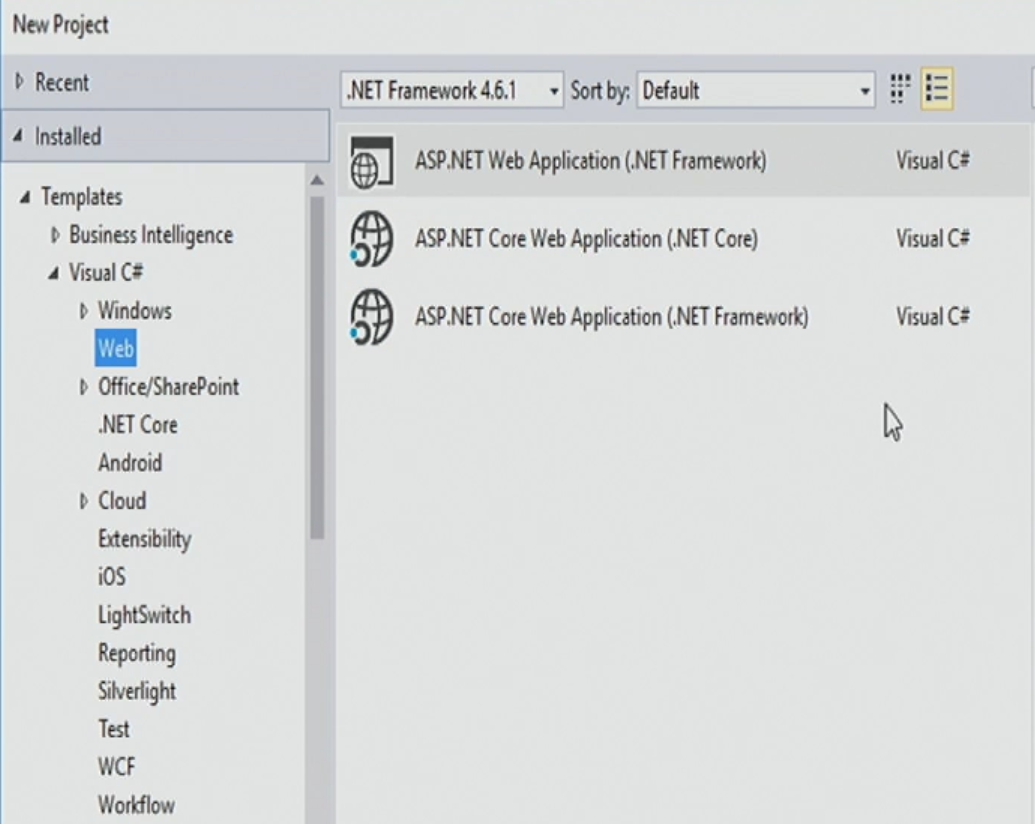
Build a custom module and deploy the solution to Azure. The custom http module should prevent access to the application if the request is coming from the following IP address:[**108.92.106.178**](https://whatismyipaddress.com/ip/108.92.106.178) .

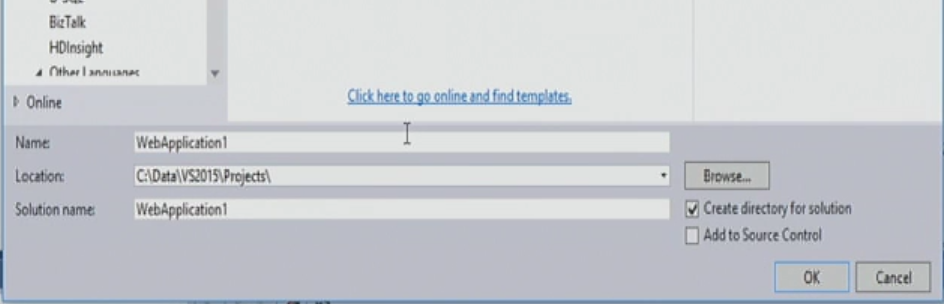
Your submission need to include just the following two information:

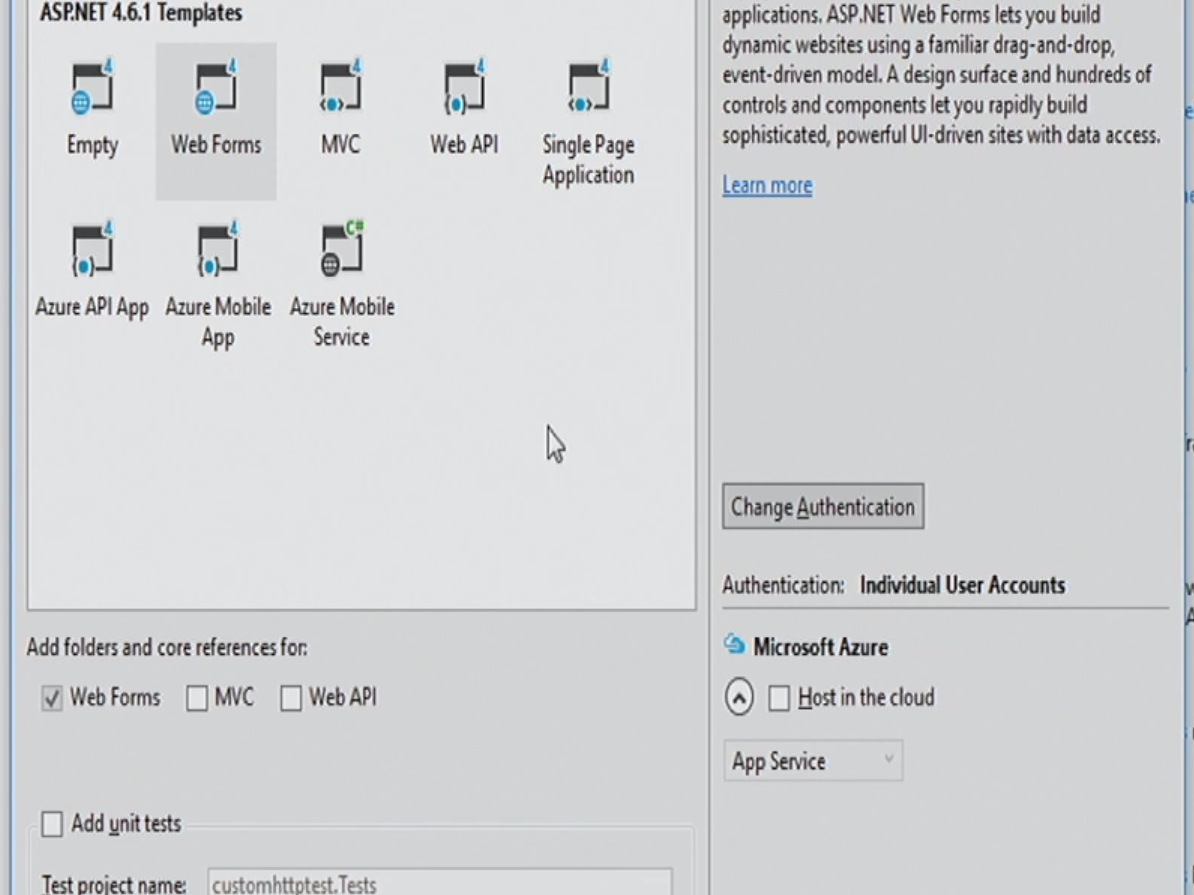
Source code url (Visual Studio Team foundation) that has access to rajkrihsnan@live.com

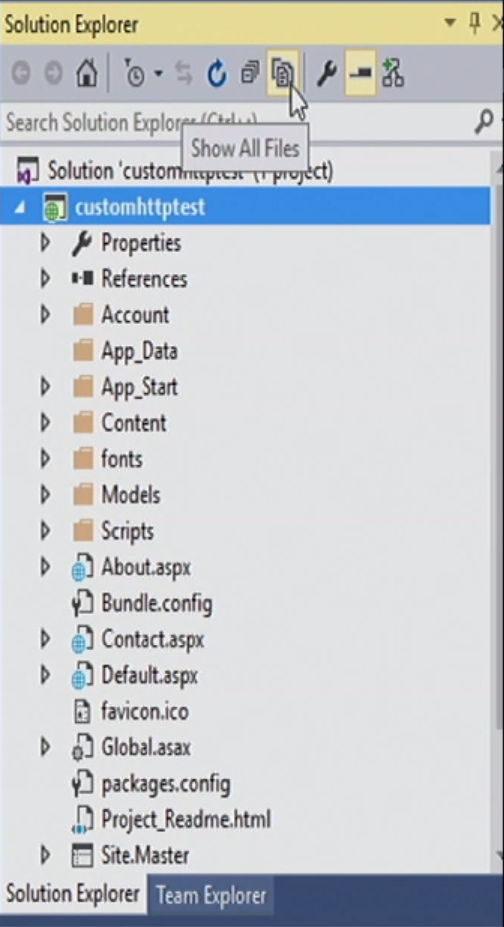
Azure web site url

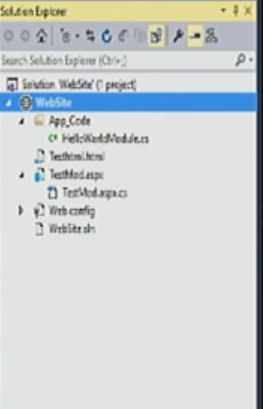
See attached sample code for writing custom http module

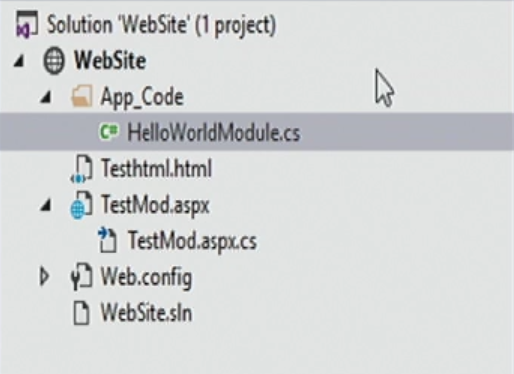


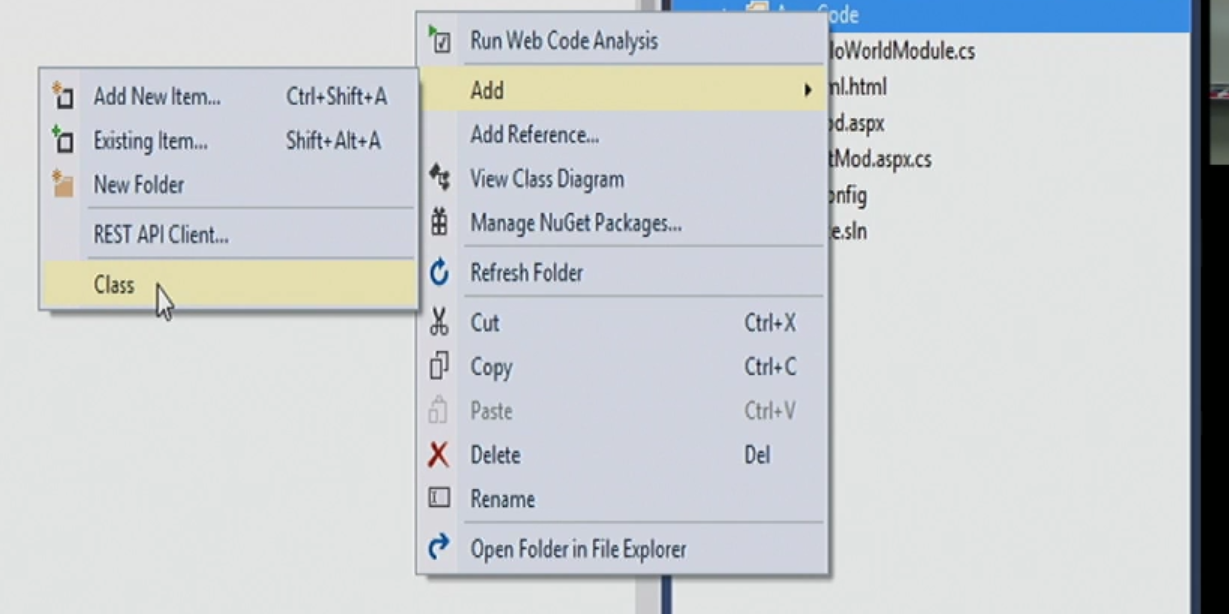












using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

/// <summary>

/// Summary description for BlockIP

/// </summary>

public class BlockIP: IHttpModule

{

public BlockIP()

{

}

public String ModuleName

{

get { return "BlockIP"; }

}

// In the Init function, register for HttpApplication

// events by adding your handlers.

public void Init(HttpApplication application)

{

application.BeginRequest +=

(new EventHandler(this.Application\_BeginRequest));

application.EndRequest +=

(new EventHandler(this.Application\_EndRequest));

}

private void Application\_BeginRequest(Object source,

EventArgs e)

{

// Create HttpApplication and HttpContext objects to access

// request and response properties.

HttpApplication application = (HttpApplication)source;

HttpContext context = application.Context;

string filePath = context.Request.FilePath;

string fileExtension =

VirtualPathUtility.GetExtension(filePath);

if (fileExtension.Equals(".aspx"))

{

context.Response.Write("<h1><font color=red>" +

"BlockIP: Beginning of Request" +

"</font></h1><hr>");

HttpBrowserCapabilities bc = context.Request.Browser;

context.Response.Write("<p>Browser Capabilities:</p>");

context.Response.Write("Name = " + bc.Browser + "<br>");

}

}

private void Application\_EndRequest(Object source, EventArgs e)

{

HttpApplication application = (HttpApplication)source;

HttpContext context = application.Context;

string filePath = context.Request.FilePath;

string fileExtension =

VirtualPathUtility.GetExtension(filePath);

if (fileExtension.Equals(".aspx"))

{

context.Response.Write("<hr><h1><font color=red>" +

"BlockIP: End of Request</font></h1>");

}

}

public void Dispose() { }

}

  public partial class \_Default : System.Web.UI.Page

    {

        protected void Page\_Load(object sender, EventArgs e)

        {

            if(!IsPostBack)

            {

                string ip = "xx.xxx.xx.xx"; //the restrictive ip

                if (Page.Request.UserHostAddress.Equals(ip))

                {

                    Response.Redirect("Error.aspx");

                }

            }

        }

    }

Web.config file:

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<system.webServer>

<modules>

<add name="BlockIP" type="BlockIP"/>

</modules>

</system.webServer>

<system.web>

<compilation debug="true" targetFramework="4.5.1" />

<httpRuntime targetFramework="4.5.1" />

</system.web>

</configuration>

<security>

<ipSecurity allowUnlisted="true">

<!-- this line allows everybody, except those listed below -->

<clear/>

<!-- removes all upstream restrictions -->

<add ipAddress="192.92.1.9"/>

</ipSecurity>

</security>

public class BlockIP : IHttpModule

{

public BlockIP() { }

public void Init(HttpApplication context)

{

context.BeginRequest += new EventHandler(Application\_BeginRequest);

}

private void Application\_BeginRequest(object source, EventArgs e)

{

HttpContext context = ((HttpApplication)source).Context;

string ipAddress = context.Request.UserHostAddress;

if (!IsValidIpAddress(ipAddress))

{

context.Response.StatusCode = 403; *// (Forbidden)*

}

}

private bool IsValidIpAddress(string ipAddress)

{

return (ipAddress == "127.0.0.1");

}

public void Dispose() { */\* clean up \*/* }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

/// <summary>

/// Summary description for BlockIP

/// </summary>

public class BlockIP : IHttpModule

{

public BlockIP()

{

}

public String ModuleName

{

get { return "BlockIP"; }

}

// In the Init function, register for HttpApplication

// events by adding your handlers.

public void Init(HttpApplication application)

{

application.BeginRequest +=

(new EventHandler(this.Application\_BeginRequest));

}

private void Application\_BeginRequest(Object source,

EventArgs e)

{

// Create HttpApplication and HttpContext objects to access

// request and response properties.

HttpApplication application = (HttpApplication)source;

HttpContext context = application.Context;

string ipAddress = context.Request.UserHostAddress;

if (!IsValidIpAddress(ipAddress))

{

context.Response.Write("You can not access this website");

}

else

{

context.Response.Write("<h1><font color=red>" +

"BlockIP: Beginning of Request" +

"</font></h1><hr>");

HttpBrowserCapabilities bc = context.Request.Browser;

context.Response.Write("<p>Browser Capabilities:</p>");

context.Response.Write("Name = " + bc.Browser + "<br>");

}

private bool IsValidIpAddress(string checkIP)

{

return (checkIP != "192.168.1.98" || checkIP != "108.92.106.178");

}

public void Dispose() { }

}

}

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<system.webServer>

<modules>

<add name="BlockIP" type="BlockIP"/>

</modules>

</system.webServer>

<system.web>

<compilation debug="true" targetFramework="4.5.1" />

<httpRuntime targetFramework="4.5.1" />

</system.web>

</configuration>