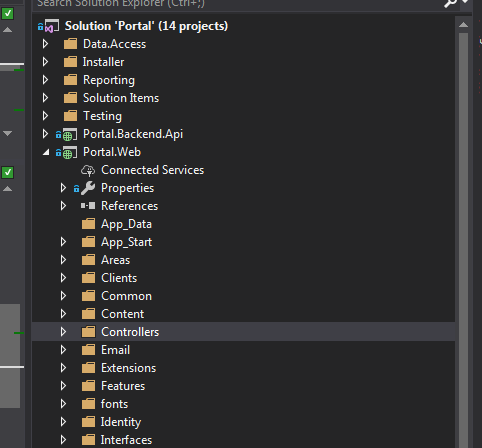
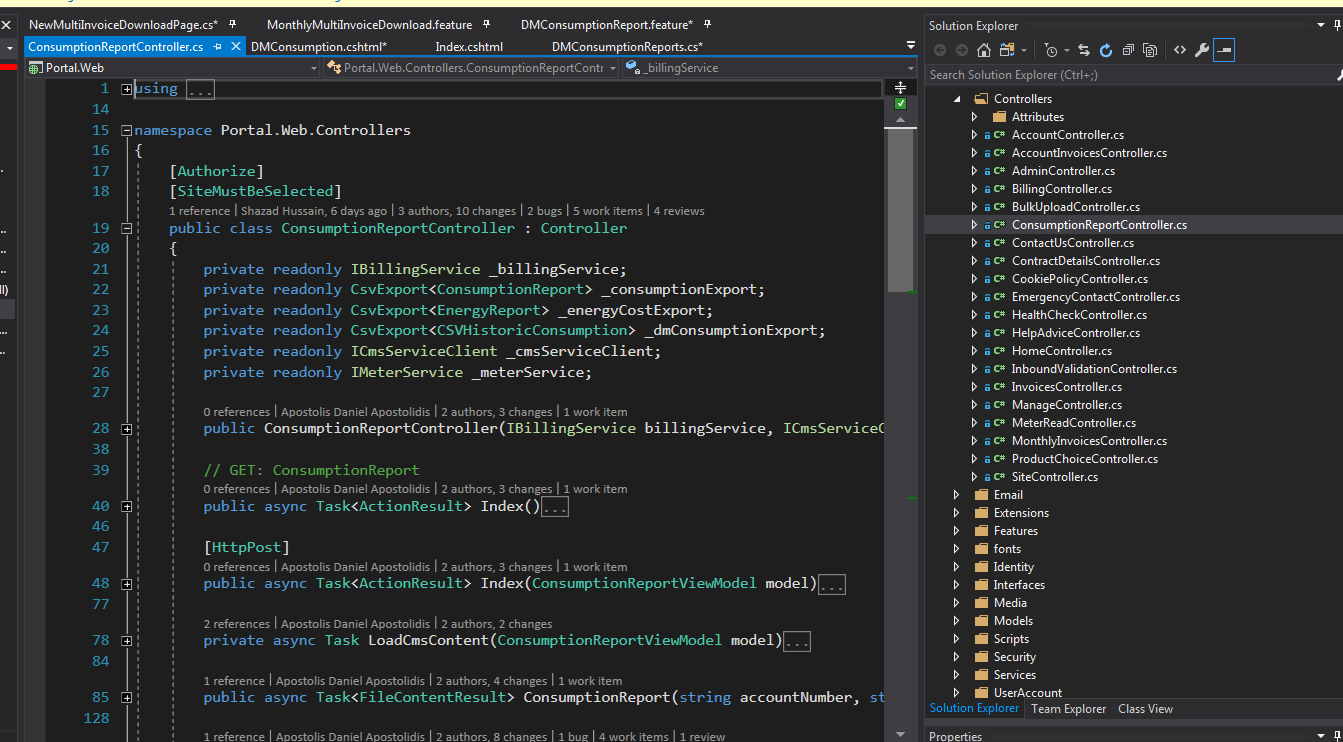
# To Add an ID to an Element on a Web Page

Sometimes a web element does not have an ID, in that case it is useful to be able to add an ID to make the element much easier to locate.

In the Portal.Web project, select the Controllers folder:



Within the Controllers folder, select the Controller file of the page that you are interested in.



Within the Controller file, locate the method that deals with area of the element you are interested in then find a line that says return View(model)

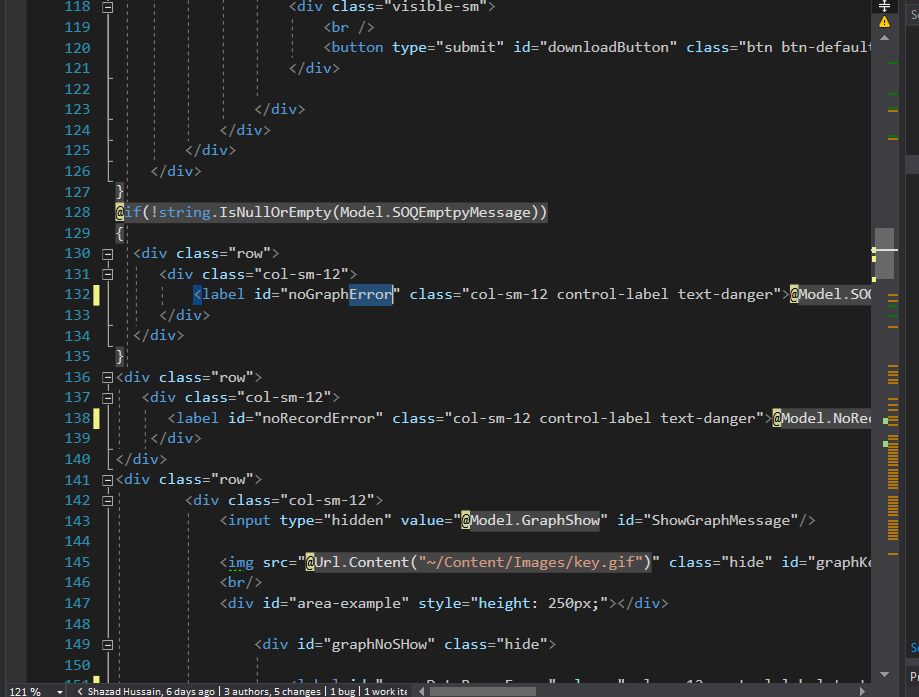
Ctrl+click on this line.

**Alternatively**

Instead of the Controllers folder, go to the Views folder and open the folder for the area of the element you are interested in. Open the .cshtml file within this folder.

Scroll down to the piece of HTML that contains the element you are interested in. Look for some (relatively) unique text from inspecting the element, e.g. its value or class. If this is not possible, there will always at least be a model property that you can look for, e.g. @Model.SOQEmptpyMessage)

Now add in your id and build your solution to pick up the changes.



## Adding ID to Element Created Using Razor Syntax

There are a few variants of this:

One that just takes 2 parameters:

@Html.ActionLink("Forgotten password?", "ForgotPassword")

One that takes a 3rd parameter that specifies the controller

@Html.ActionLink("Register", "Register", "Account", routeValues: null, htmlAttributes: new { id = "registerLink" })

And one that does not.

@Html.ActionLink("Forgotten password?", "ForgotPassword", routeValues: null, htmlAttributes: new { id = "forgottenLink" })

The first parameter is the displayed text, the second is the method that is called by the link and the last is the ID.

If you have a link that just has 2 parameters, but you want to add an ID, you have to add the other parameters.

So this

@Html.ActionLink("Forgotten password?", "ForgotPassword")

Will become this

@Html.ActionLink("Forgotten password?", "ForgotPassword", routeValues: null, htmlAttributes: new { id = "forgottenLink" })

Here is another variant

<a href="@Url.Action("ChangePassword", "Manage", new {Area = ""})" id="ChangePasswordForUser"><span>@User.Identity.GetUserName()</span></a>

In this case, you need to put the ID OUTSIDE of the @URL.Action.

Another variant

@Html.ActionLink("" + User.Identity.GetUserName() + "", "Index", "Home", routeValues: null, htmlAttributes: new { title = "Manage", id = "HomeLink" })

In this case you add the id to the list of new htmlAttributes at the end.

Another variation

@Html.ActionLink("Cookie policy", "Index", "CookiePolicy", new { Area = "" }, new { @class = "navTopLink", @id = "cookieLink" })

This is like the one above, except it does not mention htmlAttributes