RouteConfig.cs

public static void RegisterRoutes(RouteCollection routes)

{

Route myRoute = new Route(

"{controller}/{action}/{id}",

new RouteValueDictionary { { "controller", "Home" }, { "action",

"Index" }, { "id", "1" } },

new MvcRouteHandler());

routes.Add(myRoute);

}

HttpModule

Global.asax

[assembly: PreApplicationStartMethod(typeof(MvcApplication), "Register")]

Namespace XYZ

{

public class MvcApplication : System.Web.HttpApplication

{

public static void Register()

{

HttpApplication.RegisterModule(typeof(LogModule));

}

}

}

LogModule.cs

Namespace XYZ

{

public class LogModule : IHttpModule

{

public void Dispose()

{

}

public void Init(HttpApplication context)

{

context.LogRequest += LogEvent;

}

private void LogEvent(object src, EventArgs args)

{

if (HttpContext.Current.CurrentNotification ==

RequestNotification.LogRequest)

{

if ((MvcHandler)HttpContext.Current.Handler != null)

{

Debug.WriteLine("This was logged!");

}

}

}

}

}

HttpHandler

1. Create a Handler Folder

namespace HealthAssist.Handlers

{

public class SampleHandler : IHttpHandler

{

public bool **IsReusable**

{

get { return false; }

}

public void **ProcessRequest**(HttpContext context)

{

context.Response.Write("<p>This was generated by the sample

handler.</p>");

}

}

}

1. RouteConfig.cs

namespace HealthAssist

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.Add(new Route("home/about", new SampleRouteHandler()));

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id =

UrlParameter.Optional }

);

}

}

public class SampleRouteHandler : IRouteHandler

{

public IHttpHandler GetHttpHandler(RequestContext requestContext)

{

return new SampleHandler();

}

}

}

HttpModule

1. Web.config

<configuration>

<configSections>

<section name="redirects" type="HealthAssist.Configuration.RedirectSection"/>

</configSections>

<redirects>

<add Title="Marketing" Old="/Home/About" New="/Home/Contact" />

<add Title="ContactChange" Old="/Home/Services" New="/Home/Contact" />

</redirects>

</configuration>

1. Add Folder Configuration
2. Add class Redirect.cs

namespace HealthAssist.Configuration

{

public class Redirect : ConfigurationElement

{

[ConfigurationProperty("Old")]

public string Old

{

get { return (string)this["Old"]; }

set { this["Old"] = value; }

}

[ConfigurationProperty("New")]

public string New

{

get { return (string)this["New"]; }

set { this["New"] = value; }

}

[ConfigurationProperty("Title")]

public string Title

{

get { return (string)this["Title"]; }

set { this["Title"] = value; }

}

}

}

1. Add Class RedirectCollection.cs

namespace HealthAssist.Configuration

{

public class RedirectCollection : ConfigurationElementCollection

{

protected override ConfigurationElement CreateNewElement()

{

return new Redirect();

}

protected override object GetElementKey(ConfigurationElement element)

{

return ((Redirect)element).Title;

}

}

}

1. Add Class RedirectSection.cs

namespace HealthAssist.Configuration

{

public class RedirectSection : ConfigurationSection

{

[ConfigurationProperty("", IsDefaultCollection = true)]

[ConfigurationCollection(typeof(RedirectCollection))]

public RedirectCollection Redirects

{

get { return (RedirectCollection)base[""]; }

}

}

}

1. Add Class RedirectModule.cs

namespace HealthAssist.Configuration

{

public class RedirectModule : IHttpModule

{

private HttpApplication \_context;

public void **Dispose**()

{

}

public void **Init**(HttpApplication context)

{

\_context = context;

context.PostMapRequestHandler += RedirectUrls;

}

public void RedirectUrls(object src, EventArgs args)

{

RedirectSection section = (RedirectSection)WebConfigurationManager.GetWebApplicationSection("redirects");

foreach (Redirect redirect in section.Redirects)

{

if (redirect.Old == \_context.Request.RequestContext.HttpContext.Request.RawUrl)

{

\_context.Response.Redirect(redirect.New);

}

}

}

}

}

1. Select the empty project
2. Install-package Microsoft.owin.host.systemweb
3. Create the entry point for the application