

My site uses cookies to personalise content and adverts, to provide social media features and to analyse traffic.

I'm happy with this

[Learn more](#)

Razor Pages - Understanding Handler Methods

22 May 2017 20:16

★★★★★ 5 (3 votes)

2 Comments

[ASP.NET Core \(/category/36/asp-net-core\)](/category/36/asp-net-core)

[Razor Pages \(/category/37/razor-pages\)](/category/37/razor-pages)

Handler methods is a particularly nice feature introduced with the [new ASP.NET Razor Pages framework \(https://www.mikesdotnetting.com/article/307/razor-pages-getting-started-with-the-preview\)](https://www.mikesdotnetting.com/article/307/razor-pages-getting-started-with-the-preview). This feature enables you to determine what the user was doing when they requested the page, and to execute logic accordingly without having to resort to a bunch of conditional code.

One of the most common tasks in web development is to determine which HTTP verb your visitor has used to request a page. Is this an initial GET request, or has the user POST ed a form? In classic ASP or PHP, you will usually query the Server Variables collection (`Request.ServerVariables("REQUEST_METHOD")`) or `$_Server['REQUEST_METHOD']`) to determine which verb was used, or you might query the `Request.Form` collection for the presence or absence of values. The ASP.NET Web Forms and Web Pages frameworks provide convenience methods to differentiate between POST and GET requests via the `Page.IsPostBack` and `Page.IsPost` properties respectively. ASP.NET MVC offers a different solution, mapping information from the request - the URL and the HTTP verb - to controller actions using a convention-based approach.

Handler methods in Razor Pages are also based on naming conventions. The basic methods work by matching HTTP verbs used for the request to the name of the method, and they are prefixed with "On": `OnGet()` , `OnPost()` , `OnPut()` etc. They also have asynchronous equivalents: `OnPostAsync()` , `OnGetAsync()` etc. The following example illustrates basic usage in a single Razor pages file:

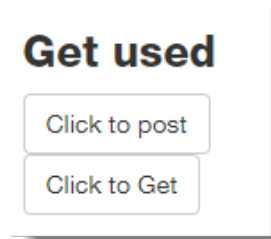
```
@page
@{
    @functions{
        public string Message { get; set; }

        public void OnGet()
        {
            Message = "Get used";
        }

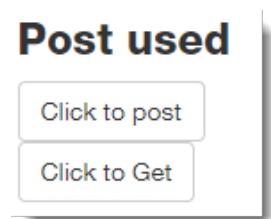
        public void OnPost()
        {
            Message = "Post used";
        }
    }
}

<h3>@Message</h3>
<form method="post"><button class="btn btn-default">Click to post</button></form>
<p><a href="/handlerexample" class="btn btn-default">Click to Get</a></p>
```

When the page is first navigated to, the "Get used" message is displayed because the HTTP GET verb was used for the request, firing the `OnGet()` handler.



When the "Click to post" button is pressed, a form is posted and the `OnPost()` handler fires, resulting in the "Post used" message being displayed.



The button underneath is a hyperlink, which initiates a GET request, resulting in the "Get used" message being displayed once more.

Named Handler Methods

Imagine that you have a number of forms on the page (perfectly legal everywhere except Web Forms...) - how will you know which one was submitted? Well, I kind of take a bit of an unfair dig at Web Forms, but in fact the solution to the problem is not too dissimilar to Web Forms button click event handlers. Razor Pages introduces *named handler methods*. The following code shows a collection of named handler methods declared in a code block at the top of a Razor page (although they can also be placed in the `PageModel` class if you are using one):

```
@page
@{
    @functions{
        public string Message { get; set; } = "Initial Request";

        public void OnGet()
        {

        }

        public void OnPost()
        {
            Message = "Form Posted";
        }

        public void OnPostDelete()
        {
            Message = "Delete handler fired";
        }

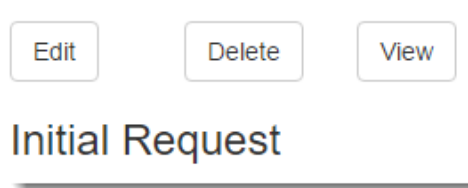
        public void OnPostEdit(int id)
        {
            Message = "Edit handler fired";
        }

        public void OnPostView(int id)
        {
            Message = "View handler fired";
        }
    }
}
```

The convention (in Preview 1) is that the name of the method is appended to "OnPost" or "OnGet". The next step is to associate a specific form action with a named handler. This is achieved by setting the `asp-page-handler` attribute value for a form taghelper:

```
<div class="row">
  <div class="col-lg-1">
    <form asp-page-handler="edit" method="post">
      <button class="btn btn-default">Edit</button>
    </form>
  </div>
  <div class="col-lg-1">
    <form asp-page-handler="delete" method="post">
      <button class="btn btn-default">Delete</button>
    </form>
  </div>
  <div class="col-lg-1">
    <form asp-page-handler="view" method="post">
      <button class="btn btn-default">View</button>
    </form>
  </div>
</div>
<h3 class="clearfix">@Model.Message</h3>
```

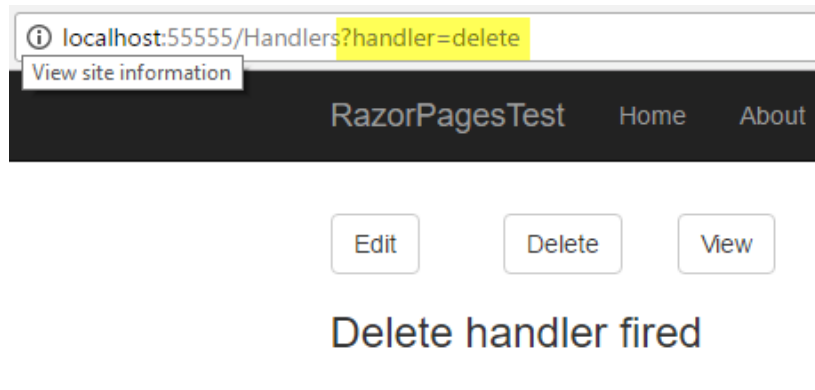
The code above renders as three buttons, each in their own form along with the default value for the Message property:



The name of the handler is added to the form's action as a querystring parameter:

```
<div class="col-lg-1">
  <form method="post" action="/Handlers?handler=edit">
    <button class="btn btn-default">Edit</button>
    <input name="__RequestVerificationToken" type="hidden" value="..." />
  </div>
<div class="col-lg-1">
  <form method="post" action="/Handlers?handler=delete">
    <button class="btn btn-default">Delete</button>
    <input name="__RequestVerificationToken" type="hidden" value="..." />
  </div>
<div class="col-lg-1">
  <form method="post" action="/Handlers?handler=view">
    <button class="btn btn-default">View</button>
    <input name="__RequestVerificationToken" type="hidden" value="..." />
  </div>
```

As you click each button, the code in the handler associated with the querystring value is executed, changing the message each time.



If you prefer not to have querystring values in the URL, you can use routing and add an optional route value for "handler" as part of the `@page` directive:

```
@page "{handler?}"
```

The name of the handler is then appended to the URL:

```
<div class="row">
  <div class="col-lg-1">
    <form method="post" action="/Handlers/edit">
      <button class="btn btn-default">Edit</button>
      <input name="__RequestVerificationToken" type="hidden"
    </form>
  </div>
  <div class="col-lg-1">
    <form method="post" action="/Handlers/delete">
      <button class="btn btn-default">Delete</button>
      <input name="__RequestVerificationToken" type="hidden"
    </form>
  </div>
  <div class="col-lg-1">
    <form method="post" action="/Handlers/view">
      <button class="btn btn-default">View</button>
      <input name="__RequestVerificationToken" type="hidden"
```

Parameters

Handler methods can be designed to accept parameters:

```
public void OnPostView(int id)
{
    Message = $"View handler fired for {id}";
}
```

The parameter name must match a form field name for it to be automatically bound to the value:

```
<div class="col-lg-1">
  <form asp-page-handler="view" method="post">
    <button class="btn btn-default">View</button>
    <input type="hidden" name="id" value="3" />
  </form>
</div>
```

[Edit](#)[Delete](#)[View](#)

View handler fired for 3

Alternatively, you can use the form tag helper's `asp-route` attribute to pass parameter values as part of the URL, either as a query string value or as route data:

```
<form asp-page-handler="delete" asp-route-id="10" method="post">
  <button class="btn btn-default">Delete</button>
</form>
```

You append the name of the parameter to the `asp-route` attribute (in this case "id") and then provide a value. This will result in the parameter being passed as a querystring value:

A screenshot of a browser's address bar. The text 'localhost:55555/Handlers/delete?id=10' is displayed. The 'id=10' portion is highlighted in yellow.

Alternatively, you can extend the route definition for the page to account for an optional parameter:

```
@page "{handler?}/{id?}"
```

This results in the parameter value being added as a separate segment in the URL:

A screenshot of a browser's address bar. The text 'localhost:55555/Handlers/delete/10' is displayed. The '/10' portion is highlighted in yellow.

Summary

Razor Pages handler methods facilitate the clear separation of processing code based on user actions on the page without resorting to a confusing pile of conditional code to determine which button was clicked. They follow a clear naming convention and are easy to use. When placed in a code-behind file, they can make the page feel very much like an MVC controller. They can also be seen as analogous to event handlers in a Web Form code behind file.

◀ [Razor Pages - Getting Started With The Preview \(/article/307/razor-pages-getting-started-with-the-preview\)](/article/307/razor-pages-getting-started-with-the-preview)
[Razor Pages - The Elevator Pitch \(/article/309/razor-pages-the-elevator-pitch\)](/article/309/razor-pages-the-elevator-pitch) ▶

You might also like...

Date Posted: 22 May 2017 20:16
Last Updated: 25 May 2017 12:39
Posted by: Mike Brind
Total Views to date: 2656

2 Comments

23 May 2017 11:41 - Satyabrata Mohapatra

Nice and easy !! Great post.

26 May 2017 05:44 - Cyrus

well done, thank you.

(http://twitter.com/intent/follow?source=followbutton&variant=1.0&screen_name=mikesdotnetting)

(<https://plus.google.com/100616357343781781699>)

(<http://www.mikesdotnetting.com>)

/feed)

- ADO.NET (/category/14/ado-net) AJAX (/category/5/ajax) ASP.NET 2.0 (/category/4/asp-net-2-0)
- ASP.NET 3.5 (/category/19/asp-net-3-5) ASP.NET 5 (/category/33/asp-net-5) ASP.NET Core (/category/36/asp-net-core)
- ASP.NET Identity (/category/38/asp-net-identity) ASP.NET MVC (/category/22/asp-net-mvc)
- ASP.NET Web Forms (/category/26/asp-net-web-forms) ASP.NET Web Pages (/category/27/asp-net-web-pages)
- Book Review (/category/24/book-review) C# (/category/3/c) Classic ASP (/category/1/classic-asp)
- CSS (/category/15/css) Entity Framework (/category/23/entity-framework) EPPlus (/category/34/epplus)
- Extension Method (/category/25/extension-method) General (/category/18/general) HTML (/category/7/html)
- HTML5 (/category/30/html5) iTextSharp (/category/20/itextsharp) Javascript (/category/6/javascript)
- jQuery (/category/21/jquery) LINQ (/category/17/linq) MS Access (/category/10/ms-access)
- Razor (/category/29/razor) Razor Pages (/category/37/razor-pages) SEO (/category/35/seo) SQL (/category/8/sql)
- SQL Server Express (/category/9/sql-server-express) VB.Net (/category/16/vb-net) VBScript (/category/2/vbscript)
- Visual Studio (/category/12/visual-studio) Web API (/category/32/web-api) WebGrid (/category/31/webgrid)
- WebMatrix (/category/28/webmatrix)

Recent Comments

Pam (/article/311/sending-email-in-razor-pages#commentId9753) 30/08/2017 11:30

In response to Sending Email in Razor Pages

Mike, RazorPages sound like a nice choice for somebody still working in ASP classic who wants to to...

Robby Robson (/article/310/routing-in-razor-pages#commentId9750) 15/08/2017 00:43

In response to Routing in Razor Pages

Mike: great stuff. Now that .Core Standard 2.0 is formally out, how soon will you rewrite your book...

Satyabrata Mohapatra (/article/311/sending-email-in-razor-pages#commentId9749) 28/07/2017 08:59

In response to Sending Email in Razor Pages

Bit off topic, but congratulation sir for your MVP award. You deserve it !!!...

Satyabrata Mohapatra (/article/309/razor-pages-the-elevator-pitch#commentId9747) 23/07/2017 16:43

In response to Razor Pages - The Elevator Pitch

@Dale Severin You can continue to build apps using asp.net web pages....

Satyabrata Mohapatra (/article/311/sending-email-in-razor-pages#commentId9746) 23/07/2017 16:40

In response to Sending Email in Razor Pages

Thanks for sharing...learned a lot...

Gfw (/article/311/sending-email-in-razor-pages#commentId9745) 22/07/2017 11:53

In response to Sending Email in Razor Pages

Question... Does System.Net.Mail support SSL?...

Dale Severin (/article/309/razor-pages-the-elevator-pitch#commentId9744) 20/07/2017 03:38

In response to Razor Pages - The Elevator Pitch

I work with razor web pages extensively. I appreciate the rapid development it permits me to I am as...

Obinna Okafor (/article/310/routing-in-razor-pages#commentId9743) 14/07/2017 01:19

In response to Routing in Razor Pages

Thank you, Mike. Good post....

Satyabrata Mohapatra (/article/310/routing-in-razor-pages#commentId9742) 11/07/2017 16:02

In response to Routing in Razor Pages

Very powerful routing system!!...

Cyrus (/article/307/razor-pages-getting-started-with-the-preview#commentId9740) 05/07/2017 03:41

In response to Razor Pages - Getting Started With The Preview

How can I trim packages and services as much as possible to use just razor pages? I don't want to to...

Archives

2017

September (2) (/article/archive/2017/9)

July (2) (/article/archive/2017/7)

May (3) (/article/archive/2017/5)

February (2) (/article/archive/2017/2)

2016

October (2) (/article/archive/2016/10)

September (1) (/article/archive/2016/9)

July (2) (/article/archive/2016/7)

June (1) (/article/archive/2016/6)

May (1) (/article/archive/2016/5)

April (1) (/article/archive/2016/4)

March (2) (/article/archive/2016/3)

February (1) (/article/archive/2016/2)

January (2) (/article/archive/2016/1)

2015

December (3) (/article/archive/2015/12)

October (3) (/article/archive/2015/10)

September (2) (/article/archive/2015/9)

August (2) (/article/archive/2015/8)

July (4) (/article/archive/2015/7)

June (2) (/article/archive/2015/6)

May (2) (/article/archive/2015/5)

April (4) (/article/archive/2015/4)

March (5) (/article/archive/2015/3)

February (4) (/article/archive/2015/2)

January (4) (/article/archive/2015/1)

2014

October (2) (/article/archive/2014/10)

August (1) (/article/archive/2014/8)

July (1) (/article/archive/2014/7)

June (12) (/article/archive/2014/6)

May (11) (/article/archive/2014/5)

April (1) (/article/archive/2014/4)

March (2) (/article/archive/2014/3)

February (2) (/article/archive/2014/2)

January (1) (/article/archive/2014/1)

2013

December (1) (/article/archive/2013/12)

November (2) (/article/archive/2013/11)

October (2) (/article/archive/2013/10)

August (3) (/article/archive/2013/8)

July (1) (/article/archive/2013/7)

June (1) (/article/archive/2013/6)
May (1) (/article/archive/2013/5)
February (3) (/article/archive/2013/2)
January (2) (/article/archive/2013/1)

2012

December (4) (/article/archive/2012/12)
November (1) (/article/archive/2012/11)
October (1) (/article/archive/2012/10)
September (5) (/article/archive/2012/9)
August (2) (/article/archive/2012/8)
July (2) (/article/archive/2012/7)
June (3) (/article/archive/2012/6)
May (1) (/article/archive/2012/5)
February (1) (/article/archive/2012/2)
January (1) (/article/archive/2012/1)

2011

December (2) (/article/archive/2011/12)
October (1) (/article/archive/2011/10)
September (1) (/article/archive/2011/9)
August (6) (/article/archive/2011/8)
May (1) (/article/archive/2011/5)
April (1) (/article/archive/2011/4)
March (2) (/article/archive/2011/3)
January (5) (/article/archive/2011/1)

2010

December (3) (/article/archive/2010/12)
October (4) (/article/archive/2010/10)
September (2) (/article/archive/2010/9)
August (2) (/article/archive/2010/8)
July (9) (/article/archive/2010/7)
June (2) (/article/archive/2010/6)
May (7) (/article/archive/2010/5)
April (1) (/article/archive/2010/4)
March (1) (/article/archive/2010/3)
February (4) (/article/archive/2010/2)
January (2) (/article/archive/2010/1)

2009

December (4) (/article/archive/2009/12)
November (2) (/article/archive/2009/11)
October (4) (/article/archive/2009/10)
September (3) (/article/archive/2009/9)
August (1) (/article/archive/2009/8)
July (2) (/article/archive/2009/7)
June (4) (/article/archive/2009/6)
May (3) (/article/archive/2009/5)
April (1) (/article/archive/2009/4)
March (1) (/article/archive/2009/3)
February (2) (/article/archive/2009/2)
January (5) (/article/archive/2009/1)

2008

December (4) (/article/archive/2008/12)
November (5) (/article/archive/2008/11)
October (6) (/article/archive/2008/10)
July (1) (/article/archive/2008/7)
May (3) (/article/archive/2008/5)
April (2) (/article/archive/2008/4)

2007

November (5) (/article/archive/2007/11)
September (1) (/article/archive/2007/9)
August (8) (/article/archive/2007/8)
July (2) (/article/archive/2007/7)
June (3) (/article/archive/2007/6)
May (20) (/article/archive/2007/5)
April (14) (/article/archive/2007/4)
March (3) (/article/archive/2007/3)

© 2006 - 2017 Mike Brind