Jonathan Tweedle

2019/07/10

CSHP 230 B Su 19: Web Applications In C#

Assignment 03

Creating ASP.NET websites with Master/Content Technology

# Introduction

Master content allows the creation of re-usable design elements. This could be elements such as embedded JavaScript or CSS3 styling. This reduces the need to maintain the shared content into multiple files.

# Getting Started

Start with a new empty ASP.NET Web Application that is not MVC based. Once your solution is created, you need to add a mast page to the main solution directory. This is done by right clicking the Solution in the Solution Explorer and adding a “Web Forms Master Page” (Figure 1).

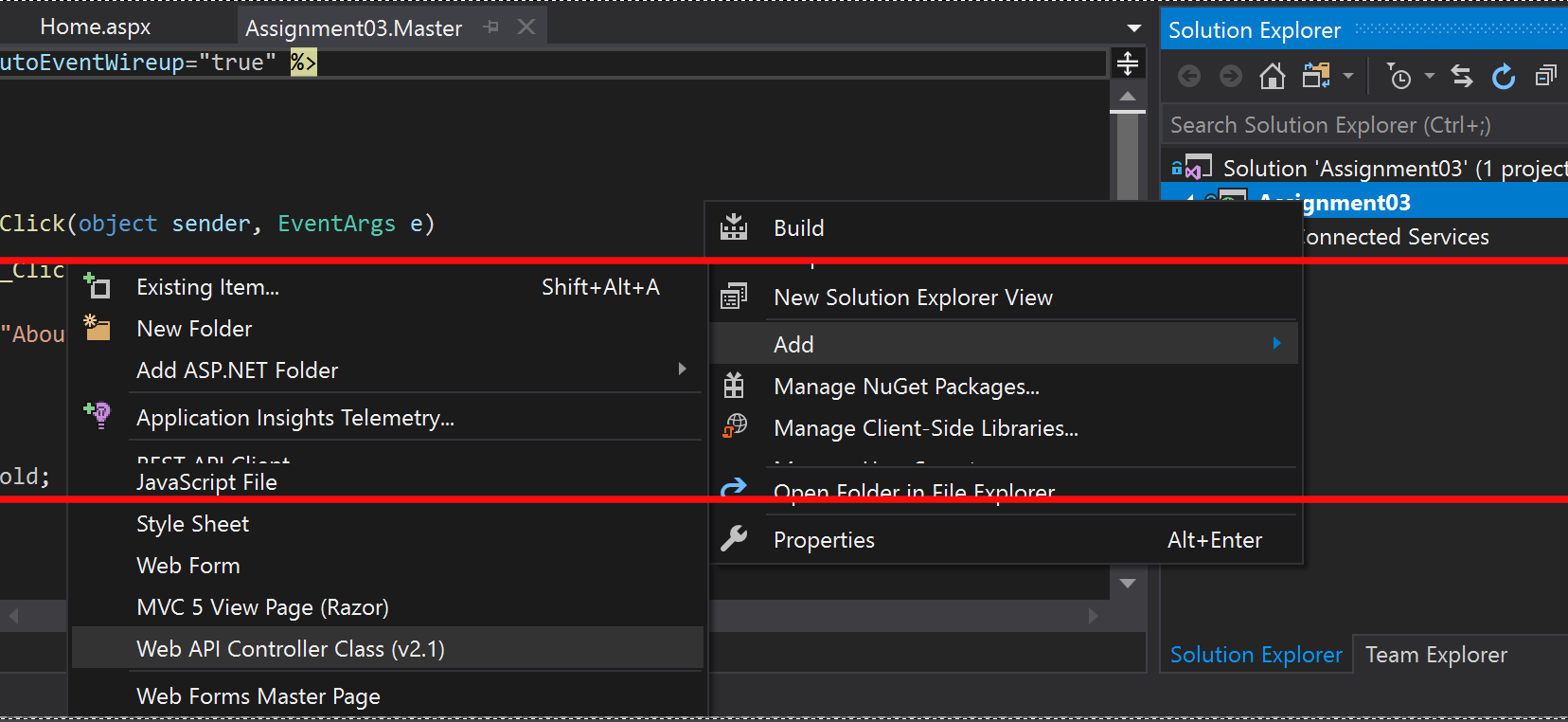


Figure 1: Adding the master page

## Master page design

You can remove the included .cs files and update the master page without using the “CodeBehind” feature to make the design a little simple to start with. Start by building the page just like you normally would, with one difference being to use the ‘Master’ razor directive instead of ‘Page’ (Figure 2). You then can design the basic layout which include the various html, head, body, as well as other shared elements like script blocks that contains your CSS3 and JavaScript. (Figure 3).

One special ASP tag helps by reserving space in the design to allow content from the other pages to be injected into the master template (Figure 4).

In the example I created a placeholder in the Head of the HTML allowing me to inject additional CSS3 and JavaScript into the design specific to a page while having a shared navigation design and space for the page content in the main body of the page (Figure 3).

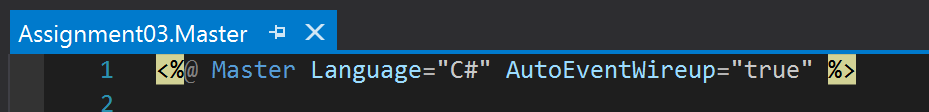


Figure 2: Master vs page directive

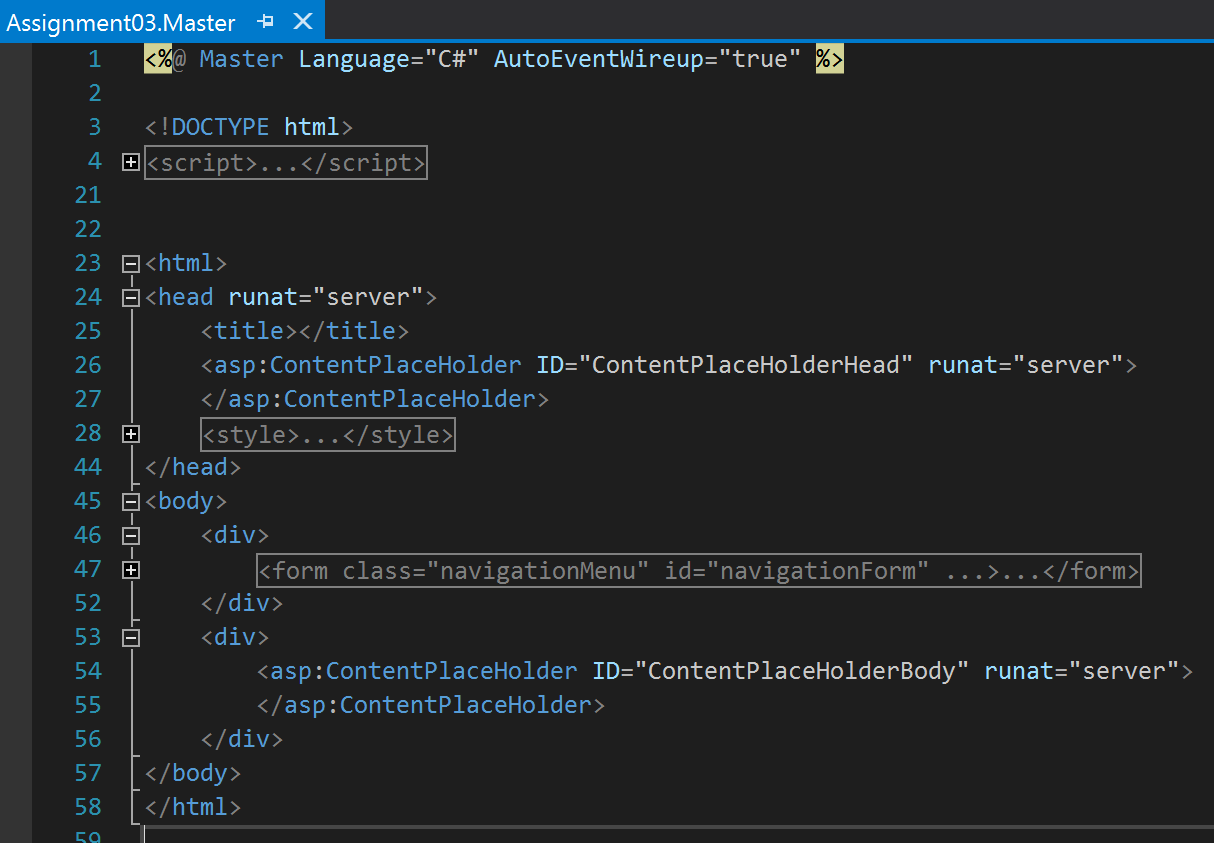


Figure 3: Master page general structure

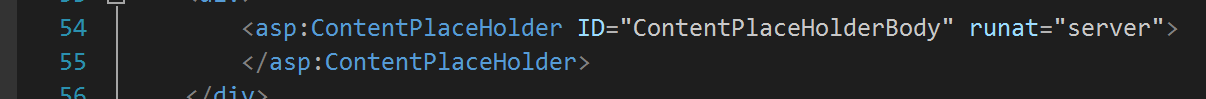


Figure 4: ASP content place holder

# Testing

With a master layout page, you cannot test the layout directly as the server will block a request to the master page.

The quickest method to make sure the master page is working, is to add a new “Web Form” page to the solution. Remove the unwanted .cs files and replace the content with a strip down simple design (Figure 5).

When you browse to the page you should then see your page layout as expected (Figure 6). The content from inside the special Content tag will be rendered inside your reserved place holder you defined in the master page (Figure 4).

Once you confirmed the master page works as needed, build out the additional pages using the same design as in the first page (Figure 5).

You will also notice that we did not define the “<html></html>”, “<head></head>”, or “<body></body>” tags as these will be provided in the master page design.

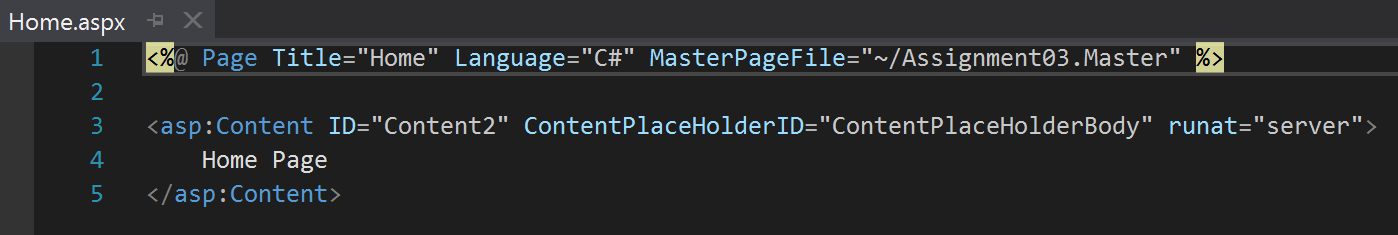


Figure 5: Home Page server code

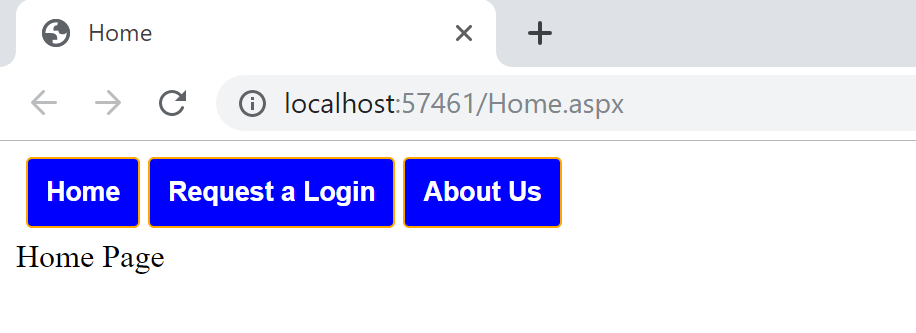


Figure 6: Home page in the browser

# Navigation

To make the navigation buttons work, the use of a server side form with buttons that are bound to click methods are used which call a special server function that causes the server to redirect the browser to another page.

Response.Redirect(“Home.aspx”);

Using this we can build out a simple navigation menu that will redirect the browser to different pages as if the user had navigated directly to those pages (Figure 7).



Figure 7: Response redirect calls

# Summary

The use of a master page can help simplify the design of more complicated sites that use a lot of shared elements to ensure a standard look and feel for a better user experience.

It also helps with site maintainability as you only have change the design in a single place and so minimizing an inconsistent look and feel.