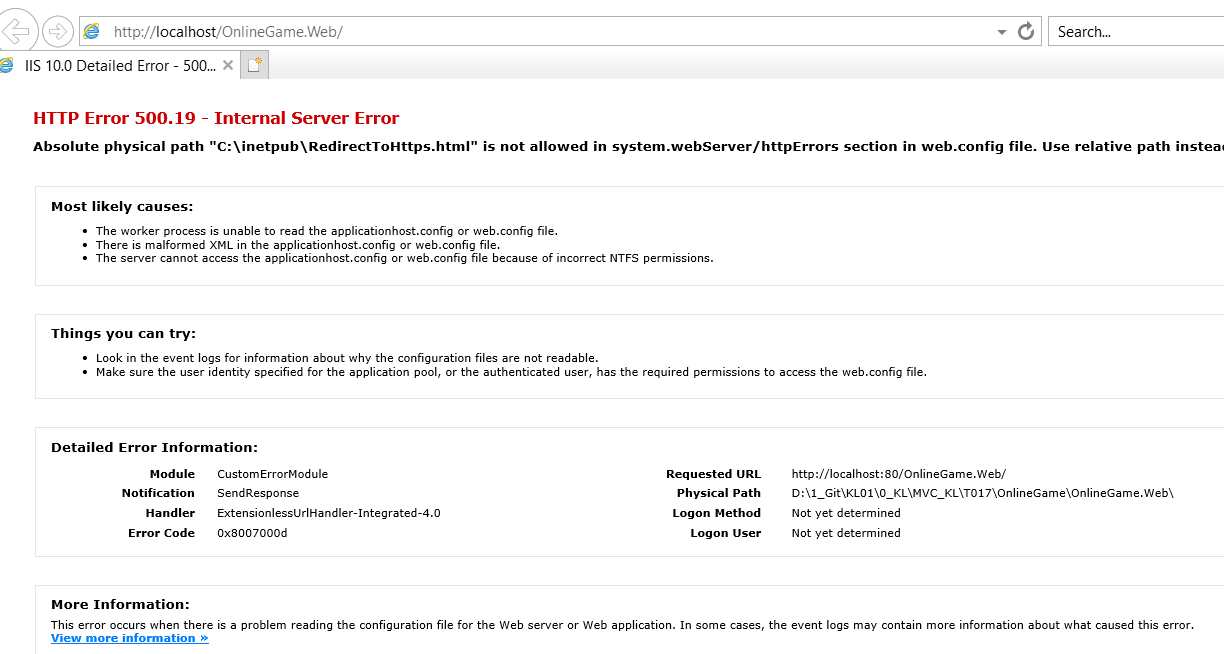
Description automatically generated

F.

Run the application

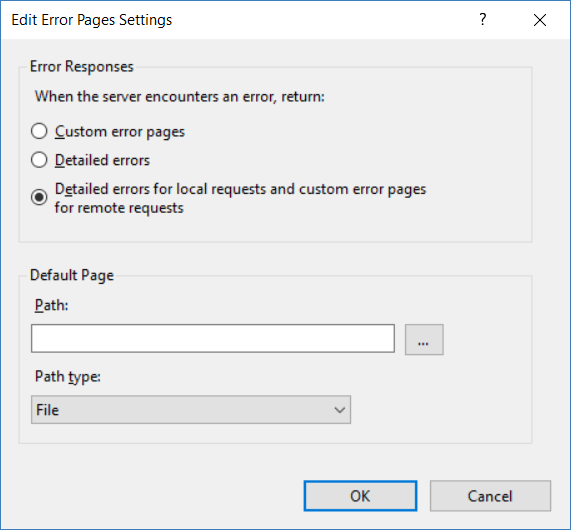
We still get the default details error page for status code 500.19

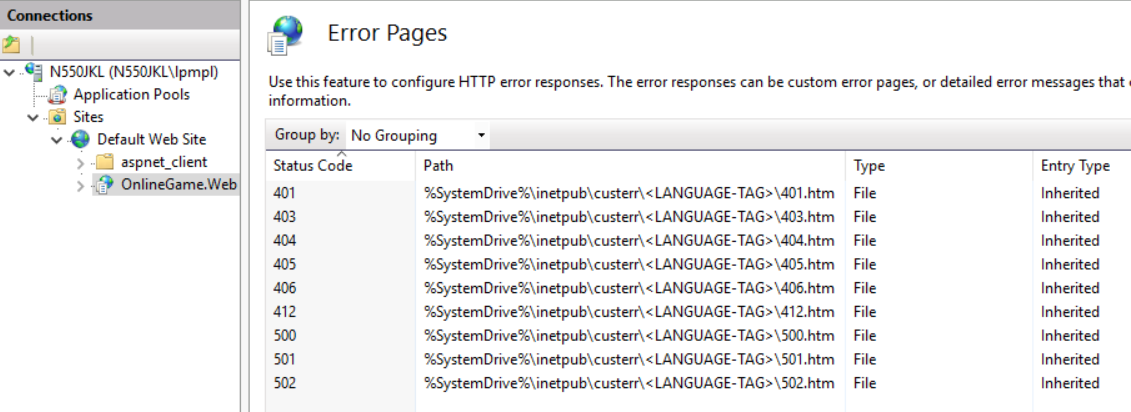
Therefore, In MVC, please use [RequireHttps]



G.

Please un-do everything.





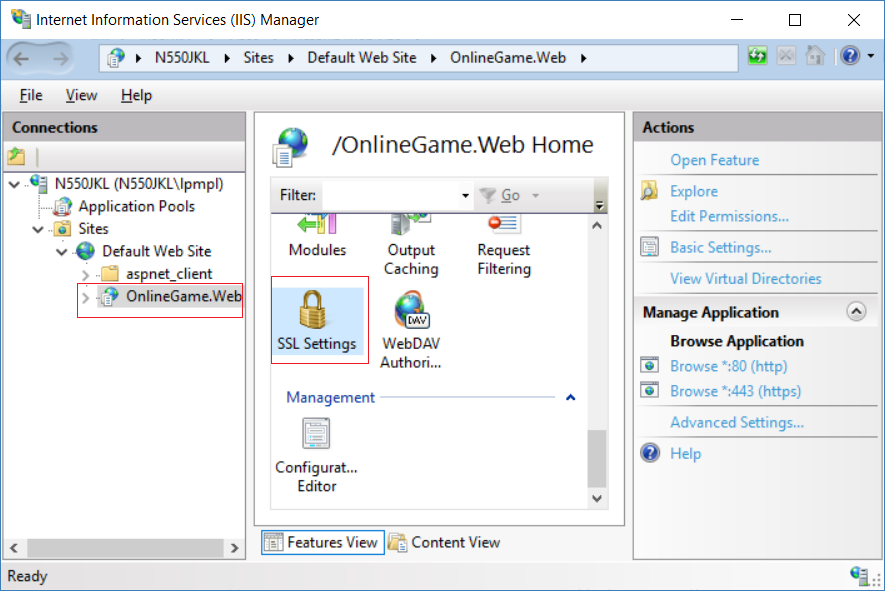
3.7. Error Controller - Status Code: 403.4

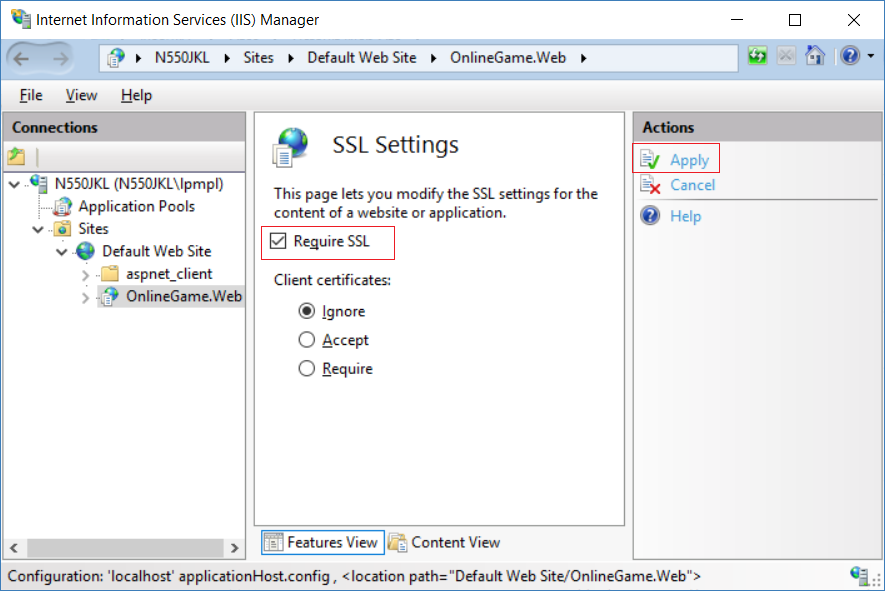
Error Pages is the place which lists a location of all Error Pages.

Let's try the following scenario.

A.

Enable the SSL required

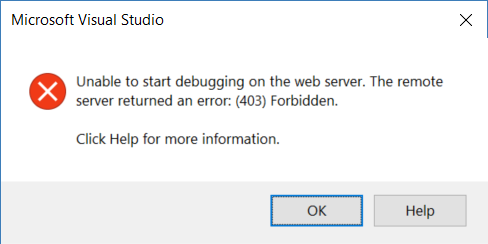




B.

Run application

The status code is actually **403.4**



3.7.1. 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="403.4" redirect="Error/RedirectToHttps" />

      <error statusCode="403" redirect="Error/RedirectToHttps" />

      <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 targetFram

3.7.2. 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();

        }

        //error statusCode="403.4"

        [HttpGet]

        public ActionResult RedirectToHttps()

        {

            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>

\*/

3.7.3. Views/Shared/RedirectToHttps.cshtml

@{

    ViewBag.Title = "RedirectToHttps";

}

<h2>RedirectToHttps</h2>

<script language="JavaScript">

    function redirectHttpToHttps() {

        var httpURL = window.location.hostname + window.location.pathname;

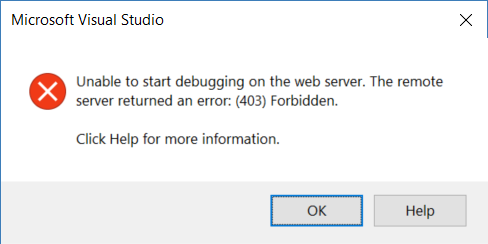
        var httpsURL = "https://" + httpURL;

        window.location = httpsURL;

    }

    redirectHttpToHttps();

</script>



Run the application

We still get the default details error page for status code 403.4

Therefore, In MVC, please use [RequireHttps]