

[articles](#) [quick answers](#) [discussions](#) [features](#) [community](#) [help](#)[Articles](#) » [Web Development](#) » [ASP.NET](#) » [General](#)

URL Routing with ASP.NET 4.0

**Brij**29 Apr 2010 [CPOL](#)Rate this:  4.93 (84 votes)

This article discusses one of the exciting features of ASP.NET 4.0, i.e., URL Routing. Hope you all like it.

[Download demo app - 376.23 KB](#)

Table of Contents

- [Introduction](#)
- [Prerequisite](#)
- [What is URL Routing](#)
- [Why URL Routing](#)
- [URL Routing Earlier](#)
- [URL Routing with ASP.NET 4.0](#)
- [Bidirectional Routing](#)
- [Components of ASP.NET 4.0 URL Routing](#)
 - [Routing Handler](#)
 - [Expression Builders](#)
- [An Example](#)

- [Feedback and Suggestions](#)
- [References](#)
- [History](#)

Introduction

One thing that has always pinched me is the long URLs that I used to see in several of my projects. Due to better manageability, I used to have several folders, i.e. folder hierarchy for maintainability of my application. But the thing that I didn't ever like was the full physical path of pages in URLs. There are other ways which allow us to get rid of it like URL rewriting, but I didn't like that.

We could also have our own **custom URL route handler** or some **third party solution**, but they have never been strong enough for an application.

Frankly speaking, one of my clients asked me several times, "Why does this .aspx extension displays in the URL" and also told me that he does not like these long URLs. I used to tell him, this is the normal behavior of **ASP.NET application**. But we can change it with our own custom solution. However, that will require a little bit of time and also require a lot of testing.

But when Microsoft released **ASP.NET MVC 2** with .NET Framework 3.5 SP1, and provided this **Routing** feature with it, I was very happy and started exploring it. In the meantime, I thought that it should also be provided with ASP.NET, because every time you may not require to follow MVC architecture and also found that MVC has nothing special to do with URL Routing. So I was expecting and waiting for **ASP.NET Webform Routing**.

Now Microsoft introduced URL Routing with ASP.NET 4.0. URL routing is **fully integrated, very powerful** and **straightforward**.

Prerequisite

- Visual Studio 2010

What is URL Routing

We access our webapplication using some URL that is normally the physical path of the pages. So URL Routing is a way to provide our own URL in lieu of the physical path of the page. One other way, Routing allows us a way to configure our application to accept a requested URL which actually doesn't map to physical files. From the security perspective of the application, it's important because one can easily know the solution structure of the application.

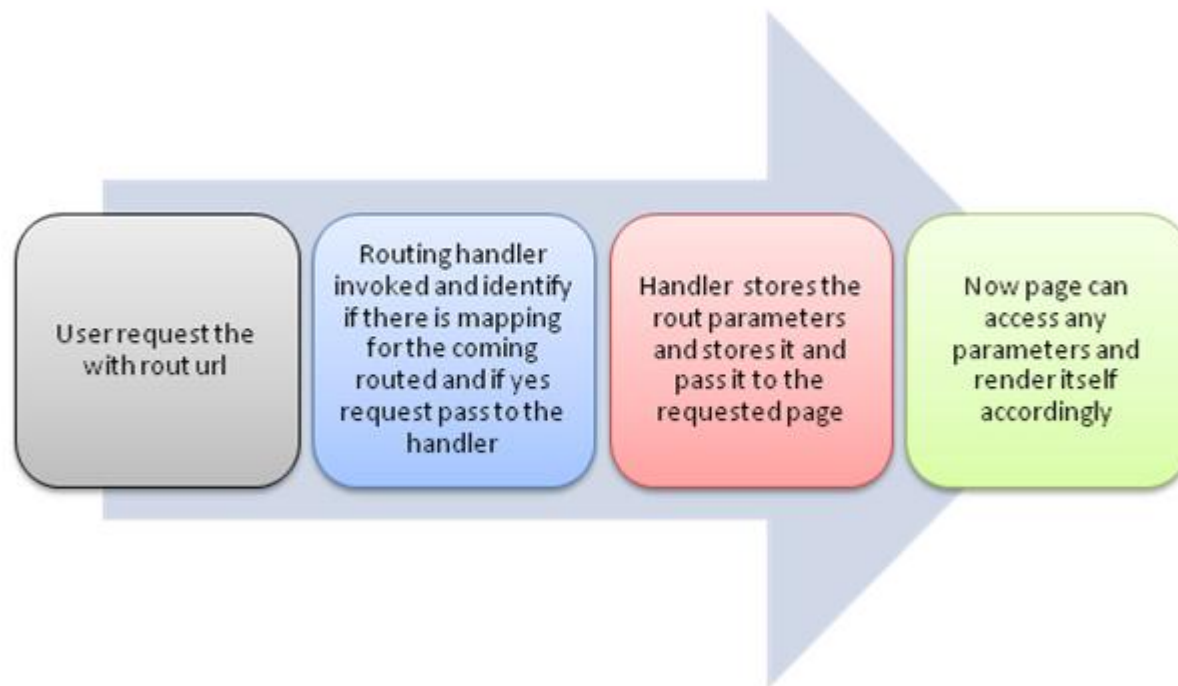


Fig 1.0: URL Routing Processing

Why URL Routing

In 1999, on usability Jakob Nielsen wrote on his blog "[URL as UI](#)" 6 essential points about URL. These are:

- A domain name that is easy to remember and easy to spell
- Short URLs
- Easy to type URLs
- URLs that visualize site structure
- URLs that are "hackable" to allow users to move to higher levels of the information architecture by hacking off the end of the URL
- Persistent URLs that don't change

As from all these points, we can see the URLs as a part of user interface and it should be simple and easy. URLs should also made us visualize the structure of our application which might be a security concern for us, etc.

ASP.NET 4.0 provides us the features, taking care of all the points mentioned above.

Also, these URLs helps in **SEO (Search engine optimization)** and improve the page hits but put in the appropriate keywords.

URL Routing Earlier

Earlier, URL routing was not so easy. For that, we require to have our own custom handler in a way that whenever a URL is requested, our custom route handler class should be invoked, and forward the request to the appropriate requested page or we could have some third party solution. So let's say if we are going to develop our own custom handler, then what do we need to do.

1. We define the mapping in *Global.asax*, which maps a route pattern to route handler class.
2. We require to develop a **Route handler** class which actually receives the URL, parses it, stores any route parameters into some location that should be accessible to the requested page and returns the instance of requested page or forwards the request to the **HTTPHandler** that handles the requested route.
3. Writing code in the target page in the way so that it can fetch the route parameters and renders the page accordingly.

So you can imagine that it was not a straightforward task. It also requires some considerable amount of effort to develop it.

URL Routing with ASP.NET 4.0

As I already discussed in my introduction section, that ASP.NET 4.0 provides us a simplified and robust way to handle the entire URL routing mechanism. To provide URL routing, ASP.NET is now equipped with myriad classes and a number of methods to provide this feature, which allows to easily decouple the URL with physical files. We just need to use them.

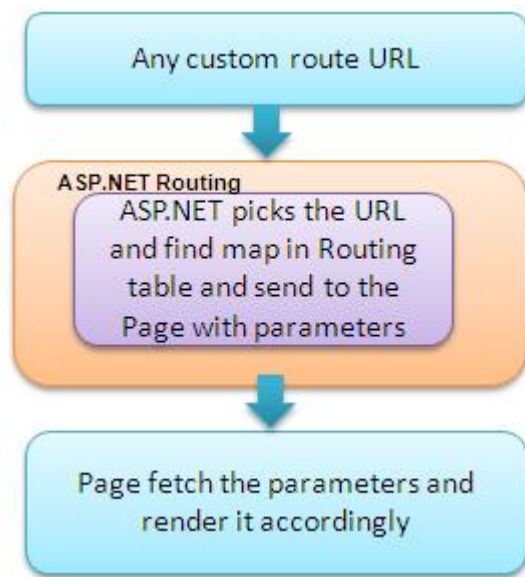


Fig 2.0: ASP.NET 4.0 URL Routing Flow

ASP.NET 4.0 router enables to define any kind of custom routes and we can easily map it to our webform page.

ASP.NET 4.0 also made our life simpler by providing us the feature "Bi- Directional routing" with the help of several components like **Route Table**, **Page Routehandler** and **ExpressionBuilders**.

What is Bi-directional Routing

With the help of the Route table, we can not only decode the Routed URL with the Route table and with the help of other methods provided by the ASP.NET 4.0, we also generate the URL with the ASP.NET routing mechanism, which gives us the opportunity, not to hard code the URL at several places, rather it will be dynamically generating the URLs with the help of Routing Definition.

So we just require to change the Route Table on any changes in the URL, and don't need to change it in several other places throughout the solution.

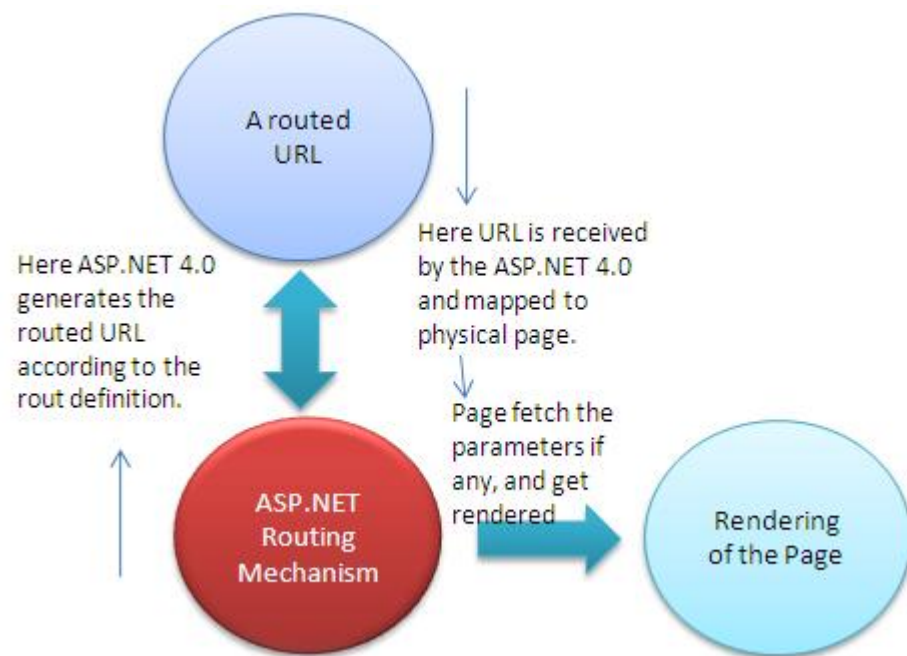


Fig 3.0: Bi-directional Routing

Components of ASP.NET 4.0 URL Routing

There are two main components of ASP.NET 4.0 URL Routing.

RoutingHandler

This is basically a normal **HTTPHandler**, which is responsible for looking into all the incoming URL requests, and looking for any Routing definition available for the URL, if yes, then pass the request and data to the corresponding resource.

Expression Builders

Expressions are provided with ASP.NET 4.0 to facilitate **Bi-Directional Routing** and more. Basically there are two types of **ExpressionBuilders**.

1. **RouteURLExpressionBuilder**: As the name suggests, it provides the syntax which results in the value, i.e., URL based on **RouteName** and **Parameter** according to the Route definitions we have.
2. **RouteValueExpressionBuilder**: As above, it generates the URL, **RouteValueExpressionBuilder** provides a syntax which receives the value from the **RouteName** and **Parameter** from **RoutedURL**.

There are also a few new properties **HttpRequest.RequestContext** and **Page.RouteData** which facilitate the availability of the parameters to all resources.

An Example

In this example, I will be displaying the image and name of the book on my web page.

Here, I will be going to an ASP.NET application and will take you step by step to create the application.

Note:

- I have used **VS2010 Beta 2** version for this sample.
- We have to use the namespace **System.Web.Routing** in our application to access Routing specific classes and methods.

Step 1

Define the Route in **Application_Start** of *Global.asax*. Also include namespace **System.Web.Routing**.

[Hide](#) [Copy Code](#)

```
void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup
    RouteTable.Routes.MapPageRoute("StoreRoute",
    "BookStore/{Name}",
    "~/Webpages/BookStore/ViewBookDemo.aspx");
}
```

Step 2

1. Now in *ViewBookDemo.aspx*, I have four links. In the first two, I have hardcoded the URL and in the last, I used **RouteURLExpressionBuilder** to generate the URL dynamically. That is known as **Bi- Directional** routing. This is the one I also liked the most, in several cases, we have had links in our pages that were generally hard coded.

```

<td>
  <table>
    <tr>
      <td><asp:HyperLink ID="lnk1" runat="server" NavigateUrl="~/Bookstore/CSS" Text="CSS" /></td>
      <td><asp:HyperLink ID="lnk2" runat="server" NavigateUrl="~/Bookstore/Django" Text="Django"/></td>
      <!--Here in below both link using RouteUrlExpressionBuilder to generate the URL dynamically-->
      <td><asp:HyperLink ID="lnk3" runat="server" NavigateUrl="<%=RouteUrl:Name=IPhone %>" Text="IPhone" /></td>
      <td><asp:HyperLink ID="lnk4" runat="server" NavigateUrl="<%=RouteUrl:Name=Linq %>" Text="Linq" /></td>
    </tr>
  </table>
</td>
</tr>
<tr>
  <!--Here in below both link using RouteValueExpressionBuilder to get the Value of the URL data-->
  <td><h2><asp:Label ID="lblBookName" runat="server" Text="<%=RouteValue:Name%>" ></asp:Label></h2></td>
</tr>

```

2. I also have a **Label** in the page and I set the **Text** property dynamically with the help of Routing information as above.

Step 3

Here, I fetch the parameter from the Routing Table and set the **Image1** URL dynamically.(Include namespace **System.Web.Routing**):

Hide Copy Code

```

//fetching the parameter that is Route table
string name = Page.RouteData.Values["Name"] as string;
if (name != null)
{
    if (name == "CSS")
    {
        Image1.ImageUrl = "~/images/css.jpg";
    }
    else if (name == "Django")
    {
        Image1.ImageUrl = "~/images/django.jpg";
    }
    else if (name == "IPhone")
    {
        Image1.ImageUrl = "~/images/iphone.jpg";
    }
    else if (name == "Linq")
    {
        Image1.ImageUrl = "~/images/Linq.jpg";
    }
}
}

```

To view the entire code, please download the attachment.

Now my solution hierarchy in the demo application is:

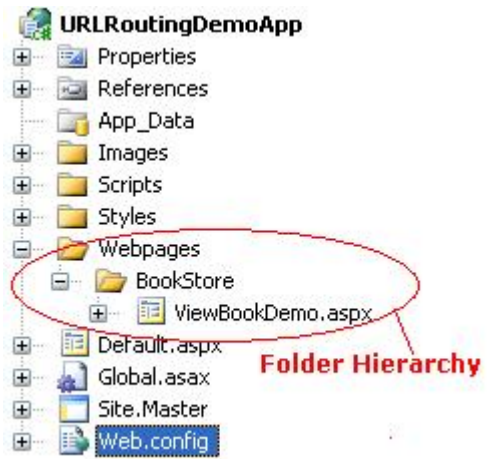


Fig 4.0: Solution Explorer

And according to it, my URL should be <http://localhost:2039/Webpages/BookStore/ViewBome.aspx?Name=CSS> as:



Fig 5.0: Demo Application

Now when a user requests this application, the **request** is processed as:

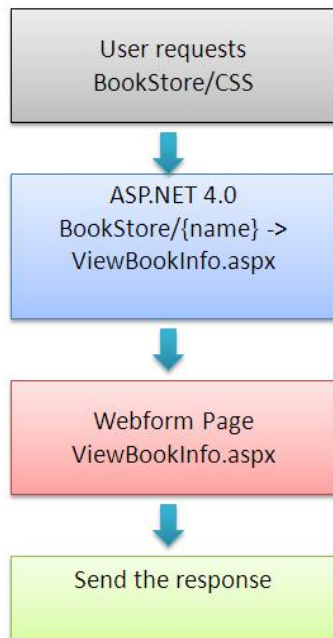


Fig 6.0: Request Processing

Feedback and Suggestions

Feedback is the key for me. I would request all of you to share your feedback and give me some suggestions, which would encourage and help me in writing more articles.

References

- <http://haacked.com/archive/2008/03/11/using-routing-with-webforms.aspx>
- http://haacked.com/archive/2006/11/29/Express_Yourself_With_Custom_Expression_Builders.aspx
- <http://sankarsan.wordpress.com/2009/08/09/asp-net-4-0-webforms-routing-part-i/>
- <http://weblogs.asp.net/scottgu/archive/2009/10/13/url-routing-with-asp-net-4-web-forms-vs-2010-and-net-4-0-series.aspx>

History

- 29th April, 2010: Initial post

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOl\)](#)

Share

About the Author

**Brij**

Software Developer (Senior)

India

Brij is a 3-times Microsoft MVP in ASP.NET/IIS Category and a passionate .NET developer. More than 6 years of experience in IT field, currently serving a MNC as a Tech Lead/Architect.

He is a very passionate .NET developer and have expertise over Web technologies like ASP.NET 2.0/3.5/4.0, jQuery, JSON, Javascript, IIS and related technologies. He is also a Exchange Server (EWS) Speciali...

[show more](#)

Comments and Discussions

You must [Sign In](#) to use this message board.



Spacing

Relaxed ▼

Layout

Normal ▼

Per page

25 ▼

Update

First Prev Next

Blank Page loaded	Mr_cool	12-Nov-17 23:50
My vote of 5	csharpbd	6-Jul-14 10:24
I have to remove ~/Views/PageName from all the request	Member 10658225	10-Mar-14 5:36
Good	williambui1412	14-Jan-14 22:05
bind dynamic name for url rewiting in listview.	Grishma Doshi	6-Jan-14 0:24
It's clear and easy to understand	Rohith Reddy Vadiyala	5-Dec-13 20:10
Why URL Routing is not working on the production server	Munjal Pandya	16-Sep-13 20:34
Excellent article and URLRoutingDemoApp	cliebza	27-Jul-13 5:22
Re: Excellent article and URLRoutingDemoApp	Brij	28-Jul-13 9:51
4 vote : it Slow if use with switch User control	Wit Peter	10-Jul-13 19:20
My vote of 3	Amir Mehrabi-Jorshari	26-Jun-13 10:13
Message Closed	Brij	27-Jun-13 6:40
Re: My vote of 3	Amir Mehrabi-Jorshari	28-Jun-13 8:42
Thanks	Saran89	27-May-13 21:15
rediect to login.aspx	plazzasele	30-Mar-13 8:06
My vote of 5	ojanacek	27-Mar-13 12:37

[Thanks](#) **Altaf N Patel**

26-Feb-13 8:45

 **My vote of 4** **veenusethi**

18-Feb-13 0:49

 **My vote of 5** **sandeepkumarvemula**

11-Feb-13 22:49

 **My vote of 5** **adriansc**

15-Jan-13 22:16

 **My vote of 5** **Savalia Manoj M**

8-Nov-12 17:34

 **Re: My vote of 5** NAVNEET KUMAR SINGH

3-Dec-12 17:05

 **My vote of 5** **hsuhsutin**

30-Sep-12 20:48

 **How to route to [www.myDomainName/MemberId](#)** **mustafeena**

13-Sep-12 11:44

 **Tracking** **Patrick Harris**

18-Aug-12 20:37

Last Visit: 4-Nov-19 0:36 Last Update: 14-Jan-20 11:48

[Refresh](#)[1](#) [2](#) [3](#) [Next »](#) General  News  Suggestion  Question  Bug  Answer  Joke  Praise  Rant  Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

[Permalink](#)[Advertise](#)[Privacy](#)[Cookies](#)[Terms of Use](#)Layout: [fixed](#) | [fluid](#)Article Copyright 2010 by Brij
Everything else Copyright © [CodeProject](#), 1999-2020

Web05 2.8.200113.1