WEB PAGES WITH RAZOR SCRIPTING

NEW VIEW ENGINE IN ASP.NET

> Razor syntax is based on technology from Microsoft called ASP.NET, which in turn is based on Microsoft .NET Framework. The Razor syntax gives you all power of ASP.NET, but using a simplified syntax that's easier to learn and it can be used with existing ASP.NET Web Site.

SANJAY GUPTA
Scientist-'C'
District Informatics Officer
Jammu
sanjay.mahajan@nic.in

Edited by
Mohan Das

ASP.NET is a Web application framework developed by Microsoft to allow programmers to build dynamic Web sites and is successor to Microsoft's ASP technology. ASP.NET is built on CLR, allowing programmers to write ASP.NET code using any supported .NET language. ASP.NET supports three different development methods. All three are ASP.NET technologies for creating dynamic web applications:

- ASP.NET Web Pages focuses on adding server-side code and features simple and lightweight syntax.
- ASP.NET Web Forms is based on a page object model. Web Forms uses event-based model.
- ASP.NET MVC implements Model-View-Controller pattern.

In ASP.NET MVC3, Microsoft introduced RAZOR, a new view engine, designed to simplify the current syntax used in asp.net pages. Razor was designed as an easy to learn, compact and expressive view engine that enables a fluid coding workflow.

ABOUT ASP.NET WEB PAGES

When .NET Framework 1.0 was released you could create web sites using Web Forms. It became a success and many shifted from classic ASP and other languages to ASP.NET.

In web forms since the code were tightly coupled, it became hard to test code as you had to have access to current HttpContext and controls used by aspx file. Because of this Microsoft released ASP.NET MVC in 2007 and it solved problems by de-coupling the code and putting it to a controller, and have full control over rendering.

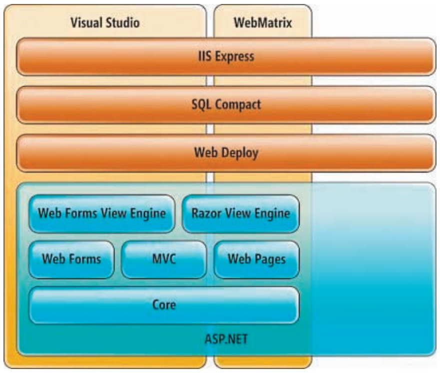
There are though still a lot of developers who still use classic ASP, PHP etc style approach. These developers want to have dynamic code on server side, and sometimes also have business logic directly in view pages since it makes easy to distribute and modify pages without need of compiling.

Microsoft released third alternative called ASP.NET Web Pages. It makes it possible to use new dynamic functions in .NET 4.0 and the rest of the .NET Framework as before. ASP.NET Web Pages can be created using C# or Visual Basic, and requires .NET 4.0. When creating Web pages you use new syntax called "Razor".

WHAT IS RAZOR?

- Razor is markup syntax for adding server-based code to web pages
- Razor has power of traditional ASP.NET markup, easier to learn and easier to use
- Razor is server side markup syntax much like ASP and PHP
- Razor supports C# and VB

Even though this syntax is simple to use, its family relationship to ASP.NET and the .NET Framework means that as your websites become more sophisticated. ASP.NET web pages with Razor syntax have the special file extension cshtml (Razor using C#) or vbhtml (Razor using VB).



ADVANTAGES OF RAZOR

The idea behind Razor is to provide an optimized syntax for HTML generation using a code-focused templating approach, with minimal transition between HTML and code. The design reduces the number of characters and keystrokes, and enables a more fluid coding workflow.

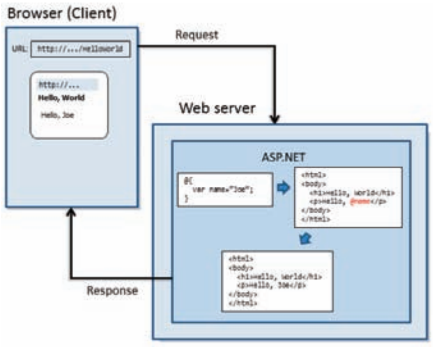
- ■ Is not a new language (no major changes to learn)
- ■ Supports IntelliSense (statement completion support)
- ■ Unit Testable
- ■ Supports "layouts" (an alternative to the "master page" concept in aspx pages)

HTML ENCODING

When you display content in a page using the @ character, ASP.NET HTML-encodes the output. This replaces reserved HTML characters (such as < and > and &) with codes that enable the characters to be displayed as characters in a web page instead of being interpreted as HTML tags or entities. Without HTML encoding, the output from your server code might not display correctly, and could expose a page to security risks.

HOW DOES IT WORK?

Razor is a simple programming syntax for embedding server code in web pages. Razor web pages can be described as HTML pages with two kinds of content: HTML content and Razor code. When the server reads the page, it runs the Razor code first, before it sends the HTML page to the browser. The code that is executed on the server can perform tasks that cannot be done in the browser. Server code can create dynamic HTML content on the fly, before it is sent to the browser.



```
@{
  var name="Joe";
}
```

```
<html>
<body>
  <h1>Hello, World</h1>
  <p>Hello, @name</p>
</body>
</html>
```

```
<html>
<body>
  <h1>Hello, World</h1>
  <p>Hello, Joe</p>
</body>
</html>
```

MAIN RAZOR SYNTAX RULES FOR C#

- ■ Razor code blocks are enclosed in @{ ... }
- ■ Inline expressions (variables and functions) start with @
- ■ Code statements end with semicolon
- ■ Variables are declared with the var keyword
- ■ Strings are enclosed with quotation marks
- ■ C# code is case sensitive
- ■ C# files have the extension .cshtml and VB files have extension .vbhtml.

RE-USABLE CONTENT

You can have reusable blocks of content (content blocks), like headers and footers, in separate files. You can also define a consistent layout for all your pages, using a layout template (layout file). Many websites have content that is displayed on every page (like headers and footers). With Web Pages you can use @RenderPage() method to import content from separate files. Content block can be imported anywhere in a web page and is just like any regular web page.

ASP.NET inserts the content blocks at the point where the RenderPage() method is called. The merged page is then sent to browser.

Another approach to creating a consistent look is to use a layout page. A layout page contains structure, but not the content, of a web page. When a web page is linked to a layout page, it will be displayed according to the layout page (template). The layout page is just like a normal web page, except from a call to the @RenderBody() method where the content page will be included.

ASP.NET HELPERS

ASP.NET helpers are components that can be accessed by single lines of Razor code. You can build your own helpers using Razor syntax, or use built-in ASP.NET helpers. Some useful Razor helpers:

1. Web Grid and Graphics
2. Google Analytics
3. Facebook & Twitter Integration
4. Sending Email
5. Validation

NUGET OVERVIEW

If you want to use a library or tool that someone else has developed, you retrieve the package from the repository and install it in your Visual Studio project or solution. Everything necessary to install a library or tool is bundled into a package (a .nupkg file).

DEVELOPMENT TOOLS

WebMatrix is a free tool that integrates a web page editor, a database utility, a web server for testing pages, and features for publishing your website. It also works for just plain HTML pages, as well as for other technologies like PHP. To install WebMatrix, you can use Microsoft's Web Platform Installer. You can also create pages by using text editor and test pages by using your existing. You can also use Visual Studio 2010 or later to work with ASP.NET Web Pages. If you don't want to use either WebMatrix or Visual Studio, you can install the component products individually using Microsoft Web Platform Installer.